

**USULAN IMPLEMENTASI *ENTERPRISE RESOURCE
PLANNING (ERP)* BERBASIS *OPEN SOURCE*
MENGUNAKAN ODOO UNTUK PENGELOLAAN
DATA PENJUALAN PRODUK PADA PT DRA
COMPONENT PERSADA**

TUGAS AKHIR

Diajukan Untuk Memenuhi Syarat Penyelesaian Program Sarjana Terapan
Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta

OLEH
AZMI NURUL AZIZAH
1314020



**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2018**

**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA**

LEMBAR PENGESAHAN

Judul Tugas Akhir :

**USULAN IMPLEMENTASI *ENTERPRISE RESOURCE PLANNING* (ERP) BERBASIS
OPEN SOURCE MENGGUNAKAN ODOO UNTUK PENGELOLAAN DATA
PENJUALAN PRODUK PADA PT DRA COMPONENT PERSADA**

Disusun oleh

Nama : Azmi Nurul Azizah

NIM : 1314020

Program Studi : Sistem Informasi Industri Otomotif

Telah diuji oleh Tim Penguji Sidang Tugas Akhir Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta Kementerian Perindustrian Republik Indonesia pada hari Senin tanggal 17 September 2018.

Jakarta, 17 September 2018

Dosen Pembimbing

Ketua Penguji

Lucky Heriyanto, S.T., M.T.I.

NIP. 197908202009011009

Ahlan Ismono, Skom, M.M.S.I

NIP. 197901072006041002

Dosen Penguji

Dosen Penguji

Drs. Jacob Saragih, M.M

NIP. 195404281986031002

Triana Fatmawati, ST, MT

NIP. 198005142005022001

**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA**

LEMBAR PERSETUJUAN DOSEN PEMBIMBING

JUDUL TUGAS AKHIR :

**USULAN IMPLEMENTASI *ENTERPRISE RESOURCE PLANNING* (ERP) BERBASIS
OPEN SOURCE MENGGUNAKAN ODOO UNTUK PENGELOLAAN DATA
PENJUALAN PRODUK PADA PT DRA COMPONENT PERSADA**

Disusun Oleh:

Nama : Azmi Nurul Azizah
NIM : 1314020
Program Studi : Sistem Informasi Industri Otomotif
Tanggal Seminar : 30 Agustus 2018
Tanggal Sidang : 17 September 2018
Tanggal Lulus : 17 September 2018

Jakarta, 20 September 2018

Dosen Pembimbing

Lucky Heriyanto, S.T., M.T.I.

NIP. 197908202009011009

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Azmi Nurul Azizah

NIM : 1314020

Berstatus sebagai mahasiswa Program Studi Sistem Informasi Industri Otomotif di Politeknik STMI Jakarta Kementerian Perindustrian RI. Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul:

USULAN IMPLEMENTASI *ENTERPRISE RESOURCE PLANNING* (ERP) BERBASIS *OPEN SOURCE* MENGGUNAKAN ODOO UNTUK PENGELOLAAN DATA PENJUALAN PRODUK PADA PT DRA COMPONENT PERSADA

- **Dibuat** dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, *survey* lapangan, dibantu oleh dosen pembimbing, dan menggunakan buku-buku maupun jurnal-jurnal ilmiah yang menjadi bahan acuan yang tertera dalam daftar pustaka Tugas Akhir ini.
- **Bukan** merupakan hasil duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai sebelumnya untuk mendapatkan gelar sarjana di Universitas/Perguruan Tinggi lain, kecuali yang telah disebutkan sumbernya dan dicantumkan pada referensi karya Tugas Akhir ini.
- **Bukan** merupakan karya tulis hasil terjemahan dari kumpulan buku atau jurnal acuan yang tertera dalam referensi karya Tugas Akhir ini.

Jika terbukti tidak memenuhi apa yang telah saya nyatakan diatas, maka saya bersedia menerima sanksi atas apa yang telah saya lakukan sesuai peraturan yang berlaku.

Jakarta, 10 Agustus 2018

Azmi Nurul Azizah

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Azmi Nurul Azizah

NIM : 1314020

Judul Laporan Tugas Akhir : Usulan Implementasi *Enterprise Resource Planning* (ERP) Berbasis *Open Source* Menggunakan Odoo Untuk Pengelolaan Data Penjualan Produk Pada PT DRA Component Persada

Pembimbing : Lucky Heriyanto, S.T., M.T.I.

Tanggal	Keterangan	Paraf
21-05-2018	Bimbingan BAB I	
08-06-2018	Revisi BAB I dan BAB II	
26-06-2018	Revisi BAB II	
28-06-2018	Revisi BAB III	
04-07-2018	Revisi BAB III	
17-07-2018	Revisi BAB IV	
19-07-2018	Revisi BAB IV	
20-07-2018	Revisi BAB IV	
23-07-2018	Revisi BAB V	
24-07-2018	Revisi BAB V	
25-07-2018	Revisi BAB V	
26-07-2018	Revisi BAB V	
03-08-2018	Revisi BAB V dan BAB VI	
06-08-2018	Revisi BAB VI	
10-08-2018	Revisi BAB V dan BAB VI	
10-08-2018	ACC Seminar	
10-09-2018	Revisi Hasil Seminar	
10-09-2018	ACC Sidang	

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri Otomotif

Dosen Pembimbing

Drs. Jacob Saragih, M.M.
NIP : 195404281986031002

Lucky Heriyanto, S.T., M.T.I.
NIP : 197908202009011009

ABSTRAK

PT DRA Component Persada merupakan perusahaan yang bergerak dalam bidang *Manufacturing Industrial Fasteners Component* yang memproduksi berbagai macam baut dengan pangsa pasar elektronik dan otomotif. Produk yang dihasilkan adalah *screw*, *bolt*, *nut* dan *pin* dengan spesifikasi yang bermacam-macam. Proses produksi yang berjalan di perusahaan ini bersifat *make to order* yang artinya proses produksi akan dilakukan apabila ada pesanan pembelian produk dari *customer*, kemudian Departemen PPIC akan mengelola PO *customer* menjadi *Job Order*, dimana *Job Order* tersebut berfungsi sebagai permintaan pembuatan produk yang ditunjukkan kepada Departemen Produksi. Dalam menangani masalah penjualan produk, dikendalikan oleh Departemen *Marketing*. Namun, pada Departemen *Marketing* ini juga tidak terlepas dari permasalahan pada proses penjualan produk yang berjalan. Kondisi Departemen *Marketing* saat ini tidak memiliki dokumen *quotation* dikarenakan proses penawaran harga dilakukan dengan cara menghubungi *customer* melalui telepon. Selain itu, proses pengajuan permintaan produksi belum terkomputerisasi dan memakan waktu cukup lama karena pada proses tersebut pemberian form PO *customer* masih berupa *copy-an* kepada Departemen PPIC dimana PO *customer* tersebut harus diolah menjadi *Job Order* sebagai permintaan produk ke Departemen Produksi untuk melakukan proses produksi serta data penjualan produk dapat diakses oleh siapa saja yang tidak berwenang dikarenakan belum adanya sistem yang dapat membatasi pengaksesan data. Maka dari itu perlu adanya pengembangan dan perancangan sebuah sistem yang dapat menjadi sebuah solusi dari permasalahan yang dihadapi. *Enterprise Resource Planning* (ERP) dipilih dan diharapkan menjadi sebuah solusi yang tepat untuk perbaikan sistem yang sedang berjalan. ERP merupakan sistem informasi yang menghubungkan antara satu bagian dengan bagian perusahaan lainnya. Salah satu kelebihan dari sistem ERP adalah memungkinkan konfigurasi yang dapat disesuaikan dengan proses bisnis perusahaan. Sistem ERP yang akan dikembangkan pada penelitian ini berbasis *open source* yaitu Odoo dengan PostgreSQL sebagai *database*-nya. Pengembangan sistem ini menggunakan metode *Prototype*. Analisis dan perancangan yang digunakan adalah analisis dan perancangan berorientasi objek dengan *tools* pemodelan sistem yaitu menggunakan *unified modeling language* (UML). Rancangan sistem dengan menggunakan ERP diharapkan dapat diimplementasikan di dalam perusahaan dan dapat membantu menjawab permasalahan tersebut, serta membangun sistem yang terintegrasi antara Departemen *Marketing* dan Departemen Produksi.

Kata Kunci : Implementasi ERP, Sistem Penjualan Produk, *Make to order*, *Prototype*, *Unified Modeling Language*, Odoo, PostgreSQL.

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, yang senantiasa sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“Usulan Implementasi *Enterprise Resource Planning* (ERP) Berbasis *Open Source* Menggunakan Odoo Untuk Pengelolaan Data Penjualan Produk Pada PT DRA Component Persada”**. Laporan ini diajukan untuk memenuhi sebagian syarat penyelesaian Program Sarjana Terapan Program Studi Program Studi Sistem Informasi pada Politeknik STMI Jakarta.

Penulis menyadari bahwa Tugas Akhir ini tidak akan terwujud tanpa adanya bantuan, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT atas rahmat serta kemudahan yang diberikan kepada penulis dalam menyelesaikan Tugas Akhir ini.
2. Bapak Eko Ariadi dan Ibu Fuji Nusa Indah selaku orang tua penulis yang selalu memberikan dukungan secara moril, materil maupun doa yang tulus tak henti-hentinya dihayatkan untuk penulis agar selalu dimudahkan dalam penyelesaian Tugas Akhir ini.
3. Kedua kakak dan adik penulis yang memberikan semangat dan motivasi agar segera menyelesaikan Tugas Akhir ini.
4. Bapak Dr. Mustofa, S.T., M.T., selaku Direktur Politeknik STMI Jakarta.
5. Bapak Drs. Jacob Saragih, M.M., selaku Ketua Program Studi Sistem Informasi Industri Otomotif.
6. Bapak Lucky Heriyanto, S.T., M.T.I., selaku dosen pembimbing yang telah sabar dalam membimbing penulis serta memberikan arahan dan masukan dalam penyusunan Tugas Akhir ini.
7. Bapak Agung Terminanto sebagai motivator yang sangat berjasa dalam memberikan pembelajaran mengenai odoo dan arahan untuk penulis untuk menyelesaikan laporan Tugas Akhir ini.
8. Bapak Sodikun, ST., MM., MT selaku Direktur PT DRA *Component* Persada.
9. Bapak Mukhlis selaku pembimbing yang telah membantu mengarahkan dan membimbing selama Kerja Lapangan.
10. Bapak Sugeng dan seluruh pegawai di PT DRA *Component* Persada yang telah membantu memberi arahan serta informasi untuk pengumpulan data.
11. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.

12. Seluruh keluarga besar Lembaga Dakwah Kampus Forum Studi Komunikasi Mahasiswa Islam (LDK FOSKOMI) yang telah banyak memberikan bantuan dan mendukung penulis dalam menyelesaikan laporan Tugas Akhir ini.
13. Seluruh keluarga besar Himpunan Mahasiswa Sistem Informasi (HIMASIS) yang banyak membantu dan mendukung penulis.
14. Teman-teman SA01 2014 yang selalu memberi motivasi dan semangat dalam menyusun laporan kerja praktik ini.
15. Serta semua pihak yang telah banyak membantu penulis dan yang tidak bisa penulis sebutkan satu-satu.

Semoga segala bantuan yang telah diberikan kepada penulis dibalaskan oleh Allah SWT. Penulis berharap semoga Tugas Akhir ini dapat dijadikan bahan kajian, walaupun dari pemikiran ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran guna perbaikan yang bersifat membangun sangat penulis harapkan. Akhir kata, penulis berharap semoga tulisan ini dapat bermanfaat.

Jakarta, 10 Agustus 2018

Penulis

Azmi Nurul Azizah

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	ii
LEMBAR PERSETUJUAN DOSEN	iii
LEMBAR BIMBINGAN TUGAS AKHIR.....	iv
LEMBAR PERNYATAAN KEASLIAN	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN.....	xviii
 BAB I PENDAHULUAN.....	 1
1.1 Latar Belakang.....	1
1.2 Pokok Permasalahan.....	5
1.3 Tujuan Penelitian.....	5
1.4 Batasan Masalah.....	5
1.5 Manfaat Tugas Akhir.....	6
1.6 Sistematika Penulisan	6
 BAB II LANDASAN TEORI	 8
2.1 Pengertian Implementasi.....	8
2.2 Konsep Sistem.....	8
2.2.1 Pengertian Sistem.....	8
2.2.2 Karakteristik Sistem	9
2.2.3 Klasifikasi Sistem	11
2.3 Pengertian Informasi	13
2.3.1 Ciri-ciri Informasi	13

2.3.2	Tipe Informasi	13
2.4	Konsep Penjualan.....	14
2.4.1	Penjualan	15
2.4.2	Sistem Penjualan	15
2.4.2.1	Sistem Penjualan Tunai	15
2.4.2.1.1	Prosedur Penjualan Tunai.....	15
2.4.2.2	Sistem Penjualan Kredit	16
2.4.2.2.1	Prosedur Penjualan Kredit.....	16
2.5	Pengertian Produk	16
2.5.1	Klasifikasi Produk	16
2.6	Pengertian Pengelolaan	17
2.7	<i>Enterprise Resource Planning (ERP)</i>	17
2.8	<i>Open Source</i>	18
2.9	<i>System Development Life Cycle (SDLC)</i>	19
2.10	<i>Metode Prototyping</i>	20
2.11	Analisis PIECES	22
2.12	<i>System Requirement</i>	23
2.13	<i>Flowmap</i>	24
2.14	<i>Unified Modelling Language (UML)</i>	26
2.14.1	<i>Use Case Diagram</i>	28
2.14.2	<i>Use Case Description</i>	29
2.14.3	<i>Activity Diagram</i>	30
2.14.4	<i>Sequence Diagram</i>	32
2.14.5	<i>Class Diagram</i>	33
2.14.6	<i>Deployment Diagram</i>	34
2.15	<i>Windows Navigation Diagram</i>	35
2.16	<i>Database</i>	36
2.17	<i>Entity Relationship Diagram (ERD)</i>	37
2.18	Odoo.....	38
2.19	Modul <i>Sales Management</i>	39
2.20	PostgreSQL	40

2.21.1 Tipe Data PostgreSQL	41
2.21 <i>Black Box Testing</i>	45
BAB III METODOLOGI PENELITIAN	46
3.1 Metodologi Penelitian	46
3.2 Jenis dan Sumber Data	46
3.3 Metode Pengumpulan Data	47
3.4 Metode Pengembangan Sistem	48
3.5 <i>Literatur Review</i>	49
3.6 Kerangka Penelitian	51
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA.....	56
4.1 Tentang Perusahaan	56
4.2 Profil Perusahaan	57
4.3 Logo Perusahaan	58
4.4 Visi, Misi dan Nilai Perusahaan.....	59
4.4.1 Visi Perusahaan.....	59
4.4.2 Misi Perusahaan	59
4.4.3 Nilai Perusahaan.....	60
4.5 Jumlah Karyawan dan Jam Kerja.....	60
4.5.1 Jumlah Karyawan.....	60
4.5.2 Jam Kerja	60
4.6 Struktur Organisasi Perusahaan	61
4.6.1 Tugas dan Wewenang Setiap Divisi	63
4.7 Produk	65
4.8 Penjualan	66
4.9 Dokumen Sistem Penjualan Produk.....	67
4.9.1 Dokumen Masukan	68
4.9.2 Dokumen Keluaran	69
4.10 Prosedur Penjualan Produk	72
4.11 Analisis Permasalahan	74

BAB V ANALISIS DAN PEMBAHASAN.....	78
5.1 Prosedur Sistem Penjualan Produk Usulan	78
5.2 Analisis Kebutuhan Sistem Informasi	80
5.2.1 <i>Fungsional Requirement</i>	82
5.2.2 <i>Non Fungsional Requirement</i>	82
5.3 <i>Use Case Diagram</i>	83
5.3.1 <i>Use Case Description</i>	83
5.4 <i>Activity Diagram</i>	91
5.5 <i>Sequence Diagram</i>	99
5.6 <i>Class Diagram</i>	104
5.7 <i>Deployment Diagram</i>	106
5.8 <i>Window Navigation Diagram</i>	106
5.9 Pemodelan Data Sistem Usulan	107
5.10 <i>Black Box Testing</i>	109
5.11 Perancangan Sistem Usulan	109
5.12 Konfigurasi Sistem Penjualan Produk Usulan	126
 BAB IV PENUTUP	 130
6.1 Kesimpulan	130
6.2 Saran.....	130
 DAFTAR PUSTAKA	 131
LAMPIRAN	

DAFTAR GAMBAR

	Halaman
Gambar II.1	Karakteristik Sistem 11
Gambar II.2	Gambaran Sistem Terbuka 12
Gambar II.3	Gambaran Sistem Tertutup 12
Gambar II.4	Tingkatan Produk 18
Gambar II.5	Tahapan SDLC 20
Gambar II.6	Tahapan <i>Prototype</i> 21
Gambar II.7	Contoh <i>Windows Navigation Diagram</i> 36
Gambar III.1	Kerangka Penelitian 55
Gambar IV.1	Logo PT DRA Component Persada 59
Gambar IV.2	Struktur Organisasi PT DRA Component Persada 62
Gambar IV.3	Struktur Organisasi Divisi <i>Marketing</i> 62
Gambar IV.4	<i>Purchase Order</i> 68
Gambar IV.5	<i>Sale Confirmation</i> 70
Gambar IV.6	<i>Delivery Order</i> 71
Gambar IV.7	Aliran Prosedur Sistem Informasi Penjualan Produk saat ini ...73
Gambar V.1	Prosedur Sistem Penjualan Produk Usulan 79
Gambar V.2	<i>Use Case Diagram</i> Usulan Sistem Penjualan Produk 83
Gambar V.3	<i>Activity Diagram</i> Login.....92
Gambar V.4	<i>Activity Diagram</i> Konfirmasi <i>Quotation</i> 92
Gambar V.5	<i>Activity Diagram</i> Mengirimkan <i>Quotation</i> 93
Gambar V.6	<i>Activity Diagram</i> Melihat <i>Manufacturing Order</i> 93
Gambar V.7	<i>Activity Diagram</i> Mengelola Data Customer 94
Gambar V.8	<i>Activity Diagram</i> Mengelola Data Produk 95
Gambar V.9	<i>Activity Diagram</i> Mengelola data <i>Quotation</i> 96
Gambar V.10	<i>Activity Diagram</i> Mengelola <i>Delivery Order</i> 97
Gambar V.11	<i>Activity Diagram</i> Mencetak <i>Delivery Order</i> 97
Gambar V.12	<i>Activity Diagram</i> Mengelola <i>Sales Order</i> 98

Gambar V.13	<i>Sequence Diagram Login</i>	99
Gambar V.14	<i>Sequence Diagram Mengelola Data customer</i>	99
Gambar V.15	<i>Sequence Diagram Mengelola Data produk</i>	100
Gambar V.16	<i>Sequence Diagram Konfirmasi Quotation</i>	100
Gambar V.17	<i>Sequence Diagram Mengelola data Quotation</i>	101
Gambar V.18	<i>Sequence Diagram Mengirimkan Quotation</i>	101
Gambar V.19	<i>Sequence Diagram Melihat Manufacturing Order</i>	102
Gambar V.20	<i>Sequence Diagram Mencetak Delivery Order</i>	102
Gambar V.21	<i>Sequence Diagram Mengelola Delivery Order</i>	103
Gambar V.22	<i>Sequence Diagram Mengelola Sales Order</i>	104
Gambar V.23	<i>Class Diagram Sistem Penjualan Produk Usulan</i>	105
Gambar V.24	<i>Deployment Diagram Sistem Penjualan Produk Usulan</i>	106
Gambar V.25	<i>Windows Navigation Sistem Penjualan Produk Usulan</i>	107
Gambar V.26	<i>Entity Relationship Diagram Sistem Penjualan Produk</i>	108
Gambar V.27	<i>Default Login</i>	109
Gambar V.28	<i>Perancangan Login</i>	110
Gambar V.29	<i>Default Menu Utama</i>	110
Gambar V.30	<i>Perancangan Menu Utama</i>	111
Gambar V.31	<i>Submenu Customer</i>	111
Gambar V.32	<i>Submenu Product</i>	112
Gambar V.33	<i>Submenu Quotation</i>	112
Gambar V.34	<i>Submenu Sales Order</i>	113
Gambar V.35	<i>Submenu Manufacturing Order</i>	113
Gambar V.36	<i>Default form customer</i>	114
Gambar V.37	<i>Perancangan form customer</i>	114
Gambar V.38	<i>Default form product</i>	115
Gambar V.39	<i>Perancangan form product</i>	115
Gambar V.40	<i>Default form quotation</i>	116
Gambar V.41	<i>Perancangan form quotation</i>	116
Gambar V.42	<i>Form Send Quotation by Email</i>	117
Gambar V.43	<i>Send Quotation by Email</i>	117

Gambar V.44	Form <i>confirm sale</i>	118
Gambar V.45	<i>Default form sales order</i>	119
Gambar V.46	Perancangan form <i>sales order</i>	119
Gambar V.47	<i>Default form Outgoing Shipping</i>	120
Gambar V.48	Perancangan form <i>Outgoing Shipping</i>	121
Gambar V.49	<i>Default form Manufacturing Order</i>	121
Gambar V.50	Perancangan form <i>Manufacturing Order</i>	122
Gambar V.51	<i>Default produce</i>	122
Gambar V.52	Perancangan <i>produce</i>	123
Gambar V.53	<i>Default Delivery Order Report</i>	123
Gambar V.54	Perancangan <i>Delivery Order Report</i>	124
Gambar V.55	<i>Transfer barang pada sistem</i>	125
Gambar V.56	Perancangan penerimaan barang.....	125
Gambar V.57	Perancangan form <i>received to customers</i>	126
Gambar V.58	Konfigurasi Modul <i>Sales Management</i>	126
Gambar V.59	Konfigurasi Data <i>Master Product</i>	127
Gambar V.40	Konfigurasi Data <i>Master Customer</i>	128
Gambar V.41	Konfigurasi Hak Akses Staf <i>Marketing</i>	128
Gambar V.43	Konfigurasi Hak Akses <i>Group Manufacturing User</i>	129

DAFTAR TABEL

	Halaman
Tabel II.1 Simbol-simbol <i>Flow map</i>	25
Tabel II.2 <i>Structure Diagram</i>	26
Tabel II.3 <i>Behavior Diagram</i>	27
Tabel II.4 <i>Use Case Diagram</i>	27
Tabel II.5 <i>Activity Diagram</i>	31
Tabel II.6 <i>Sequence Diagram</i>	32
Tabel II.7 <i>Class Diagram</i>	33
Tabel II.8 <i>Deployment Diagram</i>	35
Tabel II.9 Simbol-Simbol <i>Entity Relationship Diagram</i>	38
Tabel II.10 Tipe Data Jenis <i>Numeric</i> pada PostgreSQL	41
Tabel II.11 Tipe Data Jenis Mata Uang pada PostgreSQL	42
Tabel II.12 Tipe Data Jenis Karakter pada PostgreSQL	42
Tabel II.13 Tipe Data Jenis Data Biner pada PostgreSQL.....	43
Tabel II.14 Tipe Data Jenis Data Boolean pada PostgreSQL	43
Tabel II.15 Tipe Data Jenis Data Geometri pada PostgreSQL	43
Tabel II.16 Tipe Data Tanggal/Waktu pada PostgreSQL	44
Tabel II.17 Tipe Data Alamat Jaringan pada PostgreSQL.....	45
Tabel III.1 <i>Literature Review</i>	49
Tabel IV.1 Jumlah Karyawan PT DRA Component Persada.....	60
Tabel IV.2 Jam Kerja <i>shift 1</i>	61
Tabel IV.3 Jam Kerja <i>shift 2</i>	61
Tabel IV.4 Jam Kerja Bagian Kantor	61
Tabel IV.5 Produk.....	66
Tabel IV.6 Diagram PIECES.....	74
Tabel V.1 Analisis Kebutuhan Sistem Informasi	80
Tabel V.2 <i>Use Case Description Login</i>	84
Tabel V.3 <i>Use Case Description Mengelola Data Customer</i>	84

Tabel V.4	<i>Use Case Description</i> Mengelola Data Produk.....	85
Tabel V.5	<i>Use Case Description</i> Mengelola data Quotation.....	86
Tabel V.6	<i>Use Case Description</i> Mengirimkan Quotation.....	87
Tabel V.7	<i>Use Case Description</i> Konfirmasi Quotation.....	88
Tabel V.8	<i>Use Case Description</i> Melihat Manufacturing Order.....	89
Tabel V.9	<i>Use Case Description</i> Mengelola Delivery Order	90
Tabel V.10	<i>Use Case Description</i> Mengelola Sales Order.....	91
Tabel V.11	<i>Use Case Description</i> Mencetak Delivery Order.....	91

BAB I

PENDAHULUAN

1.1 Latar Belakang

Suatu perusahaan tidak mampu bertahan lama tanpa adanya pemasukan yang didapat secara berkelanjutan. Keadaan ini menyebabkan banyaknya perusahaan yang terus meningkatkan jumlah produksi. Sehingga dari proses produksi tersebut menghasilkan sebuah produk yang dapat dijual lalu perusahaan mendapatkan timbal-balik berupa sejumlah uang sebesar harga produk yang telah terjual, hal ini biasa disebut kegiatan penjualan produk. Dalam aktivitas bisnis, penjualan produk memegang peranan yang penting untuk menunjang operasional perusahaan, agar proses pemenuhan kebutuhan konsumen tetap terlaksana sesuai target penjualan. Kualitas produk dan strategi penjualan yang baik adalah kunci yang harus dimiliki oleh setiap perusahaan agar dapat mendorong peningkatan penjualan. Oleh karena itu dibutuhkan manajemen sistem penjualan produk yang baik yang dapat mempermudah dalam pengambilan keputusan. Namun tidak hanya sampai bagaimana pengolahan penjualan tetapi juga bagaimana menjaga kepercayaan dan kemitraan antara perusahaan dan konsumen agar produk yang berkualitas tersampaikan secara tepat waktu sehingga dapat menjaga kelangsungan produksi dan hidup perusahaan dimasa yang akan datang.

Memasuki abad 21, hampir di semua perusahaan yang bergerak dibidang perdagangan maupun industri membutuhkan suatu teknologi informasi yang baik untuk perkembangan sistem yang ada pada perusahaan yang dari hari ke hari memerlukan kemajuan. Perkembangan teknologi informasi telah memberikan kontribusi yang cukup berarti dalam meningkatkan kegiatan usaha khususnya dalam hal pengolahan data serta dalam meningkatkan kegiatan pelayanan. Salah satunya dengan memiliki sistem informasi yang baik dan terkendali dengan alur yang jelas yaitu suatu sistem penjualan yang terkomputerisasi agar dapat memperlancar serta mempermudah dalam mengelola data transaksi penjualan. Pengelolaan data dan informasi yang baik sangat di perlukan oleh pihak manajemen

untuk mempermudah dalam proses peng-*input*-an maupun pencarian data agar menjadi lebih cepat dan tepat. Selain itu juga, teknologi informasi dapat membantu menjaga keamanan data agar data tersebut hanya dapat diakses oleh orang yang telah ditentukan saja. Teknologi informasi yang dapat digunakan salah satunya yaitu dengan menggunakan *Enterprise Resource Planning* (ERP).

ERP saat ini telah banyak dikembangkan oleh berbagai vendor dunia dengan dua jenis software ERP yang digunakan yaitu yang berbayar dan juga *open source*. Karena hal ini, menjadikan ERP sudah tidak asing lagi dan bahkan menjadi hal yang harus dimiliki oleh perusahaan. Namun pada praktiknya di Indonesia, tidak semua perusahaan menggunakan ERP dikarenakan biaya yang cukup tinggi serta tingkat kerumitan yang tinggi. Dengan adanya faktor-faktor ini membuat perusahaan menengah ke bawah memutuskan untuk belum mampu menggunakan ERP. Walaupun saat implementasi ERP selalu terkesan mahal namun sudah banyak vendor yang menawarkan solusi, salah satu nya yaitu dengan menggunakan ERP yang bersifat *open source*. ERP *open source* tentu saja bisa menjadi pilihan bagi perusahaan yang belum memiliki dana yang cukup dikarenakan biaya yang diperlukan untuk implementasi ERP *open source* lebih sedikit. Adapun contoh ERP *open source* seperti *Open Bravo*, *Compiere ERP&CRM*, *Adempiere*, *TinyERP*, *OpenERP* dan masih banyak lagi. *Compiere ERP&CRM* merupakan *Open source* ERP yang pertama dan paling besar komunitasnya, menggunakan *database Oracle* dan *Sybase* yang dapat berjalan di platform Linux/Unix maupun Windows. Secara umum fiturnya cukup lengkap, hanya saja untuk CRM dan *manufacturing* masih sangat terbatas. *Adempiere* merupakan turunan dari *Compiere* yang dikembangkan oleh komunitas, menggunakan *database PostgreSQL* maupun *Oracle*. Namun telah memisahkan diri dari *compiere* sehingga fiturnya masih sangat mirip dengan *compiere*. Kelemahannya, *adempiere* bukanlah aplikasi yang siap pakai setelah diinstall namun membutuhkan proses setup dan konfigurasi sehingga pada umumnya akan membutuhkan bantuan dari konsultan atau orang yang pernah menggunakan *adempiere* untuk melakukan setup dan konfigurasi. *Open Bravo* merupakan turunan dari *compiere* juga, memiliki fitur *Full Web Base* sehingga sangat cocok untuk perusahaan yang mempunyai banyak

cabang karena dapat mengakses sistem secara *web base*, menggunakan *database* *postgreSQL* atau *oracle* dan dapat berjalan di platform Linux maupun windows. Namun sangat disayangkan *Open Bravo* meniadakan modul CRM meskipun cukup sederhana namun fitur ini sangat bermanfaat. *TinyERP* merupakan ERP yang *simple* dan mudah dipelajari, menggunakan *database* *postgreSQL* dan dapat berjalan di platform Linux maupun windows, memiliki modul-modul yang cukup standard namun kelemahannya pada komunitasnya yang masih kecil sehingga jika ada masalah susah mencari *support*. Dan ada pula *OpenERP* yang sekarang dikenal dengan Odoo merupakan sistem manajemen yang menyediakan lebih dari 4000 aplikasi untuk menunjang kebutuhan bisnis perusahaan, tersedia 18 bahasa dan memiliki komunitas dunia serta tidak hanya digunakan pada perusahaan besar saja tetapi digunakan oleh perusahaan kecil dan independen. Odoo dapat diaplikasikan untuk berbagai macam sektor, selain itu memiliki tampilan yang *user friendly* sehingga mudah dipelajari dan cepat beradaptasi, tidak hanya dijalankan *web base* tetapi juga bisa versi *mobile*. Odoo dibangun dengan menggunakan arsitektur yang modular sehingga dapat memilih modul yang dibutuhkan. Berdasarkan alasan itu penulis memilih menggunakan Odoo pada penerapan ERP di PT DRA Component Persada.

PT DRA Component Persada merupakan perusahaan yang bergerak dalam bidang *Manufacturing Industrial Fasteners Component* yang memproduksi berbagai macam baut dengan produk *Screw*, *Bolt*, *Nut* dan *Pin*. PT DRA Component Persada melakukan produksi untuk pertama kali pada bulan Februari 2010 yang memiliki pangsa pasar elektronik dan otomotif yang dipesan dari berbagai konsumen. Komponen yang dipesan pun sangat beragam, model pesanan dan jumlahnya juga selalu berbeda untuk setiap konsumen, tergantung kebutuhan konsumen tersebut. Departemen *Marketing* PT DRA Component Persada merupakan salah satu Departemen dalam struktur organisasi PT DRA Component Persada. Departemen ini bertugas untuk mencatat semua data mengenai informasi pelanggan, mencatat produk yang akan dijual dan merencanakan penjadwalan yang berhubungan dengan penjualan produk. Sebagai perusahaan yang sedang berkembang, Departemen *Marketing* pada PT DRA Component Persada tentunya

tidak terlepas dari permasalahan-permasalahan yang terjadi. Seperti proses penawaran harga masih dilakukan melalui telepon sehingga memakan waktu, proses pengajuan permintaan produksi belum terkomputerisasi dan memakan cukup waktu lama dengan memberikan *copy*-an form PO *customer* ke Departemen PPIC dimana PO *customer* tersebut harus diolah menjadi *Job Order* sebagai permintaan produk ke Departemen Produksi untuk melakukan proses produksi serta penyimpanan dan pengelolaan data masih menggunakan *Microsoft Excel* yang belum terhubung dengan *database* sebagai media penyimpanan data penjualan produk. Selain itu, belum adanya keamanan data dalam pengelolaan data penjualan sehingga siapa saja bisa mengaksesnya.

Untuk mengatasi masalah tersebut, diperlukan pengembangan suatu aplikasi yang dapat membantu perusahaan dalam mengelola setiap permintaan produk maupun proses penjualan produk yang berguna bagi perusahaan. Diharapkan dengan aplikasi pengembangan tersebut, dapat menyelesaikan masalah yang terjadi pada Departemen *Marketing* PT DRA Component Persada. Adapun judul Tugas Akhir ini adalah “USULAN IMPLEMENTASI *ENTERPRISE RESOURCE PLANNING* (ERP) BERBASIS *OPEN SOURCE* MENGGUNAKAN ODOO UNTUK PENGELOLAAN DATA PENJUALAN PRODUK DI PT DRA COMPONENT PERSADA”.

1.2 Pokok Permasalahan

Permasalahan yang terjadi di Departemen *Marketing* pada PT DRA Component Persada adalah sebagai berikut:

1. Tidak adanya aplikasi yang menghubungkan antara aktivitas peng-*input*-an surat penawaran harga dengan *database* sehingga proses penawaran harga memakan cukup waktu.
2. Proses pengajuan permintaan produksi belum terkomputerisasi dan memakan cukup waktu dengan memberikan *copy*-an form PO *customer* ke Departemen PPIC dimana PO *customer* tersebut harus diolah menjadi *Job Order* sebagai permintaan produk ke Departemen Produksi untuk melakukan proses produksi.

3. Kurangnya keamanan data dikarenakan data yang dikelola belum mempunyai hak akses sehingga masih bisa digunakan dan diakses oleh orang lain.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian Tugas Akhir ini adalah Menerapkan sistem *Enterprise Resource Planning* (ERP) berbasis *Open Source* menggunakan Odoo yang mampu menjawab permasalahan yang terjadi di Departemen *Marketing* pada PT DRA Component Persada.

1.4 Batasan Masalah

Agar dalam penulisan Tugas Akhir ini lebih fokus dan terarah, maka perlu diberikan batasan-batasan diantaranya adalah sebagai berikut:

1. Analisis dan pengamatan dilakukan di PT DRA Component Persada pada bagian *Marketing* pada setiap hari Sabtu mulai dari 1 November 2017 sampai dengan 31 Desember 2017.
2. Penelitian dilakukan hanya sebatas menangani proses penjualan produk yang mencakup proses penawaran harga (*Quotation*), *Sales Order*, pengajuan permintaan produksi (*Manufacturing Order*) dan pembuatan surat jalan (*Delivery Order*) tetapi tidak membahas mengenai proses produksi, penagihan dan pembayaran, serta *return* produk.
3. Sistem *Enterprise Resource Planning* (ERP) yang digunakan yaitu Odoo 8.0, *database* menggunakan PostgreSQL 9.3 dan modul *Sales Management*

1.5 Manfaat Tugas Akhir

Adapun manfaat yang dapat diambil dari penelitian pada Tugas Akhir ini adalah Mampu menyelesaikan permasalahan yang terjadi di Departemen *Marketing* pada PT DRA Component Persada yaitu sistem penjualan produk yang digunakan saat ini belum terkomputerisasi yang terhubung dengan basis data untuk penyimpanan dan pengolahan data serta kurangnya keamanan data-data penjualan produk.

1.6 Sistematika Penulisan

Agar lebih mempermudah perumusan dan pemecahan masalah yang akan dibahas pada penelitian Tugas Akhir ini, maka dapat diuraikan tahapan-tahapan dalam penyusunan laporan ini sebagai berikut:

BAB I PENDAHULUAN

Bab ini memuat latar belakang, pokok permasalahan, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan laporan.

BAB II LANDASAN TEORI

Bab ini membahas tentang berbagai teori berbagai teori yang diperoleh dari buku-buku referensi maupun sumber referensi lain. Teori yang dipaparkan adalah mengenai seputar pengertian rancangan, implementasi, *Enterprise Resource Planning* (ERP), sistem, penjualan produk, analisis sistem, metode pengembangan sistem, *flowchart*, *Unified Modelling Language* (UML), Odoo dan PostgreSQL.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang langkah-langkah yang akan dilakukan dalam perumusan dan pemecahan masalah, jenis dan sumber data, metode pengumpulan data serta metodologi pengembangan sistem yang digunakan.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini akan menguraikan tentang hasil pengamatan dalam penelitian yang telah dilakukan, seperti profil perusahaan dan Departemen *Marketing* sebagai objek penelitian, proses bisnis sistem penjualan produk yang sedang berjalan, dan dokumen-dokumen yang terlibat dalam penjualan produk.

BAB V ANALISIS DAN PEMBAHASAN

Bab ini berisi tentang uraian tahapan pengembangan sistem, yang dimulai dari tahapan analisis masalah sistem, perancangan sistem

usulan menggunakan *Unified Modeling Language* (UML), yaitu *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, *entity relationship diagram*, kamus data, antarmuka dan simulasi ERP.

BAB VI PENUTUP

Bab ini diuraikan kesimpulan berdasarkan dari hasil penelitian dan saran-saran berupa masukan untuk perusahaan dalam berbagai hal yang berhubungan dengan sistem penjualan produk serta bagi penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Pengertian Implementasi

Menurut Kamus Besar Bahasa Indonesia (2016) implementasi diartikan sebagai pelaksanaan atau penerapan. Jadi implementasi adalah proses, cara, perihal mempraktikkan, perbuatan melaksanakan atau menerapkan (rancangan, keputusan, dan sebagainya).

2.2 Konsep Sistem

Sistem adalah hubungan antara unit yang satu dengan unit lainnya yang saling berhubungan satu dengan yang lainnya dan tidak dapat dipisahkan serta menuju suatu kesatuan dalam rangka mencapai tujuan yang telah ditetapkan sebelumnya. Misalnya, apabila satu unit dalam suatu perusahaan mengalami gangguan, unit yang lainnya pun akan terganggu dalam mencapai tujuan yang telah ditetapkan. Sistem terdiri dari atas unsur-unsur yang saling berhubungan dan beroperasi secara bersama untuk mencapai tujuan yang telah ditetapkan. Jadi, sistem bukanlah unsur yang tersusun secara tidak beraturan, melainkan unsur-unsur yang saling berhubungan dan bekerja sama untuk mencapai suatu tujuan (Rusdiana dan Moch Irfan, 2014)

2.2.1 Pengertian Sistem

Rusdiana dan Moch Irfan (2014) mengatakan bahwa kata sistem berasal dari bahasa Yunani, yaitu *systema* yang artinya himpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan. Selain itu, bisa diartikan sekelompok elemen yang indenpenden, namun saling terkait sebagai suatu kesatuan. Sistem terdiri atas struktur dan proses:

1. Struktur sistem merupakan unsur-unsur yang membentuk sistem tersebut.
2. Proses sistem merupakan menjelaskan cara kerja setiap unsur sistem dalam mencapai tujuan.

Setiap sistem merupakan bagian dari sistem lain yang lebih besar dan terdiri dari atas berbagai sistem yang kecil yang disebut subsistem.

Menurut Jogianto dikutip dari buku Sistem Infomasi Manajemen oleh Rusdiana dan Moch Irfan (2014), sistem adalah kumpulan dari elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian dan kesatuan yang nyata, seperti tempat, benda, serta orang-orang yang ada dan terjadi. Sedangkan menurut Murdick R.G sistem adalah seperangkat elemen yang membentuk kumpulan atau bagan-bagan pengolahan yang mencari suatu tujuan dengan mengoperasikan data dan/atau barang pada waktu tertentu untuk menghasilkan informasi.

2.2.2 Karakteristik Sistem

Menurut Edhi Sutanta yang dikutip oleh Rusdiana dan Moch Irfan (2014), karakteristik sistem yaitu sebagai berikut:

1. *Komponen (Component)*

Komponen sistem adalah segala sesuatu yang menjadi bagian penyusunan sistem. Komponen sistem dapat berupa benda nyata ataupun abstrak. Komponen sistem disebut sebagai subsistem.

2. *Batas (Boundary)*

Batas sistem diperlukan untuk membedakan suatu sistem dengan sistem yang lain. Tanpa adanya batas sistem, sangat sulit untuk memberikan batasan *scope* tinjauan terhadap sistem.

3. *Lingkungan (Environments)*

Lingkungan sistem adalah segala sesuatu yang berada diluar sistem lingkungan sistem yang dapat menguntungkan ataupun merugikan. Umumnya lingkungan yang menguntungkan akan selalu dipertahankan untuk menjaga keberlangsungan sistem, sedangkan lingkungan sistem yang merugikan akan diupayakan agar mempunyai pengaruh seminimal mungkin, bahkan ditiadakan.

4. Penghubung/antarmuka (*Interface*)

Penghubung/antarmuka merupakan sarana memungkinkan setiap komponen sistem, yaitu segala sesuatu yang bertugas menjembatani hubungan antarkomponen dalam sistem. Penghubung/antarmuka merupakan sarana setiap komponen saling berinteraksi dan berkomunikasi.

5. Masukan (*Input*)

Masukan merupakan komponen sistem, yaitu segala sesuatu yang perlu dimasukan ke dalam sistem sebagai bahan yang akan diolah lebih lanjut untuk menghasilkan keluaran (*output*) yang berguna.

6. Pengolahan (*Processing*)

Pengolahan merupakan komponen sistem yang mempunyai peran utama mengolah masukan agar menghasilkan *output* yang berguna bagi pemakainya.

7. Keluaran (*Output*)

Keluaran merupakan komponen sistem yang berupa berbagai macam bentuk keluaran yang dihasilkan oleh komponen pengolahan.

8. Sasaran (*Objectives*) dan tujuan (*Goal*)

Setiap komponen dalam sistem perlu dijaga agar saling bekerja sama agar mampu mencapai sasaran dan tujuan sistem.

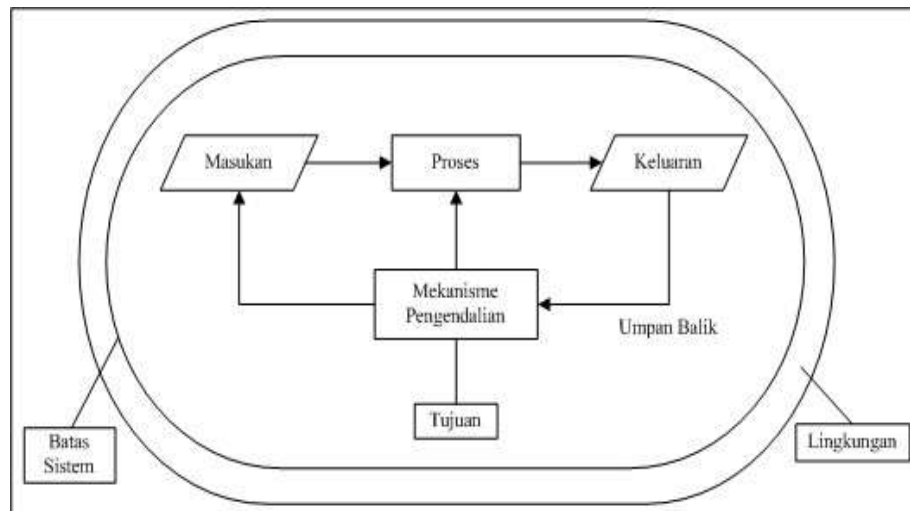
9. Kendali (*Control*)

Setiap komponen dalam sistem perlu dijaga agar tetap bekerja sesuai dengan peran dan fungsinya masing-masing.

10. Umpan balik (*Feedback*)

Umpan balik diperlukan oleh bagian kendali (*control*) sistem untuk mengecek terjadinya penyimpanan proses dalam sistem dan mengembalikannya pada kondisi normal.

Selain itu menurut Tata Sutabri yang dikutip oleh Rusdiana dan Moch Irfan (2014), sebuah sistem memiliki karakteristik tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Untuk lebih jelasnya dapat dilihat pada Gambar II.1 berikut:



Gambar II.1 Karakteristik Sistem

(Sumber: Rusdiana dan Moch Irfan, 2014)

2.2.3 Klasifikasi Sistem

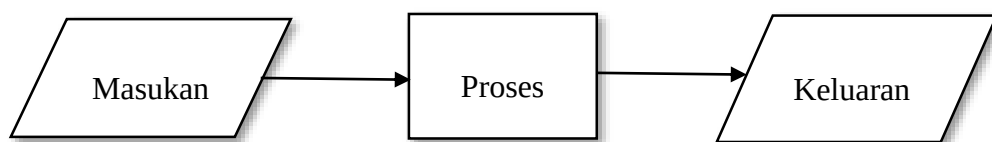
Menurut Rusdiana dan Moch Irfan (2014), sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya sebagai berikut:

1. Sistem diklasifikasikan sebagai sistem abstrak dan sistem fisik. Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik. Misalnya sistem komputer, sistem akuntansi, sistem produksi, dan sebagainya.
2. Sistem diklasifikasikan sebagai sistem alamiah dan sistem buatan manusia. Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia yang melibatkan interaksi antara manusia dan mesin disebut *human-machine system*. Misalnya sistem informasi akuntansi karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.
3. Sistem diklasifikasikan sebagai sistem tertentu dan sistem tidak tertentu. sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Misalnya sistem komputer yang tingkah lakunya dapat dipastikan berdasarkan program-

program yang dijalankan. Sistem tidak tertentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

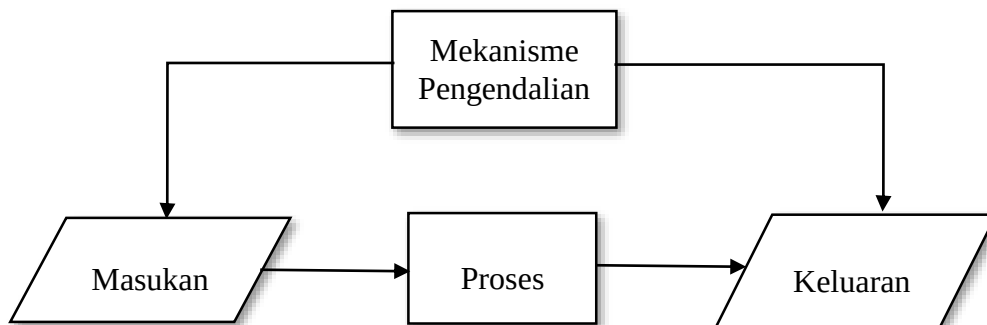
4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka. Sistem tertutup merupakan sistem yang tidak berhubungan dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya campur tangan dari pihak luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem lainnya.

Menurut McLeod yang dikutip oleh Rusdiana dan Moch Irfan (2014). Jenis sistem secara umum terdiri atas sistem terbuka dan sistem tertutup. Sistem terbuka adalah sistem yang tidak memiliki sasaran, pengendalian mekanis dan umpan balik. Adapun sistem yang tertutup adalah sebuah sistem yang memiliki sasaran, pengendalian mekanis dan umpan balik. Secara sederhana, sistem terbuka dapat digambarkan sebagai berikut:



Gambar II.2 Gambaran Sistem Terbuka
(Sumber: Rusdiana dan Irfan, 2014)

Sedangkan sistem tertutup adalah sistem yang tidak berinteraksi secara langsung dengan lingkungannya melalui arus sumber daya. Skema sistem tertutup dapat dilihat pada gambar berikut ini:



Gambar II.3 Gambaran Sistem Tertutup
(Sumber: Rusdiana dan Irfan, 2014)

2.3 Pengertian Informasi

Rusdiana dan Moch Irfan (2014) mengatakan bahwa informasi atau dalam bahasa Inggrisnya adalah *information*, berasal dari kata *informacion* bahasa Perancis. Kata tersebut diambil dari bahasa Latin, yaitu "*informationem*" yang artinya "konsep, ide, garis besar". Informasi adalah suatu data yang sudah diolah atau diproses sehingga menjadi suatu bentuk yang memiliki arti bagi penerima informasi yang memiliki nilai bermanfaat. Sedangkan menurut Gordon B. Davis dikutip oleh Rusdiana dan Moch Irfan (2014), informasi dapat mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi dan sebagainya. Sedangkan menurut Samuel Eilon, bahwa informasi adalah sebagai pernyataan yang menjelaskan suatu peristiwa atau suatu objek atau suatu konsep, sedemikian rupa sehingga membantu kita untuk membedakan dari yang lain.

2.3.1 Ciri-ciri Informasi

Menurut McLeod dikutip dari buku Sistem Informasi Akuntansi oleh Azhar Susanto (2017), suatu informasi berkualitas harus memiliki ciri-ciri sebagai berikut:

1. Akurat artinya informasi harus mencerminkan keadaan yang sebenarnya. Pengujian akurasi dilakukan oleh dua orang atau lebih yang berbeda, apabila pengujian tersebut menghasilkan hasil yang sama maka dianggap data tersebut akurat.
2. Tepat waktu artinya informasi harus tersedia atau ada pada saat informasi tersebut diperlukan, tidak besok atau tidak beberapa jam lagi
3. Relevan artinya informasi yang diberikan harus sesuai dengan yang dibutuhkan oleh individu yang ada diberbagai tingkatan dan bagian dalam organisasi.
4. Lengkap artinya informasi harus diberikan secara lengkap. Misalnya informasi tentang penjualan tidak ada bulannya atau tidak ada data fakturnya.

2.3.2 Tipe Informasi

Menurut Jogiyanto dikutip dari buku Sistem Informasi Akuntansi Edisi III oleh Agustinus Mulyono (2015), tipe informasi akan berbeda untuk tingkatan manajemen yang berbeda:

1. Informasi pengumpulan data (*scorekeeping information*). Informasi ini berguna berguna bagi manajer bawah untuk mengevaluasi kinerja personil-personilnya.
2. Informasi pengarah perhatian (*attention directing information*). Merupakan informasi untuk membantu manajemen memusatkan perhatian pada masalah-masalah yang menyimpang, ketidakberesan, ketidakefisienan dan kesempatan-kesempatan yang dapat dilakukan. Informasi ini membantu manajer menengah.
3. Informasi pemecah masalah (*problem solving information*). Merupakan informasi untuk membantu manajer atas pengambil keputusan memecahkan permasalahan yang dihadapinya. Informasi ini digunakan untuk membantu analisa yang dilakukan manajemen tingkat atas.

2.4 Penjualan

Danang Sunyoto (2012) mengatakan bahwa penjualan merupakan salah satu kegiatan pemasaran. Adanya penjualan dapat terciptakan suatu proses pertukaran barang atau jasa antara penjual dengan pembeli. Didalam perekonomian kita atau ekonomi uang, seseorang yang menjual sesuatu akan mendapatkan imbalan berupa uang. Dengan alat penukar berupa uang, orang akan lebih mudah memenuhi segala keinginannya dan penjualan menjadi lebih mudah dilakukan. Jarak yang jauh tidak menjadi masalah bagi penjual. Sedangkan menurut Wiratna (2015) Penjualan adalah suatu sistem kegiatan pokok perusahaan untuk memperjual-belian barang dan jasa yang perusahaan hasilkan.

2.4.1 Sistem Penjualan

Menurut Wiratna (2015) dalam bukunya Sistem Akuntansi, sistem penjualan terdapat dua macam, yaitu sistem penjualan tunai dan sistem penjualan kredit:

2.4.1.1 Sistem Penjualan Tunai

Sistem penjualan tunai merupakan sistem yang diberlakukan oleh perusahaan dalam menjual barang dengan cara mewajibkan pembeli untuk melakukan pembayaran harga terlebih dahulu sebelum barang diserahkan pada

pembeli. Setelah pembeli melakukan pembayaran, baru barang diserahkan, kemudian transaksi penjualan dicatat.

2.4.1.1.1 Prosedur Penjualan Tunai

Prosedur yang membentuk sistem dalam sistem penjualan tunai adalah sebagai berikut:

1. Prosedur order penjualan. Dalam prosedur ini, bagian penjualan menerima order dari pembeli dan membuat faktur penjualan tunai. Setelah pembeli membayar, bagian gudang mengirimkan barang kepada pembeli.
2. Prosedur penerimaan kas. Dalam prosedur ini, bagian kasa menerima pembayaran dari pembeli dan memberikan tanda pembayaran (berupa pita register kas dan cap “Lunas” pada faktur penjualan tunai), kemudian pembeli mengambil barang.
3. Prosedur pembungkusan dan penyerahan barang. Dalam prosedur ini pengiriman hanya menyerahkan barang kepada pembeli
4. Prosedur pencatatan penjualan tunai. Dalam prosedur ini, bagian akuntansi melakukan pencatatan transaksi penjualan tunai.

2.4.1.2 Sistem Penjualan Kredit

Sistem penjualan kredit merupakan sistem penjualan dimana pembayarannya dilakukan setelah barang diterima pembeli. Jumlah dan jatuh tempo pembayarannya disepakati oleh kedua pihak.

2.4.1.2.1 Prosedur penjualan kredit

Prosedur yang membentuk sistem dalam sistem penjualan kredit adalah sebagai berikut:

1. Prosedur order penjualan. Dalam prosedur ini, fungsi penjualan menerima order dari pembeli.
2. Prosedur persetujuan kredit. Dalam prosedur ini bagian penjualan meminta persetujuan kredit pada bagian kredit yaitu bagian keuangan.
3. Prosedur pengiriman. Dalam prosedur ini, bagian pengiriman mengirimkan barang pada pembeli sesuai surat order pengiriman

4. Prosedur faktur/penagihan. Dalam prosedur ini, bagian penagihan membuat faktur penjualan dan dikirim pada pembeli
5. Prosedur pencatatan akuntansi. Dalam prosedur ini, bagian akuntansi membuat kartu piutang berdasarkan faktur penjualan.

2.5 Pengertian Produk

Berikut ini merupakan pengertian produk menurut para ahli dikutip dari buku Dasar-Dasar Manajemen Pemasaran oleh Danang Sunyoto (2012):

1. Philip Kotler mendefinisikan produk sebagai berikut: *a product is anything that be offered to a market for attention acquisition, use or consumption that might satisfy a want or need* (Sebuah produk adalah sesuatu yang dapat ditawarkan ke pasar untuk diperhatikan, dimiliki, dipakai atau dikonsumsi sehingga dapat memuaskan keinginan atau kebutuhan). Produk bisa berupa benda fisik, jasa, orang, tempat, organisasi dan gagasan (ide).
2. Menurut Indriyo Gitosudarmo, produk adalah segala sesuatu yang diharapkan dapat memenuhi kebutuhan manusia ataupun organisasi. Suatu perusahaan seringkali menjual atau memasarkan tidak hanya satu produk saja tetapi bermacam produk yang dipasarkannya.
3. Menurut Irawan, produk adalah sesuatu yang ditawarkan dan dapat memuaskan keinginan dan kebutuhan konsumen.

2.5.1 Klasifikasi Produk

Menurut Danang Sunyoto (2012), produk berdasarkan daya tahannya diklasifikasikan menjadi tiga kelompok, yaitu:

1. Barang yang tahan lama (durable goods) adalah merupakan barang nyata yang biasanya melayani banyak kegunaan, misalnya pakaian, peralatan otomotif, komputer, peralatan bengkel, lemari es dan sebagainya.
2. Barang yang tidak tahan lama
Barang yang tidak tahan lama (nondurable goods) adalah merupakan barang nyata yang biasanya dikonsumsi untuk satu atau beberapa kegunaan, misalnya pasta gigi, kuliner, minuman energi, obat generik dan lainnya.
3. Jasa

Jasa adalah merupakan kegiatan, manfaat atau kegunaan yang ditawarkan untuk dijual, misalnya bengkel sepeda motor, reparasi komputer dan televisi, laundry, jasa angkutan barang, jasa olah data, rental mobil dan sepeda motor, kursus bahasa asing, kursus program komputer dan lainnya.

2.6 Pengertian Pengelolaan

Pengelolaan berasal dari kata kelola dan merupakan terjemahan dari kata “*management*” (Bahasa Inggris). Istilah Inggris tersebut kemudian menjadi Manajemen atau menejemen. Menurut Kamus Besar Bahasa Indonesia kata pengelolaan mempunyai 4 pengertian, yaitu:

1. Pengelolaan adalah proses atau cara perbuatan mengelola;
2. Pengelolaan adalah proses melakukan kegiatan tertentu dengan menggerakkan tenaga orang lain;
3. Pengelolaan adalah proses yang membantu merumuskan kebijakan dan tujuan organisasi;
4. Pengelolaan adalah proses yang memberikan pengawasan pada semua hal yang terlibat dalam pelaksanaan dan pencapaian tujuan.

2.7 Enterprise Resource Planning (ERP)

Menurut Mc.Leod dikutip dari buku Teknologi Informasi untuk Bisnis dan Akuntansi oleh Azhar Susanto (2017) menyatakan bahwa sistem ERP sebagai sistem berbasis komputer yang memungkinkan pengelolaan sumber daya perusahaan berdasarkan kepada organisasi yang besar. Menurut Gelinas dan Sutton mendefinisikan sistem ERP sebagai paket *software* terintegrasi yang dirancang untuk mendukung integrasi yang lengkap dalam sistem pengolahan data yang terkait pada organisasi tersebut. Sistem ini dibangun berdasarkan aktivitas-aktivitas keuangan dan non keuangan yang diperlukan untuk mendukung proses pengambilan keputusan. Beberapa contoh aplikasi ERP berbayar, seperti SAP, PeopleSoft, Oracle ERP dan JDE. Sedangkan aplikasi ERP yang *open source* contohnya adalah Adempiere, OpenBravo, Opentaps, Odoo dan WebERP (Santosa, 2010).

Menurut Rusdiana dan Moch Irfan (2014) *Enterprise Resources Planning* (ERP) adalah suatu perangkat lunak paket dengan aplikasi yang terintegrasi untuk digunakan secara luas dalam organisasi.

1. Aplikasi *Enterprise Resources Planning* (ERP) meliputi:
 - a. Fungsi akuntansi, meliputi modul-modul seperti buku besar, piutang dagang, dan utang dagang
 - b. Fungsi keuangan, meliputi modul-modul analisis portofolio, analisis resiko, dan analisis kredit
 - c. Fungsi sumber daya manusia, meliputi modul-modul rekrutmen, penggajian, dan manajemen personal.
 - d. Fungsi pemasaran, meliputi manajemen relasi pelanggan, pemasukan order, dan pemrosesan order.
 - e. Fungsi logistik, meliputi perencanaan produksi, manajemen material, dan manajemen pabrik.

2.8 Open Source

Sumber terbuka ([bahasa Inggris](#): *open source*) adalah sistem pengembangan yang tidak dikoordinasi oleh suatu individu / lembaga pusat, tetapi oleh para pelaku yang bekerja sama dengan memanfaatkan kode sumber (*source-code*) yang tersebar dan tersedia bebas (biasanya menggunakan fasilitas komunikasi [internet](#)). Pola pengembangan ini mengambil model ala *bazaar*, sehingga pola Open Source ini memiliki ciri bagi komunitasnya yaitu adanya dorongan yang bersumber dari *budaya memberi*, yang artinya ketika suatu komunitas menggunakan sebuah program Open Source dan telah menerima sebuah manfaat kemudian akan termotivasi untuk menimbulkan sebuah pertanyaan apa yang bisa pengguna berikan balik kepada orang banyak ([wikipedia.org/wiki/ Sumber_terbuka](http://wikipedia.org/wiki/Sumber_terbuka), 2018).

Menurut Tennant dikutip dari buku Metodologi Pengembangan Sistem Informasi oleh Samiaji Sarosa (2017) Sejarah perangkat lunak *open source* sering dikaitkan dengan gerakan yang dimulai oleh Richard Stallman. Richard Stallman sebagai pengembang aplikasi merasa praktik lisensi oleh banyak perusahaan besar tidak adil untuk konsumen. Konsumen dibebani harga beli yang besar demi meraup

keuntungan besar. Richard Stallman kemudian memulai *GNU Project* dan menjadi motor pergerakan *Free and Open Source Software* (FOSS). FOSS berarti suatu aplikasi bersifat gratis untuk digunakan siapa saja dan disediakan kode sumber (*source code*) untuk dimodifikasi oleh siapa saja. Secara spesifik lisensi untuk FOSS dijabarkan dalam *GNU Public License* (GPL). Semua aplikasi yang dirilis menggunakan lisensi GPL bebas dan gratis untuk digunakan, dibagikan dan dimodifikasi. GPL menganut paham *copyleft* dan bukan *copyright*. Artinya adalah modifikasi atau turunan aplikasi berlisensi GPL juga harus menganut lisensi GPL.

2.9 System Development Life Cycle (SDLC)

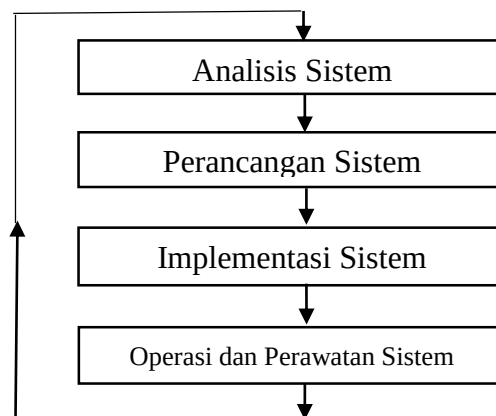
Menurut Azhar Susanto (2017) *System Development Life Cycle* adalah tahapan-tahapan pekerjaan yang dilakukan analisis sistem informasi dan programmer dalam membangun sistem informasi. SDLC juga merupakan alat untuk manajemen proyek yang bisa digunakan untuk merencanakan, memutuskan dan mengontrol proses pengembangan sistem informasi. Metode SDLC ini seringkali dinamakan juga sebagai proses pemecahan masalah, yang langkah-langkahnya meliputi:

1. Melakukan survei dan menilai kelayakan proyek pengembangan sistem informasi
2. Mempelajari dan menganalisis sistem informasi yang sedang berjalan
3. Menentukan permintaan pemakai sistem
4. Memilih solusi atau pemecahan masalah yang paling baik
5. Menentukan perangkat keras (*hardware*) dan perangkat lunak (*software*) komputer
6. Merancang sistem informasi baru
7. Membangun sistem informasi baru
8. Mengomunikasikan dan mengimplementasikan sistem informasi baru
9. Memelihara dan melakukan perbaikan/peningkatan sistem informasi baru bila diperlukan.

Menurut Hartono dikutip dari buku *Analisis dan Perancangan Sistem* oleh Mujilan (2013), metode siklus hidup pengembangan sistem (*system development*

life cycle / SDLC) memiliki beberapa tahapan. Disebut siklus karena pengembangan sistem selanjutnya dapat dimulai lagi dari awal tahap sampai dengan tahap akhir. Tahapan-tahapan tersebut dapat meliputi pula sub-sub kegiatan, yaitu:

1. Analisis Sistem
 - a. Studi pendahuluan
 - b. Studi kelayakan
 - c. Mengidentifikasi permasalahan dan kebutuhan pemakai
 - d. Memahami sistem yang ada
2. Perancangan sistem
 - a. Perancangan awal
 - b. Perancangan rinci
3. Implementasi sistem
4. Operasi dan perawatan sistem

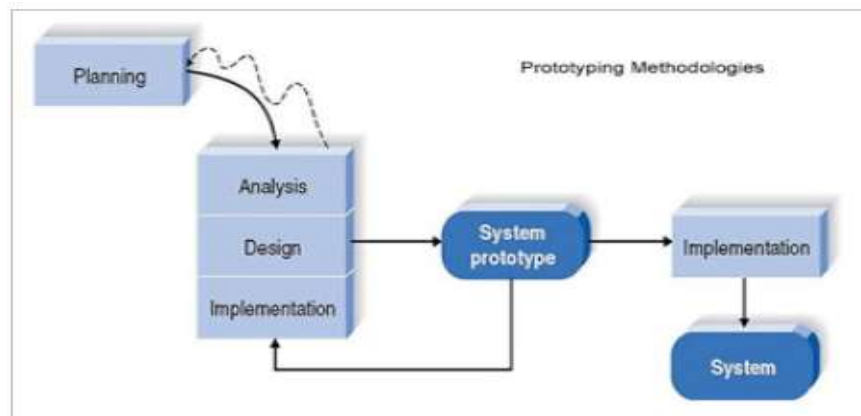


Gambar II.5 Tahapan SDLC

(Sumber: Mujilan, 2013)

2.10 Metode *Prototyping*

Metode *Prototyping* merupakan salah satu dari *Rapid Application Development (RAD) Methodologies* yang melakukan tahapan analisis, desain, dan implementasi secara bersama dan tiga tahapan tersebut dilakukan secara berulang dalam satu siklus sampai sistem selesai dibangun yang dapat dilihat pada gambar II.7:



Gambar II.6 Tahapan *Prototype*

(Sumber: Dennis, 2010)

Tahapan – tahapan dalam metode *prototyping*:

1. Fase Perencanaan (*Planning*)

Dalam tahap ini, pengembang menentukan aspek-aspek perangkat lunak yang akan dibuat, yaitu meliputi konten dan alur kerja perangkat lunak.

2. Fase Analisis (*Analysis*)

Dalam fase ini akan menentukan kebutuhan data, proses data dan informasi. Tahap analisis menjawab pertanyaan tentang siapa yang menggunakan sistem, sistem apa yang akan dilakukan, di mana dan kapan sistem tersebut akan digunakan.

3. Fase Desain (*Design*)

Pada fase ini akan diputuskan bagaimana sistem akan beroperasi, dalam hal perangkat keras, perangkat lunak, dan infrastruktur jaringan, antarmuka pengguna, formulir, dan laporan yang akan digunakan, dan program khusus, *database*, dan file yang akan diperlukan. Adapun pembangunan sistem dilakukan sesuai dengan perancangan sistem yang dibuat berdasarkan analisis kebutuhan pengguna. Pada tahap ini, hasil perancangan direalisasikan dalam bentuk media perangkat lunak yang siap diimplementasikan lingkungan kerja.

4. Fase Implementasi (*Implementation*)

Fase implementasi akan menunjukkan bagaimana sistem akan dikembangkan dan diuji untuk mengetahui apakah sistem dapat bekerja dengan sistem yang

telah ditentukan. Fase ini adalah fase paling panjang diantara fase-fase lainnya dalam proses pengembangan.

Dengan metodologi ini, dasar-dasar analisis dan desain dilakukan dan segera bekerja pada sistem *prototype*. *Prototype* pertama memiliki fitur yang sedikit dan biasanya merupakan bagian paling awal dari sistem yang akan digunakan, *prototype* ini akan ditampilkan kepada user yang akan memberikan komentarnya atas pengalaman dari pemakaian *prototype* tersebut, komentar ini selanjutnya akan dimanfaatkan untuk melakukan fase re-analisis, re-desain, dan re-implementasi untuk *prototype* kedua, proses ini akan terus berulang sampai analis, user, dan sponsor setuju dengan *prototype* yang dibuat telah menyediakan fungsionalitas yang cukup untuk dilakukan instalasi pada sebuah organisasi.

2.11 Analisis PIECES

Menurut Wukil Ragil (2010), metode PIECES adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain adalah kinerja, informasi, ekonomi, pengendalian, efisiensi, dan pelayanan. Analisis ini disebut dengan *PIECES Analysis (Performance, Information, Economy, Control, Efficiency, and Service)*.

Analisis PIECES ini sangat penting untuk dilakukan sebelum mengembangkan sebuah sistem informasi karena dalam analisis ini biasanya akan ditemukan beberapa masalah utama maupun masalah yang bersifat gejala dari masalah utama. Metode ini menggunakan enam variabel evaluasi, yaitu:

1. *Performance* (kinerja)

Memiliki peran penting untuk menilai apakah proses atau prosedur yang ada masih mungkin ditingkatkan kinerjanya, dan melihat sejauh mana dan seberapa handal suatu sistem informasi dalam berproses untuk menghasilkan tujuan yang diinginkan.

2. *Information* (informasi)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki sehingga kualitas informasi yang dihasilkan menjadi semakin baik. Informasi yang

disajikan harus benar-benar mempunyai nilai yang berguna. Hal ini dapat diukur dengan:

- a. Keluaran (*output*): Suatu sistem dalam memproduksi keluaran.
- b. Masukan (*input*): Dalam memasukkan suatu data sehingga kemudian diolah untuk menjadi informasi yang berguna.

3. *Economics* (ekonomi)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan manfaatnya (nilai gunanya) atau diturunkan biaya penyelenggaraannya.

4. *Control* (pengendalian)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian menjadi semakin baik, dan kemampuannya untuk mendeteksi kesalahan / kecurangan menjadi semakin baik pula.

5. *Efficiency* (efisiensi)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki, sehingga tercapai peningkatan efisiensi operasi, dan harus lebih unggul dari pada sistem *manual*.

6. *Service* (layanan)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki kemampuannya untuk mencapai peningkatan kualitas layanan. Buatlah kualitas layanan yang sangat *user friendly* untuk *user* (pengguna) sehingga pengguna mendapatkan kualitas layanan yang baik.

2.12 *System Requirement*

System requirement atau kebutuhan sistem adalah semua aktivitas yang harus dilakukan atau didukung oleh sistem baru dan batasan-batasan yang harus dicapai sistem baru (Satzinger, 2012). *System requirement* dibagi menjadi dua kategori yaitu:

1. *Functional requirements* atau kebutuhan fungsional.

Functional requirements merupakan penjelasan tentang layanan yang perlu disediakan oleh sistem, bagaimana sistem menerima dan mengolah masukan, dan bagaimana sistem mengatasi situasi-situasi tertentu. Selain itu kadang-kadang juga

secara jelas menentukan apa yang tidak dikerjakan oleh sistem. *Functional requirements* menggambarkan *system requirement* secara detil seperti *input*, *output*, dan pengecualian yang berlaku. Contoh: Sistem mampu menginput dan menampilkan data *customer*, data *PO customer*.

2. *Non-functional requirements* atau kebutuhan non fungsional

Non-functional requirements secara umum berisi batasan-batasan pada pelayanan atau fungsi yang disediakan oleh sistem. Termasuk didalamnya adalah batasan waktu, batasan waktu proses pembangunan, standar-standar tertentu. Karena berkaitan dengan kebutuhan sistem secara keseluruhan, maka kegagalan memenuhi kebutuhan jenis ini berakibat pada sistem secara keseluruhan. Contoh: Hak akses admin divisi *marketing* dapat melakukan semua kegiatan pada sistem sementara hak ases untuk kepala divisi *marketing* hanya dapat bisa mengontrol dan melihat kegiatan tersebut.

2.13 *Flowmap*

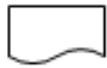
Flowmap adalah campuran peta dan *flowchart* yang menunjukkan pergerakan benda dari suatu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, jumlah barang yang diperdagangkan, atau jumlah paket dalam jaringan. *Flowmap* menolong analis dan *programmer* untuk memecahkan masalah kedalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif lain dalam pengoperasian (Jogiyanto, 2010). Adapun pedoman-pedoman dalam pembuatan *flowmap* adalah sebagai berikut:

1. *Flowmap* sebaiknya digambarkan dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam *flowmap* harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir.
4. Masing-masing kegiatan didalam *flowmap* sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan didalam *flowmap* harus didalam urutan yang semestinya.

6. Kegiatan yang terpotong dan akan disambung ditempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakan simbol simbol *flowmap* yang standar.

Adapun simbol-simbol yang sering digunakan dalam *flowmap* dapat dilihat pada Tabel II.1 berikut ini:

Tabel II.1 Simbol-simbol *Flowmap*

No	Gambar	Nama	Keterangan
1.		Mulai/Akhir	Menunjukkan dimulainya/akhir dari sebuah proses
2.		Proses	Menunjukkan kegiatan proses dari operasi program komputer
3.		Kegiatan Manual	Menunjukkan kegiatan manual
4.		Simpanan	<i>File</i> komputer yang diarsip urut
5.		Garis Alir	Menunjukkan arus dari proses
6.		Keputusan	Keputusan dalam suatu program
7.		<i>Keyboard</i>	Menunjukkan <i>input</i> yang menggunakan <i>on-line keyboard</i>
8.		Dokumen	Menunjukkan dokumen <i>input/output</i> baik untuk proses manual, mekanik, atau komputer

(Sumber Jogiyanto, 2010)

2.14 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Tujuan dari UML adalah untuk menyediakan kosa kata yang umum dari istilah-istilah berbasis objek dan teknik yang cukup banyak untuk memodelkan proyek pengembangan sistem dari analisis ke desain.

Menurut Dennis (2010), diagram-diagram yang ada dalam UML terbagi menjadi dua bagian utama yaitu, *Structure Diagram* dan *Behavior Diagram*. *Structure Diagram* biasanya digunakan untuk merepresentasikan data dan hubungan statik yang ada di dalam sebuah sistem informasi. Sedangkan *Behavior Diagram* menyediakan para analis dengan sebuah gambaran hubungan yang dinamis antara instansi atau objek yang merepresentasikan sistem informasi bisnis. Berikut adalah tabel dari jenis diagram yang ada dan fungsi-fungsinya:

1. *Structure Diagram*

Structure Diagram dapat dilihat pada Tabel II.2 berikut:

Tabel II.2 *Structure Diagram*

Nama Diagram	Fungsi
<i>Class Diagram</i>	Menggambarkan hubungan antara pemodelan-pemodelan <i>class</i> di dalam sistem.
<i>Object Diagram</i>	Menggambarkan hubungan antara pemodelan-pemodelan objek di dalam sistem.
<i>Package Diagram</i>	Mengelompokkan elemen UML lainnya menjadi satu untuk membentuk tingkat konstruksi yang lebih tinggi.
<i>Component Diagram</i>	Menggambarkan hubungan fisik di antara komponen-komponen <i>software</i> .

Tabel II.2 *Structure Diagram* (Lanjutan)

Nama Diagram	Fungsi
--------------	--------

<i>Composite Diagram</i>	Menggambarkan struktur internal dari suatu <i>class</i> dan hubungan di antara bagian-bagian dari suatu <i>class</i> .
<i>Deployment Diagram</i>	Menampilkan arsitektur fisik dari suatu sistem. Bisa juga digunakan untuk menunjukkan komponen <i>software</i> yang sedang dijadikan arsitektur fisik suatu sistem.

(Sumber: Dennis, 2010)

2. Behavior Diagram

Behavior Diagram dapat dilihat pada Tabel II.3 berikut:

Tabel II.3 Behavior Diagram

Nama Diagram	Fungsi
<i>Activity Diagram</i>	Menggambarkan proses bisnis masing-masing <i>class</i> .
<i>Interaction Overview Diagram</i>	Menggambarkan suatu ikhtisar alur dari kontrol suatu proses.
<i>Sequence Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam aktivitas berdasarkan urutan waktu
<i>Communication Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam komunikasi antara satu set dari kolaborasi objek dari suatu objek.
<i>Timing Diagram</i>	Menggambarkan interaksi yang terjadi di antara suatu set dari objek-objek dan perubahan keadaan selama perjalanan waktu.
<i>Behavioral State Machine Diagram</i>	Memeriksa kebiasaan dari suatu <i>class</i> .

Tabel II.3 Behavior Diagram (Lanjutan)

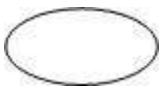
Nama Diagram	Fungsi
<i>Use Case Diagram</i>	Menangkap kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi di antara sistem dan lingkungannya.

<i>Protocol State Machine Diagram</i>	Menggambarkan ketergantungan di antara perbedaan-perbedaan <i>interface</i> dari suatu <i>class</i> .
---------------------------------------	---

(Sumber: Dennis, 2010)

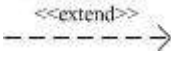
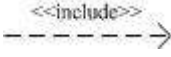
2.14.1 Use Case Diagram

Use Case Diagram dibuat untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Suatu landasan pola pikir yang ditekankan dalam diagram *use case* ini adalah “apa” yang dapat diperbuat oleh sistem, dan bukan “bagaimana” sistem melakukannya. *Use Case Diagram* menggambarkan kebutuhan sistem dari sudut pandang *user*, yaitu proses yang dilakukan oleh sistem dalam melayani *user* yang berinteraksi dengan sistem tersebut. Diagram ini sangat tepat untuk menggambarkan hubungan yang terjadi antara sistem dengan penggunaanya. Simbol-simbol *use case diagram* dapat dilihat pada Tabel II.4 berikut:

Simbol	Nama	Deskripsi
	<i>Use Case</i>	• Merepresentasikan bagian utama dari sistem secara fungsional. • Diletakkan di dalam <i>system boundary</i>. • Dilabelkan dengan frasa kata kerja deskriptif.
	<i>Actor</i>	• Seseorang atau sistem yang mendapatkan keuntungan dari sistem. • Diletakkan di luar batas sistem.
	<i>Association</i>	• Menghubungkan suatu aktor dengan <i>use case</i>.

Tabel II.4 *Use Case Diagram* (Lanjutan)

Simbol	Nama	Deskripsi
	<i>System Boundary</i>	• Nama dari sistem. • Merepresentasikan ruang lingkup sistem.

	<i>Generalization</i>	Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
	<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan ini.
	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Tabel II.4 *Use Case Diagram*

(Sumber: Dennis, 2010)

2.14.2 *Use Case Description*

Use Case Description merupakan deskripsi yang mencatat mengenai detail pemrosesan dari suatu *use case* (Satzingeretal, 2010). *Use Case Description* memiliki urutan yang lengkap dari tahapan-tahapan untuk menyelesaikan suatu proses bisnis.

Menurut Dennis (2010), elemen yang dimiliki *Use Case Description* adalah sebagai berikut;

1. *Basic Information*

a. Nama

Nama digunakan untuk menjelaskan *use case* dan membedakan *use case* satu dengan yang lainnya. Nama yang digunakan harus sederhana namun mampu menjelaskan kondisi yang memungkinkan.

b. *Description*

Description berarti uraian singkat untuk menyampaikan tujuan dari *use case*.

c. Aktor

Aktor merujuk kepada orang, sistem, perangkat lunak ataupun perangkat keras yang berinteraksi langsung dengan sistem untuk mencapai tujuan.

2. *Normal Courses*

Bagian utama dari *use case* berikutnya adalah *normal courses*. Pada *normal courses* menjelaskan mengenai langkah-langkah yang dilakukan untuk mengeksekusi respon terhadap masukan dan keluaran.

3. *Alternative Courses*

Untuk beberapa *use case* memiliki jalur alternatif pada setiap kasusnya. Pada penggambaran jalur alternatif ini terdapat cabang yang juga akan menghasilkan kesimpulan dari *use case* tersebut.

2.14.3 *Activity Diagram*

Digunakan untuk model perilaku dalam independen proses bisnis benda. Dalam banyak hal, diagram aktivitas dapat dipandang sebagai diagram aliran data yang canggih yang digunakan dalam hubungannya dengan analisis terstruktur. Namun, tidak seperti aliran data diagram, diagram aktivitas termasuk notasi yang membahas pemodelan paralel, kegiatan bersamaan dan proses. (Dennis, 2010)

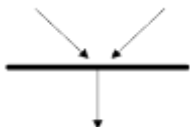
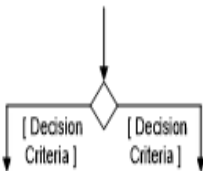
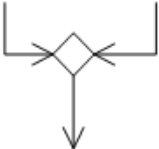
Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
- Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
- Rancangan menu yang akan ditampilkan pada perangkat lunak

Berikut adalah simbol-simbol yang ada pada *Activity Diagram*:

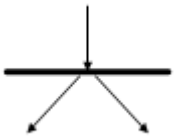
Tabel II.5 *Activity Diagram*

No.	Simbol	Deskripsi
-----	--------	-----------

1.	<i>Initial Node</i> 	Merupakan tanda awal dari sebuah aktifitas.
2.	<i>Activity</i> 	Merupakan sebuah gambaran aktifitas yang terjadi
3.	<i>Final-Activity Node</i> 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
4.	<i>Control flow</i> 	Menunjukkan urutan eksekusi
5.	<i>Join</i> 	Join node digunakan untuk menyatukan / menutup aktivitas yang berjalan paralel yang sebelumnya dibentuk oleh Fork node
6.	<i>Decision Node</i> 	Alur logika ini merupakan pilihan atas jalur aktivitas yang bernilai "True" dan "False", dan hanya salah satu dari jalur tersebut yang akan dipilih sesuai dengan syarat/kriteria pada <i>Decision node</i> yang telah ditentukan
7.	<i>Merge Node</i> 	<i>Merge node</i> digunakan untuk menyatukan / menutup alur logika yang sebelumnya dibentuk oleh <i>Decision node</i> .
8.	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Tabel II.5 Activity Diagram (Lanjutan)

No.	Simbol	Deskripsi
-----	--------	-----------

9.	<i>Fork</i> 	Fork node digunakan untuk membagi (<i>split</i>) sekumpulan aktivitas dalam suatu proses bisnis, dimana sekumpulan aktivitas tersebut dapat berjalan secara paralel dalam satu kondisi waktu yang sama.
----	--	---

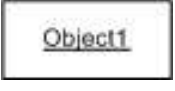
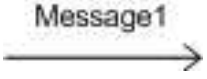
(Sumber: Dennis, 2010)

2.14.4 Sequence Diagram

Sequence Diagram adalah salah satu dari dua jenis diagram interaksi. Mereka menggambarkan benda-benda yang berpartisipasi dalam kasus penggunaan dan pesan yang melewati antara mereka dari waktu ke waktu untuk satu *use case*. Sebuah *Sequence Diagram* adalah model dinamis yang menunjukkan urutan eksplisit pesan yang lewat di antara objek dalam interaksi yang didefinisikan. Karena urutan diagram menekankan pemesanan berbasis waktu kegiatan yang terjadi di antara set benda, mereka sangat membantu untuk memahami spesifikasi *real time* dan kompleks menggunakan kasus (Dennis, 2010).

Simbol-simbol *Sequence Diagram* dapat dilihat pada Tabel II.6 berikut:

Tabel II.6 *Sequence Diagram*

Simbol	Nama	Deskripsi
	<i>Object</i>	Berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan yang ditempatkan di atas diagram.
	<i>Message</i>	Pesan yang menggambarkan komunikasi yang terjadi antar objek.
	<i>Actor</i>	Orang atau sistem yang berasal dari manfaat dan eksternal ke sistem yang berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan.

Tabel II.6 *Sequence Diagram* (Lanjutan)

Simbol	Nama	Deskripsi
--------	------	-----------

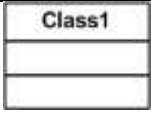
	<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
	<i>Execution Occurrence</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
X	<i>Object Destruction</i>	Ditempatkan pada akhir dari suatu <i>object lifeline</i> untuk menunjukkan bahwa itu akan keluar dari eksistensi.

(Sumber: Dennis, 2010)

2.14.5 Class Diagram

Class Diagram merupakan salah satu diagram dalam konsep *Unified Modelling Language* (UML) yang menjelaskan mengenai berbagai jenis objek yang terdapat dalam sistem beserta beberapa hubungan antar objek tersebut. Sebuah *class* merupakan deskripsi dari sekumpulan objek yang memiliki properti (*attribute*), operasi (*method*), relasi (*association*), dan tingkah laku (*behavior*) yang sama. Sebuah *class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi) (Dennis, 2010). Simbol-simbol *Class Diagram* dapat dilihat pada Tabel II.7 berikut:

Tabel II.7 *Class Diagram*

Simbol	Nama	Deskripsi
	<i>Class</i>	Digunakan sebagai <i>template</i> dari kumpulan objek.
	<i>Association</i>	Hubungan statis antar <i>class</i> yang direpresentasikan dengan kata kerja. Menggambarkan <i>class</i> yang memiliki atribut berupa <i>class</i> lain, atau <i>class</i> yang harus mengetahui eksistensi <i>class</i> lain.

Tabel II.7 *Class Diagram* (Lanjutan)

Simbol	Nama	Deskripsi
	<i>Generalization</i>	Merupakan sebuah <i>taxonomic relationship</i> antara <i>class</i> yang lebih umum dengan <i>class</i> yang lebih khusus.

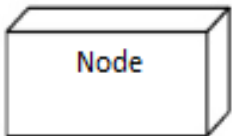
	<i>Aggregation</i>	Menggambarkan suatu <i>class</i> terdiri dari <i>class</i> lain atau suatu <i>class</i> adalah bagian dari <i>class</i> lain.
1 0..1 (0..*) 2..4 1..* 2, 4..6, 8 *	<i>Multiplicity</i>	Merupakan gambaran dari sejumlah objek pada suatu <i>class</i> yang terlibat dalam asosiasi dengan sejumlah objek pada <i>class</i> yang lainnya.

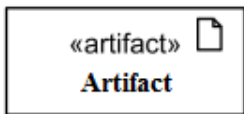
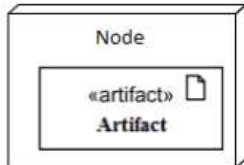
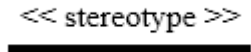
(Sumber: Dennis, 2010)

2.14.6 Deployment Diagram

Deployment diagram digunakan untuk mewakili hubungan antara komponen-komponen *hardware* yang digunakan dalam infrastruktur fisik sistem informasi. Misalnya, ketika mengarang suatu sistem informasi terdistribusi yang akan menggunakan jaringan luas, *deployment diagram* dapat digunakan untuk menunjukkan hubungan komunikasi antara node yang berbeda dalam jaringan. *Deployment diagram* juga dapat digunakan untuk mewakili komponen-komponen *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi. Dalam hal ini, *deployment diagram* juga dapat digunakan untuk mewakili komponen-komponen *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi. Dalam hal ini, *deployment diagram* mewakili lingkungan pembuatan *software* (Dennis, 2012). Berikut simbol-simbol yang digunakan pada *deployment diagram*:

Tabel II.8 *Deployment Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Node</i>	Untuk menggambarkan sumber daya komputasi dalam sebuah sistem (misalnya, computer klien, <i>server</i> , jaringan yang terpisah, atau individu perangkat jaringan).

2		<i>Artifact</i>	Untuk menggambarkan spesifikasi dari <i>software</i> atau <i>database</i> , misalnya <i>file</i> sumber, tabel <i>database</i> , <i>executable file</i> .
3		<i>Node with a Deployed Artifact</i>	Untuk menggambarkan <i>artifact</i> yang ditempatkan pada node fisik. Mendukung pemodelan distribusi perangkat lunak melalui jaringan.
4		<i>Communication Path</i>	Untuk menggambarkan hubungan antara dua node untuk bertukar pesan.

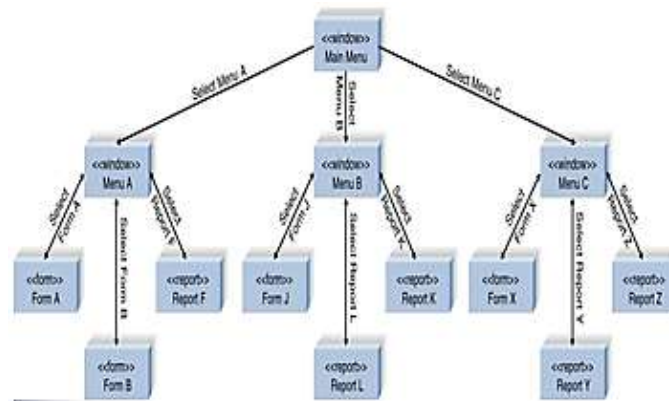
(Sumber: Dennis, 2011)

2.15 Windows Navigation Diagram

Windows Navigation Diagram merupakan *statechart* diagram khusus yang berfokus pada *user interface*. *Windows Navigation Diagram* (WND) menurut Dennis et al (2010) adalah sebagai berikut:

1. Menunjukkan bagaimana semua layar, formulir, dan laporan terkait.
2. Menunjukkan bagaimana pengguna bergerak dari satu ke yang lain.
3. Seperti diagram keadaan untuk *user interface*
 - a. Kotak mewakili komponen.
 - b. Panah mewakili transisi.
 - c. Stereotipe menunjukkan tipe antarmuka.

Contoh dari *Windows Navigation Diagram* dapat dilihat pada Gambar II.4.



Gambar

II.7 Contoh

Windows Navigation Diagram

(Sumber: Dennis et al, 2010)

2.16 Database

Basis Data terdiri dari 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya (Fathansyah, 2012). Sebagai satu kesatuan istilah, Basis data (*Database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.17 Entity Relationship Diagram (ERD)

Mujilan (2013) mengatakan bahwa E-R diagram dikenalkan oleh Chen (1976) yang ia sebut sebagai *entity-relationship model*. E-R model menggunakan teknik diagram khusus sebagai peralatan dalam mendisain *database*. Pada saat itu terdapat tiga model utama yang digunakan untuk menggambarkan data logik yaitu: *the network model*, *the relational model*, dan *the entity set model*. E-R model dimaksudkan untuk menanggapi kelebihan dan kelemahan dari ketiga model tersebut. Diharapkan E-R model mampu menjabarkan keadaan senyatanya yang terjadi di dalam kepentingan penjelasan entitas dan hubungannya.

Sedangkan menurut Brady dan Loonam (2010), *Entity Relationship Diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *System Analys* dalam tahap analisis persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain *database* relational yang mendasari sistem informasi yang dikembangkan.

Jenis-jenis atribut pada ERD (Brady dan Loonam, 2010) yaitu:

1. *Identifier (Key)*

Digunakan untuk menentukan suatu entitas secara unitk (primary key)

2. *Descriptor (nonKey attribute)*

Digunakan untuk menspesifikasian karakteristik dari suatu entitas yang tidak unik.

Relasi adalah hubungan beberapa entitas sedangkan kardinalitas menyatakan jumlah himpunan relasi antar entitas. Berikut ini merupakan maam-macam dari kardinalitas (Brady dan Loonam, 2010) yaitu:

1. *One-to-One*

Menyatakan hubungan sebuah entitas A dengan entitas B paling banyak satu jenis hubungan. Contoh, relasi pegawai dan departemen dimana setiap pegawai hanya bekerja pada 1 departemen.

2. *One-to-Many*

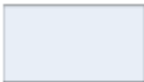
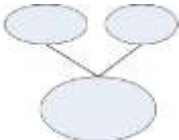
Menyatakan hubungan sebuah entitas A dengan entitas B sebanyak lebih dari satu jenis hubungan. Contoh, 1 departemen memiliki banyak pegawai.

3. *Many-to-Many*

Menyatakan sebuah hubungan entitas A dan entitas B memiliki lebih dari satu jenis hubungan begitu pula sebaliknya. Contoh, setiap 1 mahasiswa dapat mengambil banyak mata kuliah dan setiap 1 mata kuliah dapat diikuti oleh banyak mahasiswa.

Simbol-simbol ERD dapat dilihat pada Tabel II.9 berikut:

Tabel II.9 Simbol-Simbol *Entity Relationship Diagram*

Simbol/Notasi	Nama
	<i>Entity</i>
	<i>Weak Entity</i>
	<i>Relationship</i>
	<i>Identifying Relationship</i>
	Atribut
	Atribut <i>Primary Key</i>
	Atribut <i>Composite</i>

(Sumber: Mujilan, 2013)

2.18 Odoo

Dalam jurnal Aldi, Deden, Wahjoe (2016), Nopriyanto dkk berpendapat OpenERP (dulu bernama TinyERP, sekarang bernama Odoo) adalah sebuah perangkat lunak Enterprise Resource Planning (ERP) atau perangkat lunak perencanaan sumber daya perusahaan yang dilisensikan free/open source. Odoo dikembangkan dengan bahasa pemrograman *Python*. Terdapat modul pada odoo diantaranya: *CRM, Social Network, Invoicing & payments, Point of sale, Project*

management, Issue tracker, Accounting finance, Sales management, Warehouse management, MRP, Purchase management, Employee directory, Timesheets, Leave management, Expense management, Assets management, Payroll. Semua modul yang terinstall dirancang untuk dapat saling terhubung. Sedangkan menurut Adam, Luciana, R. Wahjoe (2016, Odoo (dulunya OpenERP) adalah sebuah perangkat lunak manajemen perusahaan berbasis Open source. Aplikasi ini mampu melakukan seluruh otomatisasi perusahaan, meliputi sebagian besar kebutuhan dan proses perusahaan yang terintegrasi. Odoo tidak membatasi jenis, kategori dan skalabilitas perusahaan-perusahaan yang akan menggunakannya karena odoo dapat diaplikasikan baik pada bidang, sektor, maupun skala apapun. Odoo dapat diimplementasikan secara modular. Selain itu odoo juga dapat diimplementasikan pada sistem operasi linux dan windows untuk server serta klien linux, windows, dan Mac OS X. *Database* yang digunakan pada Odoo adalah PostgreSQL dengan bahasa pemrograman *Phyton*.

Odoo menawarkan dua versi terpisah yaitu versi Odoo *Community* dan Odoo *Enterprise*. Versi komunitas adalah versi *open source* sementara versi *Enterprise* yaitu edisi yang dilengkapi fitur dan layanan komersial. Kode Sumber untuk kerangka kerja *OpenObject* dan modul inti ERP (*enterprise resource planning*) ini dikuratori oleh Odoo S.A yang berbasis di Belgia. Odoo adalah sistem ERP open source yang paling populer (wikipedia.org/wiki/Odoo, 2018).

2.19 Modul Sales Management

Menurut Wibisono dikutip oleh Risa, Yuli dan Azani (2017), modul Sales ditekankan pada penggunaan strategi penjualan yang mampu mengantisipasi perubahan pasar. Prioritas utama dari penggunaan modul ini adalah untuk membuat struktur data yang mampu merekam, menganalisis, dan mengontrol aktifitas untuk memberikan kepuasan kepada pelanggan dan menghasilkan keuntungan yang layak dalam periode akuntansi yang akan datang. Sub modul yang terdapat pada proses sales pada Odoo yaitu:

1. *Customers*

Mendaftarkan calon *customer* dalam perusahaan yang berupa identitas, lokasi dan contact person calon *customer* sehingga perusahaan dapat dengan mudah dalam mengidentifikasi data *customer*.

2. Pembentukan Harga Barang

Adalah proses pembentukan harga barang yang telah ditetapkan oleh kedua belah pihak, antara perusahaan dengan *customer*.

3. *Quotations*

Suatu dokumen yang berisikan penawaran harga kepada *customer*. Saat *customer* sudah sepakat dengan harga yang telah ditawarkan, maka *quotations* dikonfirmasi dan menjadi *sales order*.

4. *Sales Order*

Dokumen yang berisi tentang konfirmasi penjualan barang kepada *customer*, sehingga perusahaan dapat mempersiapkan barang yang akan diproduksi dan dapat langsung dikirim kepada *customer*.

5. *Invoice*

Invoice merupakan dokumen yang digunakan sebagai pernyataan tagihan yang harus di bayar oleh *customer*. *Invoice* digunakan sebagai bukti pengiriman dan biasanya diberikan setelah pengiriman barang dilakukan.

6. *Products*

Perusahaan dapat mengatur barang yang dimiliki dan mengatur pengadaan hingga distribusi kepada *customer*.

7. *Delivery*

Delivery digunakan untuk mengatur informasi pengiriman barang yang meliputi jadwal pengiriman, jumlah barang, dan lokasi pengiriman barang.

2.20 PostgreSQL

PostgreSQL atau sering disebut Postgres merupakan database server yang memiliki lisensi GPL (*General Public Licence*). SQL di Postgres tidak seperti pada RDBMS (*Relational Database Management System*). PostgreSQL adalah *Object Relational Database Management System* (ORDBMS) yang bersifat *open source*, sehingga Postgres memiliki kekuatan besar yakni *source code* dan arsitektur yang

luwes. Postgres mendukung bahasa SQL92 dan SQL99 dimana bahasa tersebut didukung oleh banyak antar muka ke berbagai bahasa pemrograman seperti C++, Java, Perl, PHP, Python dan Tcl (Kusumawati, 2015).

2.20.1 Tipe Data PostgreSQL

Tipe data yang ada pada *database* PostgreSQL adalah:

1. Jenis *Numeric*

Jenis data numerik dapat dilihat pada tabel II.10 berikut ini:

Tabel II.10 Tipe Data Jenis *Numeric* pada PostgreSQL

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Range</i>
<i>Smallint</i>	2 bytes	<i>Small-range integer</i>	-32.768 to 32.767
<i>Integer</i>	4 bytes	<i>Typical choice for integer</i>	-2147463648 to 2147463647
<i>Bigint</i>	8 bytes	<i>Large-range integer</i>	-9223372036854775808 to 9223372036854775807
<i>Decimal</i>	<i>Variable</i>	<i>User-specified precision, exact</i>	<i>Up to 131.072 digits before the decimal point; up to 16.383 digits after decimal points</i>
<i>Numeric</i>	<i>Variable</i>	<i>User-specified precision, exact</i>	<i>Up to 131.072 digits before the decimal point; up to 16.383 digits after decimal points</i>
<i>Real</i>	4 bytes	<i>Variable-precision, inexact</i>	6 decimal digits precision

Tabel II.10 Tipe Data Jenis *Numeric* pada PostgreSQL (Lanjutan)

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Range</i>
<i>Double precision</i>	8 bytes	<i>Variable-precision, inexact</i>	15 decimal digits precision
<i>Smallserial</i>	2 bytes	<i>Small autoincrementing integer</i>	1 to 32767
<i>Serial</i>	4 bytes	<i>Autoincrementing integer</i>	1 to 2147483647

<i>Bigserial</i>	8 bytes	<i>Large autoincrementing integer</i>	1 to 9223372036854775807
------------------	---------	---	--------------------------

(Sumber: PostgreSQL.org, 2018)

2. Jenis Mata Uang (*Monetary Types*)

Jenis data mata uang dapat dilihat pada tabel II.11 berikut ini:

Tabel II.11 Tipe Data Jenis Mata Uang pada PostgreSQL

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Range</i>
<i>Money</i>	8 bytes	<i>Currency amount</i>	-9223372036547758,08 to + 9223372036547758,07

(Sumber: PostgreSQL.org, 2018)

3. Jenis Karakter (*Character Types*)

Jenis data karakter dapat dilihat pada tabel II.12 berikut ini:

Tabel II.12 Tipe Data Jenis Karakter pada PostgreSQL

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Range</i>
<i>Character (n), char (n)</i>		<i>Fixed-length, blank padded</i>	
<i>text</i>		<i>Variable unlimited length</i>	

Tabel II.12 Tipe Data Jenis Karakter pada PostgreSQL (Lanjutan)

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Range</i>
<i>Character varying (m), varchar (n)</i>		<i>Variable-length with limit</i>	

(Sumber: PostgreSQL.org, 2018)

4. Jenis Data Biner (*Binary Data Types*)

Jenis data biner dapat dilihat pada tabel II.13 berikut ini:

Tabel II.13 Tipe Data Jenis Data Biner pada PostgreSQL

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Range</i>
<i>Bytea</i>	1 or 4 bytes plus the actual binary string	<i>Variable-length binary string</i>	

(Sumber: PostgreSQL.org, 2018)

5. Jenis Data Boolean (*Boolean Data Types*)

Jenis data boolean dapat dilihat pada tabel II.14 berikut ini:

Tabel II.14 Tipe Data Jenis Data Boolean pada PostgreSQL

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Range</i>
Boolean	1 byte	State of true or false	

(Sumber: PostgreSQL.org, 2018)

6. Jenis Data Tipe Geometri (*Geometric Types*)

Jenis data geometri dapat dilihat pada tabel II.15 berikut ini:

Tabel II.15 Tipe Data Jenis Data Geometri pada PostgreSQL

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Range</i>
Point	16 bytes	Point on a plane	(x,y)
Line	32 bytes	Infinite line (not fully implemented)	((x1,y1), (x2,y2))
Lseg	32 bytes	Finite line segment	((x1,y1), (x2,y2))

Tabel II.15 Tipe Data Jenis Data Geometri pada PostgreSQL (Lanjutan)

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Range</i>
Box	32 bytes	Rectangular box	((x1,y1), (x2,y2))
Path	16+16n bytes	Close path (similar to polygon)	((x1,y1),....)
Path	16+16n bytes	Open path	[(x1,y1),....]
Polygon	40+16n bytes	Polygon (similar to closed path)	((x1,y1),....)
Circle	24 bytes	Circle	<(x,y),r> (center point and radius)

(Sumber: PostgreSQL.org, 2018)

7. Tanggal/Waktu (*Dates/Time Types*)

Jenis data tanggal/waktu dapat dilihat pada tabel II.16 berikut ini:

Tabel II.16 Tipe Data Tanggal/Waktu pada PostgreSQL

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Low Value</i>	<i>High Value</i>	<i>Resolution</i>
Timestamp [(p)] [without time zone]	8 bytes	Both date and time (no time zone)	4713 BC	294276 AC	1 microsecond / 14 digits

<i>Timestamp [(p)] with time zone</i>	8 bytes	<i>Both date and time with time zone</i>	4713 BC	294276 AC	1 microsecond / 14 digits
<i>Date</i>	4 bytes	<i>Date (no time of day)</i>	4713 BC	5874897 AD	1 day
<i>Time [(p)] [without time zone]</i>	8 bytes	<i>Time of day (no date)</i>	00:00:00	24:00:00	1 microsecond / 14 digits
<i>Time [(p)] with time zone</i>	12 bytes	<i>Time of day only, with time zone</i>	00:00:00 +1459	24:00:00 - 1459	1 microsecond / 14 digits

Tabel II.16 Tipe Data Tanggal/Waktu pada PostgreSQL (Lanjutan)

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Low Value</i>	<i>High Value</i>	<i>Resolution</i>
<i>Interval [fields] [(p)]</i>	16 bytes	<i>Time interval</i>	- 178000000 years	178000000 years	1 microsecond / 14 digits

(Sumber: PostgreSQL.org, 2018)

8. Jenis Data Alamat Jaringan (Network Address Types)

Jenis data alamat jaringan dapat dilihat pada tabel II.18 8berikut ini:

Tabel II.17 Tipe Data Alamat Jaringan pada PostgreSQL

<i>Name</i>	<i>Storage Size</i>	<i>Description</i>	<i>Range</i>
<i>cidr</i>	7 or 9 bytes	<i>IPv4 and IPv6 network</i>	
<i>inet</i>	7 or 9 bytes	<i>IPv4 and IPv6 hosts and network</i>	
<i>macaddr</i>	6 bytes	<i>MAC addresses</i>	

(Sumber: PostgreSQL.org, 2018)

2.21 Black Box Testing

Menurut Rizky (2011) *Black Box testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para *tester* memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses *testing* di bagian luar.

Jenis *testing* ini hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan. Contoh, jika terdapat sebuah perangkat lunak yang merupakan sebuah sistem informasi *inventory* di sebuah perusahaan. Maka pada jenis *black box testing*, perangkat lunak tersebut akan dieksekusi kemudian dites apakah telah memenuhi kebutuhan pengguna yang diidentifikasi pada saat awal tanpa harus membongkar listing programnya.

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Menurut Sugiyono dikutip dari buku Metode Penelitian Hukum oleh Jonaedi dan Johnny (2016), metode penelitian adalah cara-cara ilmiah untuk mendapatkan data yang valid, dengan tujuan dapat ditemukan, dikembangkan, dan dibuktikan suatu pengetahuan tertentu sehingga pada gilirannya dapat digunakan untuk memahami, memecahkan dan mengantisipasi masalah. Untuk menghasilkan penelitian tugas akhir yang lengkap dibutuhkan metode penelitian yang sesuai dengan masalah yang akan dibahas. Metode penelitian yang digunakan untuk menyelesaikan tugas akhir ini terdiri dari beberapa tahapan, yaitu tahapan pengumpulan data atau informasi dan pengembangan sistem. Dalam tahap pengumpulan data dan informasi dilakukan dengan cara observasi langsung di tempat penelitian, wawancara dengan pengguna sistem yang diamati dan studi kepustakaan. Sedangkan dalam pengembangan sistem menggunakan model *prototype*.

3.2 Jenis dan Sumber Data

Data yang dibutuhkan dalam penelitian ini dikelompokkan menjadi dua jenis, yaitu sumber data primer dan data sekunder. Sumber dari data-data ini berasal dari tempat yang diamati pada praktek kerja lapangan pada PT DRA Component Persada.

1. Data primer

Data yang diperoleh melalui pengamatan langsung pada Departemen *Marketing* terhadap sistem yang sedang berjalan, masalah-masalah pada sistem yang berjalan dan wawancara dengan pegawai sebagai sumber informasinya. Dalam penelitian ini data yang diperoleh berupa alur proses penjualan produk yang berjalan.

2. Data Sekunder

Data sekunder merupakan sumber data penelitian yang diperoleh secara tidak langsung melalui media perantara, kepustakaan, buku-buku, internet, dan referensi. Dalam penelitian ini data tersebut berupa data umum perusahaan seperti profil perusahaan, dan struktur organisasi perusahaan.

3.3 Metode Pengumpulan Data

Metode pengumpulan data merupakan sarana yang menampung dan mengumpulkan data-data dari berbagai sumber dan bentuk yang akan digunakan dalam keperluan menganalisis sistem untuk perancangan sistem usulan. Metode yang digunakan dalam pengumpulan data pada penelitian ini, diantaranya:

1. Studi Lapangan

Studi lapangan adalah usaha melakukan pengumpulan data secara langsung pada objek, dilakukan dengan cara sebagai berikut:

- a. Pengamatan

Pada tahap ini dilakukan pengamatan langsung terhadap pengguna sistem penjualan produk dan pengamatan langsung terhadap sistem penjualan produk yang sedang berjalan pada PT DRA Component Persada. Observasi ini dilakukan pada Departemen *Marketing* untuk mengetahui alur proses penjualan produk.

- b. Wawancara

Penulis melakukan wawancara dengan para pengguna sistem penjualan produk yang sedang berjalan pada tahap ini untuk memperoleh keterangan mengenai alur proses penjualan produk serta mengenai kebutuhan sistem yang akan dikembangkan selanjutnya. Wawancara ini dilakukan dengan cara berdialog dan bertanya dengan *General Manager* yang menangani proses keseluruhan pada proses bisnis di PT DRA Component Persada salah satunya adalah proses bisnis di Departemen *Marketing*.

2. Studi Pustaka

Tahap ini merupakan tahap pengumpulan data dari sumber-sumber seperti internet, buku, jurnal, yang berhubungan dengan masalah yang sedang dianalisis yaitu mengenai sistem penjualan produk. Studi pustaka yang dilakukan berkaitan dengan sistem, informasi, ERP, PostgreSQL, Odoo, penjualan, produk, analisis dan desain sistem.

3.4 Metode Pengembangan Sistem

Metode pengembangan sistem adalah metode-metode, prosedur-prosedur, konsep-konsep dan aturan-aturan untuk mengembangkan sistem (Yakub, 2012). Untuk mengatasi masalah yang ada pada sistem, pada penelitian ini diputuskan untuk membuat pengembangan sistem. Dalam pengembangan sistem ini digunakan metode *prototype*. Dalam model *prototype* dilakukan tahap

analisis, desain, dan implementasi secara bersamaan dan tiga tahapan tersebut dilakukan secara berulang dalam satu siklus sampai sistem selesai dibangun (Dennis et al, 2010).

Tahapan-tahapan dalam metode *prototype* adalah sebagai berikut:

1. Tahap Perencanaan

Dalam tahap ini, pengembang menentukan aspek-aspek perangkat lunak yang akan dibuat, yaitu meliputi konten dan alur kerja perangkat lunak.

2. Tahap Pengembangan Sistem

- a. Analisis Sistem

Dalam fase ini akan menentukan kebutuhan data, proses data dan informasi. Tahap analisis menjawab pertanyaan tentang siapa yang menggunakan sistem, sistem apa yang akan dilakukan, di mana dan kapan sistem tersebut akan digunakan.

- b. Desain Sistem

Pada fase ini akan diputuskan bagaimana sistem akan beroperasi, dalam hal perangkat keras, perangkat lunak, infrastruktur jaringan, antarmuka pengguna, formulir, laporan yang akan digunakan, program khusus, *database*, dan file yang akan diperlukan. Pembangunan sistem dilakukan sesuai dengan perancangan sistem yang dibuat berdasarkan analisis kebutuhan pengguna. Hasil perancangan direalisasikan dalam bentuk media perangkat lunak yang siap diimplementasikan lingkungan kerja.

- c. Implementasi *Prototype*

Pada tahap ini dilakukan pembangunan *prototype* berdasarkan desain sistem yang telah dibuat.

3. Tahap Implementasi Sistem

Prototype yang telah dibangun akan diperlihatkan kepada *user*, jika disetujui maka sistem akan dibangun jika tidak akan dibuat *prototype* ulang hingga *prototype* disetujui oleh *user*. Selanjutnya dilakukan pengujian sistem terhadap fungsi-fungsi pada *prototype* tersebut apakah sudah berjalan dengan baik atau belum. Metode yang digunakan untuk pengujian *prototype* ini yaitu *black box testing*.

3.5 **Literatur Review**

Literature review tidak hanya bermakna membaca literatur, tapi lebih ke arah evaluasi yang mendalam dan kritis tentang penelitian sebelumnya pada suatu topik. *Literature review* memiliki peran penting dalam membuat suatu tulisan ataupun karangan ilmiah, karena dapat memberikan ide dan tujuan tentang topik penelitian yang akan dilakukan. Pada umumnya berisi ulasan, rangkuman, dan pemikiran penulis tentang beberapa pustaka (buku, jurnal, majalah) yang berkaitan dengan topik yang dibahas. Menurut Romi Satrio (2016), yang disebut *literature review* dapat berupa:

- a. Paper dari *Journal* ilmiah

- b. Paper dari *Conference (Procedings)*

- c. *Thesis* dan Disertasi
- d. *Report* (laporan) dari Organisasi yang terpercaya
- e. Buku *Textbook*

Berikut adalah *literature review* dari beberapa jurnal ilmiah:

Tabel III.1 *Literature Review*

No	Pengarang/ Tahun	Topik Penelitian	Permasalahan	Metode yg digunakan	Kesimpulan Penelitian	Hasil Review (Perbandingan)
1.	Dika Anugerah Pratama, Ari Yanuar Ridwan, R. Wahjoe	Penerapan Sistem <i>Sales</i>	Pada proses penjualan, belum terintegrasinya	Metode <i>Rapid Application Development</i>	Dengan menggunakan OpenERP modul <i>sales</i>	Perbedaan antara jurnal dan tugas akhir yaitu, pada jurnal menggunakan

Tabel III.1 *Literature Review* (lanjutan)

No	Pengarang/ Tahun	Topik Penelitian	Permasalahan	Metode yg digunakan	Kesimpulan Penelitian	Hasil Review (Perbandingan)
----	---------------------	---------------------	--------------	------------------------	--------------------------	-----------------------------------

	Witjaksono, 2016	<i>Managemen</i> t menggunakan OpenERP Dengan Metode RAPID Application Development pada PT. Genta Trikarya	antara bagian <i>marketing</i> dan produksi, ketidaksesuaian stok barang dengan data yang ada dengan stok <i>real</i> yang ada di <i>warehouse</i> dan kesalahan dalam pendataan hasil penjualan.	(RAD)	<i>management, sales order</i> yang dibuat dapat langsung terintegrasi dan langsung digenerate ke bagian <i>manufactur</i> , informasi mengenai stok barang dapat dilihat secara <i>realtime</i> melalui sistem, dan aplikasi OpenERP juga terdapat <i>report</i> sehingga bisa mengetahui detail penjualan secara langsung.	OpenERP (versi Odoo lama) dan metode RAD sedangkan TA menggunakan Odoo versi 8.0 dan metode <i>Prototype</i> . Pada jurnal melakukan penerapan di PT Genta Trikarya sedangkan TA pada PT DRA. Selain itu pada jurnal, peneliti melakukan penerapan ERP sampai dengan implementasi pada perusahaan sedangkan pada TA hanya sampai pada pengenalan <i>prototype</i> ERP yang akan dikembangkan. Sementara itu, persamaan antara jurnal dan Tugas Akhir yaitu sama-sama menggunakan modul <i>Sales Management</i> untuk menjalankan proses penjualan pada sistem usulannya. Selain itu pada jurnal dan Tugas Akhir melakukan pengamatan pada proses penjualan yang terjadi di perusahaan. Persamaan lain yaitu masalah yang terjadi pada proses penjualan yaitu belum terintegrasi antar bagian <i>marketing</i> dengan bagian produksi.
2.	Anisa Utami, Nia Ambarsari, R. Wahoje Witjaksono, 2015	Penerapan Sistem <i>Sales Management</i> Menggunakan	Pada proses penjualan, tidak ada integrasi antara bagian <i>sales</i> dengan <i>warehouse</i> ,	Metode Spiral	Dengan menggunakan OpenERP, penjualan pada perusahaan dapat	Perbedaan antara jurnal dan tugas akhir yaitu, pada jurnal menggunakan OpenERP (versi Odoo lama) dan metode spiral

Tabel III.1 Literature Review (lanjutan)

No	Pengarang/ Tahun	Topik Penelitian	Permasalahan	Metode yg digunakan	Kesimpulan Penelitian	Hasil Review (Perbandingan)
----	---------------------	---------------------	--------------	------------------------	--------------------------	-----------------------------------

		OpenERP pada PT.XYZ Dengan Metode Spiral	<i>purchasing</i> dan <i>accounting</i> , tidak adanya pencatatan data <i>customer</i> , tidak adanya dokumen <i>quotation</i> dan lamanya proses pembuatan laporan penjualan.		Terdokumentasi sehingga dapat mengetahui hasil penjualan secara <i>realtime</i> , proses penjualan pada bagian <i>sales</i> dapat terintegrasi dan bagian <i>sales</i> dapat terintegrasi dengan bagian <i>warehouse</i> , <i>purchase</i> dan <i>accounting</i> .	sedangkan pada TA menggunakan Odoo versi 8.0 dan metode <i>Prototype</i> . Pada jurnal melakukan penerapan di PT XYZ sedangkan TA pada PT DRA. Selain itu pada jurnal, peneliti melakukan penerapan ERP sampai dengan implementasi pada perusahaan sedangkan pada TA hanya sampai pada pengenalan <i>prototype</i> ERP yang akan dikembangkan. Sementara persamaan antara jurnal dan Tugas Akhir yaitu sama-sama menggunakan modul <i>Sales Management</i> untuk menjalankan proses penjualan pada proses bisnis usulannya. Selain itu pada jurnal dan TA melakukan pengamatan pada proses penjualan yang terjadi di perusahaan. Persamaan lainnya juga terletak pada masalah yang terjadi di perusahaan yaitu kegiatan penjualan belum terdokumentasi dengan baik.
--	--	--	--	--	--	---

(Sumber: Hasil Analisis, 2018)

3.6 Kerangka Penelitian

Dalam penelitian yang dibahas dalam tugas akhir ini, dilakukan tahap-tahap dalam melakukan penyusunan penelitian. Beberapa tahapan yang penulis lakukan dalam melakukan penelitian ini adalah sebagai berikut:

1. Studi pendahuluan yang dilakukan yaitu dengan melakukan proses pemagangan di PT DRA Component Persada. Studi pendahuluan ini dimaksudkan untuk dapat mengetahui gambaran yang jelas mengenai kondisi dan situasi pada saat ini, serta untuk mengetahui masalah yang

sedang dihadapi. Selain itu juga membaca buku, literatur, serta sumber-sumber lain dalam lingkup perkuliahan maupun di luar lingkup perkuliahan yang berhubungan dengan judul dan permasalahan Tugas Akhir.

2. Identifikasi Masalah

Cara mengidentifikasi masalah dalam penelitian ini adalah dengan:

- a. Observasi di Departemen *Marketing*
- b. Wawancara dengan pihak-pihak terkait di Departemen *Marketing*
- c. Analisis hasil wawancara dalam bentuk transkrip wawancara.
- d. Analisis hasil observasi dalam bentuk tabel observasi.
- e. Analisis PIECES untuk memperoleh pokok-pokok permasalahan yang lebih spesifik.

3. Identifikasi Solusi

Pada tahap ini bertujuan untuk mencari solusi dari masalah yang telah diidentifikasi pada tahap sebelumnya. Cara mengidentifikasi solusi dalam penelitian ini adalah dengan:

- a. Mengidentifikasi maksud penelitian, yaitu menerapkan usulan implementasi sistem ERP berbasis *open source* untuk membantu mengelola data penjualan.
- b. Mengidentifikasi tujuan penelitian, yaitu:
 - Pembuatan sistem yang terhubung dengan database yang dapat membantu proses peng-*input*-an surat penawaran harga dan pengajuan permintaan produksi menjadi terkomputerisasi
 - Membuat sistem penjualan produk yang memiliki hak akses.
- c. Mengidentifikasi batasan masalah penelitian, yaitu:
 - Penelitian dilakukan di PT Component Persada pada Departemen *Marketing* mulai dari 1 November sampai dengan 31 Desember 2017.
 - Analisis dan penelitian hanya sebatas mengenai proses penjualan produk yang mencakup proses penawaran harga (*Quotation*), pengajuan permintaan produksi (MO atau *Manufacturing Order*) dan pembuatan surat jalan (*Delivery Order*) tetapi tidak membahas mengenai proses produksi, penagihan dan pembayaran serta *retur* penjualan produk.
 - Menggunakan Odoo 8.0, *database* PostgreSQL 9.3 dan modul *Sales Management*
 - Menggunakan Modul *Sales Management* untuk menjalankan proses penjualan produk pada modul tersebut berdasarkan proses penjualan produk yang berjalan pada perusahaan

4. Tahap Pengembangan Sistem

Pada tahap pengembangan sistem ini, penulis menggunakan metode *prototype*. Adapun yang dilakukan pada tahap pengembangan sistem ini yaitu:

a. Analisis Sistem (*Analysis*)

- Analisis sistem berjalan, berupa:
 - Identifikasi dokumen masukan dan dokumen keluaran.
 - Identifikasi pada proses bisnis penjualan produk
- Analisis sistem usulan, berupa:
 - Identifikasi *Functional Requirement*
Menggambarkan persyaratan sistem secara detil seperti *input*, *output*, dan pengecualian yang berlaku.
 - Identifikasi *Non Functional Requirement*
Berisi batasan-batasan pada pelayanan atau fungsi yang disediakan oleh sistem.

b. Desain Sistem (*Design*)

Penulis melakukan desain sistem untuk pembuatan *prototype* proses bisnis usulan berdasarkan analisis sistem yang telah dilakukan. Adapun desain sistem yang dilakukan yaitu:

- Pemodelan sistem, menggunakan:
 - *Use case diagram* untuk menggambarkan hubungan yang terjadi antara sistem dengan penggunanya.
 - *Activity diagram* untuk menggambarkan aliran data dalam sistem berdasarkan *use case* yang telah dibuat.
 - *Sequence diagram* untuk menggambarkan interaksi antara aktor dan dokumen yang terkait untuk per *use case* yang dibuat.
 - *Class diagram* untuk menggambarkan objek yang terdapat pada sistem dan hubungan antar objek tersebut.
 - *Window navigation diagram* untuk menggambarkan desain alur *interface* pada sistem usulan.
- Pemodelan data, menggunakan *Entity relationship diagram* untuk memodelkan kebutuhan pada sistem.
- Desain proses bisnis berjalan, menggunakan *flowmap*.

c. Implementasi Sistem *Prototype*

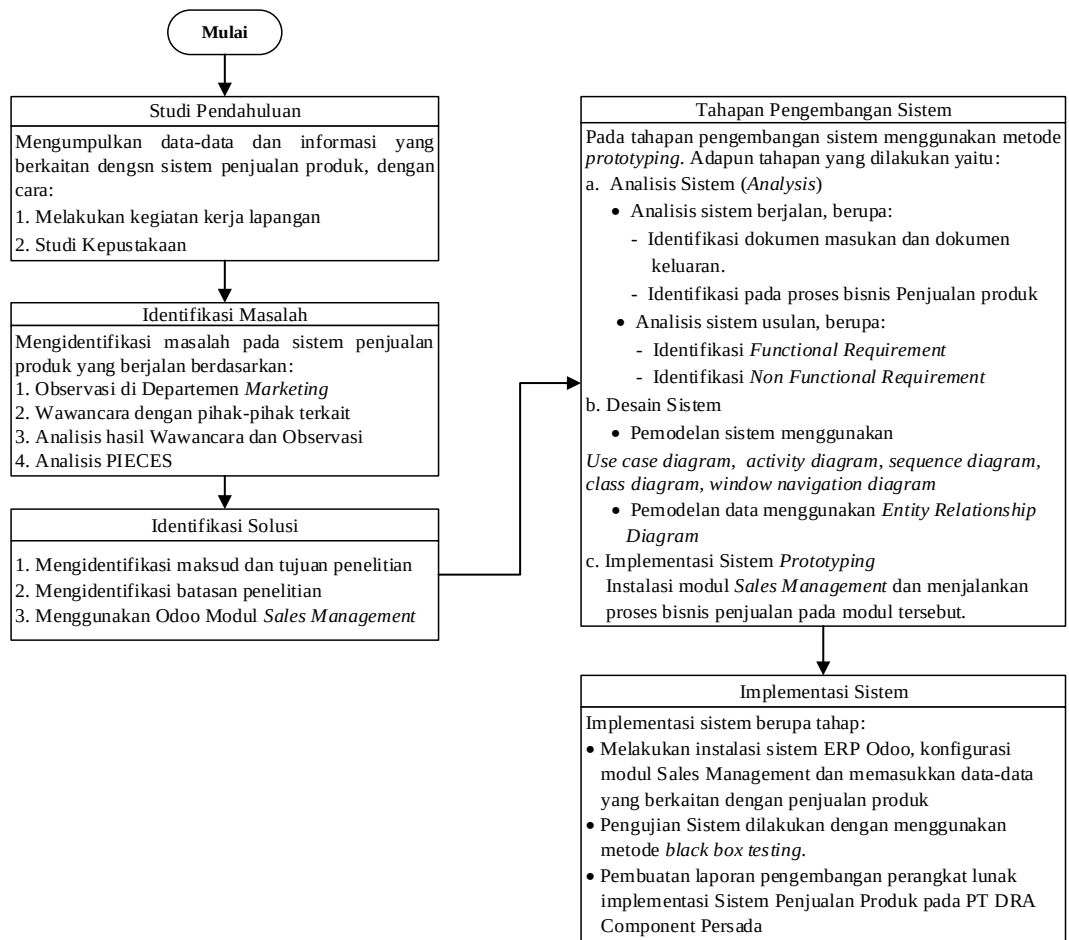
Pada tahap ini dilakukan instalasi modul *Sales Management* dan menjalankan proses penjualan produk pada modul tersebut berdasarkan desain sistem yang dibuat.

Prototype merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode *prototype* pengembang dan *user* dapat saling berinteraksi selama proses pembuatan sistem. Sistem *Prototype* akan melakukan tahapan analisis, desain, dan implementasi secara bersama dan tiga tahapan tersebut dilakukan secara berulang dalam satu siklus sampai sistem selesai dibangun. *Prototype* pertama dibangun biasanya masih memiliki fitur yang sedikit dan merupakan bagian awal dari sistem yang akan digunakan. Pada tahap ini *prototype* akan ditampilkan kepada *user* apakah *prototype* yang bangun ini telah sesuai dengan keinginan *user*. Jika *prototype* tersebut disetujui oleh *user* maka proses akan dilanjut ke tahapan berikutnya namun jika tidak disetujui maka akan dibuat *prototype* kedua dengan mengulang kembali tahapan analisis, desain dan implementasi, proses ini akan terus berulang sampai *user* setuju dengan *prototype* yang telah dibangun.

5. Implementasi Sistem

Implementasi sistem berupa tahap pembangunan sistem berdasarkan *prototype* yang telah disetujui. Implementasi sistem berupa instalasi sistem ERP Odoo versi 8.0 dengan *database* PostgreSQL 9.3, instalasi modul *Sales Management*, konfigurasi modul *Sales Management* berdasarkan sistem penjualan usulan, memasukkan data-data yang berkaitan dengan penjualan produk, dan melakukan pengujian fitur-fitur pada *prototype* yang telah dibuat apakah telah berjalan dengan baik atau belum. Pengujian dilakukan menggunakan metode *black box testing*, yaitu melakukan pengujian terhadap fungsionalitas sistem usulan penjualan produk serta mendokumentasikan dan melakukan pembuatan laporan tugas akhir berdasarkan semua tahapan yang telah dilakukan.

Berikut adalah *flowchart* kerangka penelitian untuk menyelesaikan masalah dalam Tugas Akhir:



Gambar III.1 Kerangka Penelitian
(Sumber: Pengolahan Data, 2018)

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Tentang Perusahaan

PT DRA Component Persada merupakan perusahaan yang bergerak dalam bidang *Manufacturing Industrial Fasteners Component* yang memproduksi berbagai macam baut dengan pangsa pasar elektronik dan otomotif. Produk utama yang dihasilkan adalah *screw*, *bolt*, *nut* dan *pin* dengan spesifikasi yang bermacam-macam. Proses produksi yang berjalan di perusahaan ini bersifat *make to order* yaitu proses produksi akan dilakukan apabila ada pesanan dari *customer* dimana pesanan tersebut sesuai dengan keinginan *customer*. Untuk menentukan produk yang diinginkan, *customer* dapat mengirimkan spesifikasi baut berupa gambar desain, bentuk asli barang yang diinginkan maupun berupa spesifikasi berdasarkan perkiraan. Kemudian tim produksi akan menentukan gambar desain baut yang dibuat. Setelah mendapatkan persetujuan maka kemudian barang akan diproduksi sesuai pesanan.

PT DRA Component Persada pertama kali didirikan pada tanggal 18 Februari 2010 oleh Bapak Sodikun. Beliau memiliki latar belakang pendidikan Sarjana Teknik Industri di Universitas Pancasila, Bogor pada tahun 1992-1996 dan melanjutkan pendidikan Pasca Sarjana Administrasi Bisnis di Universitas Mercu Buana, Jakarta pada tahun 1998-2001. Kemudian sekarang beliau menjadi dosen di Universitas Pancasila semenjak tahun 2011 sekaligus menjabat sebagai direktur PT DRA Component Persada. Pada mulanya perusahaan ini berlokasi di Jl. Raya Narogong KM 14,5 Pangkalan 5, Ciketing Udik, Bantar Gebang, Bekasi. Di lokasi tersebut pemilik lahan hanya menyewakan tempat guna dijadikan pabrik dengan mesin yang digunakan hanya berkisar kurang lebih 7 buah. Saat itu pak Sodikun merasa mesin yang digunakan belum cukup untuk memenuhi tingkat permintaan yang semakin tinggi. Jika ingin menambah mesin, area lokasi pabrik tidak cukup muat untuk memenuhi standar yang ada. Masalah pun bertambah ketika harga sewa tempat semakin tinggi sehingga mengurangi keuntungan dari perusahaan. Untuk

menghadapi masalah tersebut dan meminimalisir biaya sewa bangunan akhirnya pak Sodikun memutuskan untuk pindah lokasi dan membangun pabrik yang baru dengan tanah bangunan kepemilikan sendiri di daerah Cileungsi Setu.

Setelah melakukan pembangunan pabrik pada tahun 2015-2016 tepatnya pada tanggal 23 Januari 2017, semua aktifitas pada kantor pusat dan produksi dipindahkan ke pabrik baru yang berlokasi di Jl. Raya Cileungsi Setu, Kampung Pasirangin, Desa Pasirangin RT.001/RW. 004, Kecamatan Cileungsi, Kabupaten Bogor-Jawa Barat. Dengan jumlah karyawan 23 orang (November 2017). Dan luas area tanah 900 m² dan luas bangunan 584 m², yang terdiri dari ruang produksi, ruang kantor, gudang *finish good* dan ruang *packing*.

PT DRA Component Persada mengutamakan kualitas dan berkomitmen menghadirkan produk bermutu bagi pelanggan dengan harga yang kompetitif sehingga mampu disejajarkan dengan perusahaan lain yang sejenis. Oleh karena itu, seluruh produk yang dirancang sesuai dengan Standar Nasional Indonesia (SNI) dan menggunakan sistem manajemen mutu ISO 9001:2008 untuk menjamin dan menjaga kesesuaian mutu proses produk demi kepuasan pelanggan. Selain itu PT DRA Component Persada menerapkan RoHS *Free* pada produknya untuk menjaga kelestarian lingkungan dan kelangsungan umat manusia.

4.2 Profil Perusahaan

PT DRA Component Persada, produsen baut yang digunakan untuk peralatan elektronik dan otomotif. Seluruh produk dirancang dan diproduksi sesuai dengan Standar Nasional Indonesia (SNI) dan pengakuan Internasional dengan sertifikasi sistem mutu ISO 9001:2008.

Nama perusahaan	: PT DRA Component Persada
Jenis usaha	: <i>Manufacturing Industrial Fastener Component</i>
Tahun berdiri	: 2010
Produk	: <i>Screw, Bolt, Nut, dan Pin</i>
Alamat usaha	: Jalan Raya Cileungsi Setu, Kampung Pasirangin, Desa Pasirangin RT. 001/ RW.

	004, Kec. Cileungsi, Kab. Bogor-Jawa Barat, Indonesia	
Nama Pemilik	: Sodikun	
Area	: Tanah	: 900 m ²
	<i>Line Produksi</i>	: 500 m ²
	<i>Area Packing</i>	: 32 m ²
	<i>Area Finish Good</i>	: 26 m ²
	<i>Area Kantor</i>	: 26 m ²
Telepon	: 021-2296-7878	
<i>Fax</i>	: 021-2296-7878	
Jumlah pekerja	: 23 karyawan (November, 2017)	
<i>Email</i>	: <u>dra_component@yahoo.com</u>	
Kapasitas Mesin	: 1. Mesin <i>Heading</i>	: 9 unit
	2. Mesin <i>Rolling</i>	: 9 unit
	3. Mesin <i>Trimming</i>	: 2 unit
	4. Mesin <i>Cutting</i>	: 1 unit

4.3 Logo Perusahaan

Logo merupakan simbol dari organisasi, kelompok, atau perorangan yang mencerminkan makna atau pesan yang ingin disampaikan dari organisasi/kelompok tersebut. Logo juga dapat dikatakan sebagai suatu identitas yang unik. Logo untuk sebuah perusahaan juga sangatlah penting keberadaannya karena dengan logo, suatu perusahaan dapat dikenal oleh konsumen atau khalayak.



Berikut ini merupakan logo dari PT DRA Component Persada (Gambar IV.1):

Gambar IV.1 Logo PT DRA Component Persada

(Sumber: PT DRA Component Persada)

4.4 Visi, Misi dan Nilai Perusahaan

Visi dan Misi merupakan suatu hal yang penting yang harus dimiliki setiap perusahaan, dengan visi, misi dan motto perusahaan menjadi lebih terarah dalam pencapaian tujuan.

4.4.1 Visi Perusahaan

Menurut DR.A.B.Susanto dalam bukunya yang berjudul Visi dan Misi, visi adalah sebuah gambaran mengenai tujuan dan cita-cita di masa depan yang harus dimiliki organisasi sebelum disusun rencana bagaimana mencapainya.

Visi yang di miliki PT DRA Component Persada Industries adalah “Menjadi perusahaan manufaktur yang mengutamakan kepuasan pelanggan serta menjaga kelestarian lingkungan”

4.4.2 Misi Perusahaan

Menurut DR.A.B.Susanto dalam bukunya yang berjudul Visi dan Misi, misi adalah bagaimana untuk menghadirkan impian tadi menjadi kenyataan.

Misi yang di miliki PT DRA Component Persada sebagai berikut:

1. Meningkatkan dan mengembangkan mutu layanan berdasarkan prinsip profesionalisme.
2. Meningkatkan budaya kerja melalui kualitas SDM.
3. Memelihara dan menjaga lingkungan kerja dan lingkungan masyarakat sekitar

4.4.2 Nilai perusahaan

Nilai perusahaan merupakan kondisi tertentu yang telah dicapai oleh suatu perusahaan sebagai gambaran dari kepercayaan masyarakat terhadap perusahaan setelah melalui suatu proses kegiatan selama beberapa tahun, yaitu sejak perusahaan tersebut didirikan sampai dengan saat ini.

Berikut ini Nilai Perusahaan yang di miliki PT DRA Component Persada:

1. Kerjasama tim
2. Kualitas produk

3. Bertanggungjawab
4. Berintegritasi

4.5 Jumlah Karyawan dan Jam Kerja

Tabel di bawah ini menjelaskan jumlah karyawan secara keseluruhan dan juga jam kerja karyawan berdasarkan waktu kerja normal

4.5.1 Jumlah Karyawan

Berikut ini merupakan jumlah karyawan PT DRA Component Persada pada tahun 2018 (Tabel IV.1):

Tabel IV.1 Jumlah Karyawan PT DRA Component Persada

(Sumber: PT DRA Component Persada)

4.5.2 Jam Kerja

Berikut ini merupakan jam kerja PT DRA Component Persada pada tahun 2017 (Tabel IV.2, IV.3, IV.4):

1. Bagian

Tabel

Karyawan	Jumlah
Karyawan Tetap	23 orang
Karyawan Tidak Tetap	22 orang

Produksi

IV.2 Jam Kerja

Hari	Waktu Kerja 1	Istirahat	Waktu Kerja 2
Senin s.d. Kamis	08.00 – 12.00	12.00 – 12.45	12.45 – 17.00
Jumat	08.00 – 11.00	11.00 – 12.45	13.45 – 17.00
Sabtu	Lembur	Lembur	Lembur
Minggu	-	-	-

shift 1

shift 1

Tabel IV.3 Jam Kerja *shift 2*

Hari	Waktu Kerja 1	Istirahat	Waktu Kerja 2
-------------	----------------------	------------------	----------------------

Senin s.d. Jumat	20.00 – 00.00	00.00 – 01.00	01.00 – 05.00
Sabtu	Lembur	Lembur	Lembur
Minggu	-	-	-

(Sumber: PT DRA Component Persada)

2. Bagian Kantor

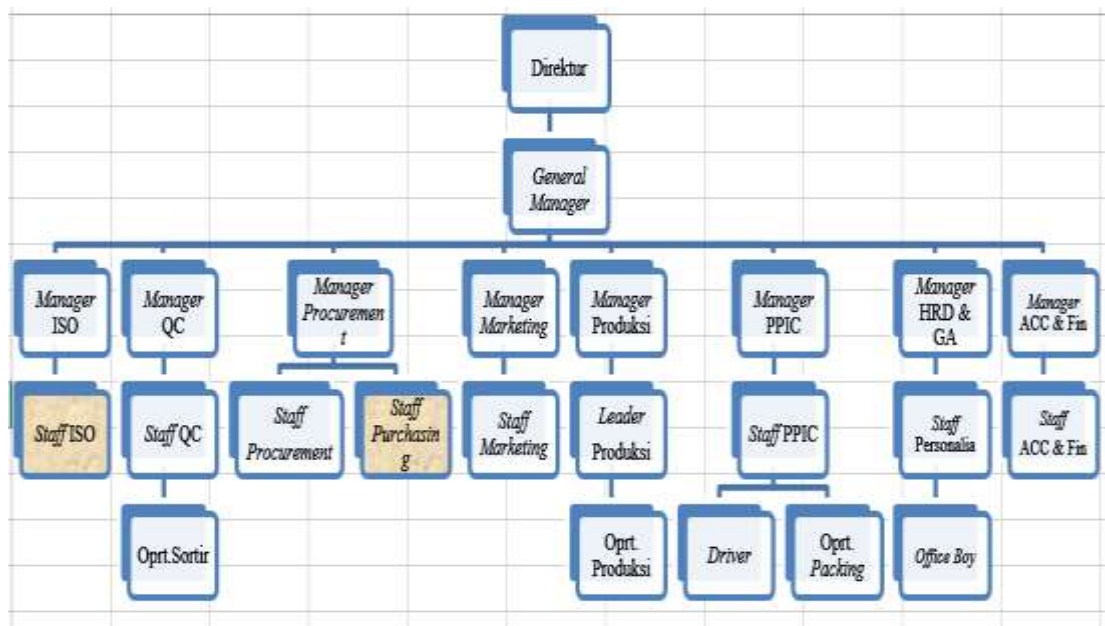
Tabel IV.4 Jam Kerja Bagian Kantor

Hari	Waktu Kerja 1	Istirahat	Waktu Kerja 2
Senin s.d. Jumat	08.00 – 12.00	12.00 – 12.45	12.45 – 17.00
Jumat	08.00 – 11.00	11.00 – 12.45	13.45 – 17.00
Sabtu	-	-	-
Minggu	-	-	-

(Sumber: PT DRA Component Persada)

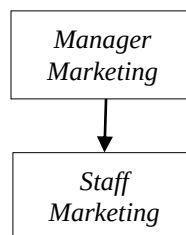
4.6 Struktur Organisasi Perusahaan

Tujuan perusahaan menyusun struktur organisasi adalah untuk mengatur kegiatannya secara efektif dan efisien serta memberikan gambaran yang sistematis mengenai hubungan kerjasama antara orang-orang yang terlibat untuk mencapai suatu tujuan. Oleh karena itu, struktur organisasi harus disusun secara jelas mengenai pemisahan fungsi, tugas dan tanggungjawab masing-masing bagian kepada pimpinan. Dilihat dari struktur organisasinya PT DRA Component Persada menggunakan bentuk organisasi garis/lini (*Line Organization*), dimana pada bentuk organisasi ini kekuasaan dan tanggung jawab bercabang dan mempunyai bawahan tertentu. Berikut merupakan struktur organisasi PT DRA Component Perada secara keseluruhan (Gambar IV.2):



Gambar IV.2 Struktur Organisasi PT DRA Component Persada
(Sumber: PT DRA Component Persada, 2018)

Berikut adalah Struktur Organisasi dari Departemen *Marketing* (Gambar IV.3):



Gambar IV.3 Struktur Organisasi Departemen *Marketing*

4.6.1 T (Sumber: PT DRA Component Persada, 2018)

Dengan menjalankannya bisnisnya sebagai PT (Perusahaan Terbatas), dengan PT DRA Component Persada di Indonesia mempunyai struktur organisasi yang solid dengan menempatkan personil yang tepat sesuai dengan tuntutan unit bisnis yang di jalankan dengan susunan sebagai berikut:

1. Direktur Perusahaan
 - a. Memimpin dan mengkoordinir organisasi agar tetap berjalan lancar.
 - b. Menentukan kebijakan perusahaan.
 - c. Mempunyai kewenangan dalam membuat perencanaan.
 - d. Mengendalikan segala sesuatu yang terjadi dalam perusahaan.

- e. Mengatur dan mengawasi semua yang terlibat dalam perusahaan.
- f. Membangun relasi dengan mitra kerja.
- g. Memberikan pelatihan dan pengembangan karyawan.
- h. Memiliki kewenangan mengangkat dan memutuskan hubungan kerja dengan karyawan.

2. *General Manager*

- a. Mengelola operasional harian perusahaan.
- b. Merencanakan, melaksanakan, mengkoordinasi, mengawasi dan menganalisis semua aktivitas bisnis perusahaan.
- c. Merencanakan, mengelola dan mengawasi proses penganggaran di perusahaan.
- d. Memastikan setiap departemen melakukan strategi perusahaan dengan efektif dan optimal
- e. Memutuskan dan membuat kebijakan untuk kemajuan perusahaan.
- f. Merencanakan dan mengeksekusi rencana strategis perusahaan jangka menengah dan jangka panjang untuk kemajuan perusahaan.
- g. Menghadiri pertemuan, seminar, konferensi maupun pelatihan.

3. Departemen ISO (*International Organization for Standardization*)

- a. Bertugas untuk mengkoordinasi setiap departemen untuk menyiapkan laporan-laporan terkait untuk keperluan audit.
- b. Memeriksa dan mengevaluasi kelayakan dari *supplier*.

4. Departemen *Quality Control*

Bertugas untuk menjamin produk yang dihasilkan memenuhi standar kualitas yang diinginkan pelanggan.

5. Departemen *Procurement*

- a. Bertugas sebagai penyedia barang dan jasa agar proses produksi berjalan dengan baik.
- b. Melakukan perbandingan harga satu barang dengan yang lainnya.
- c. Melakukan evaluasi kinerja *supplier*.
- d. Melakukan pemilihan/seleksi rekanan *supplier* sesuai kinerja perusahaan.

- e. Membuat laporan penerimaan barang.
6. Departemen *Marketing*
- a. Bertugas untuk melakukan analisa konsumen dan mencari pesanan untuk produksi.
 - b. Mempromosikan barang yang akan dipasarkan.
 - c. Mencatat barang yang akan dipasarkan.
 - d. Mencatat seluruh jadwal yang berhubungan dengan pemasaran seperti jadwal pameran.
 - e. Bertanggungjawab atas penjualan produk.
 - f. Membangun relasi dengan mitra kerja.
7. Departemen Produksi
- a. Membangun kerjasama yang baik antar bagian produksi.
 - b. Bertanggungjawab terhadap proses produksi.
 - c. Mengatur jalannya operasional sehingga berjalan dengan lancar.
 - d. Mengawasi proses produksi.
 - e. Bertugas untuk menjamin kondisi mesin dapat beroperasi dengan baik dan membuat proses produksi berjalan dengan baik dan efisien.
 - f. Memastikan bahan baku yang diterima sesuai dengan yang dibutuhkan.
8. Departemen PPIC (*Production Planning Inventory Control*)
- a. Bertugas untuk mempersiapkan, merencanakan dan melakukan produksi yang bersifat *job order* (pesanan tertentu).
 - b. Mencatat seluruh jadwal produksi pengiriman barang.
 - c. Mencatat bahan baku dan bahan penolong yang sudah terpakai dan yang masih tersisa di gudang.
9. Departemen HRD (*Human Resources Development*) dan *General Affair*
- Bertugas dalam hal penanganan sumber daya manusia seperti perekrutan karyawan baru dan peningkatan *skill* karyawan.
10. Departemen *Accounting* dan *Finance*
- a. Bertugas menangani semua laporan keuangan perusahaan.

- b. Membuat seluruh laporan keuangan perusahaan.
- c. Mengatur keluar masuknya keuangan perusahaan.
- d. Memeriksa keuangan perusahaan.

4.7 Produk

Philip Kotler (2012) mendefinisikan produk adalah sesuatu yang dapat ditawarkan ke pasar untuk diperhatikan, dimiliki, dipakai atau dikonsumsi sehingga dapat memuaskan keinginan atau kebutuhan). Produk bisa berupa benda fisik, jasa, orang, tempat, organisasi dan gagasan (ide). PT DRA *Component* Persada sendiri telah menghasilkan banyak produk seperti *screw*, *bolt*, *nut*, dan *pin*, dan lain-lain. Produk yang dihasilkan berdasarkan spesifikasi permintaan konsumen. Konsumen dapat mengirimkan spesifikasi berbentuk gambar atau contoh bentuk asli dari barang dari produk yang diinginkan ataupun spesifikasi berdasarkan perkiraan dari konsumen. Apabila konsumen tidak memberikan spesifikasi berupa gambar, maka departemen produksi yang akan membuat gambar desainnya. Setelah mendapatkan persetujuan dari konsumen, maka bagian produksi akan mulai memproduksi barang sesuai dengan gambar desain yang telah ditentukan. Hal tersebut untuk memudahkan dalam hal pencatatan.

Berikut produk yang ada pada PT DRA *Component* Persada dapat dilihat dalam tabel IV.5 berikut:

Tabel IV.5 Produk

Nama Produk	Gambar
<i>Screw</i>	
<i>Bolt</i>	

<i>Nut</i>	
<i>Pin</i>	

(Sumber: PT DRA Component Persada, 2018)

4.8 Penjualan

Penjualan merupakan salah satu kegiatan pemasaran. Adanya penjualan dapat terciptakan suatu proses pertukaran barang atau jasa antara penjual dengan pembeli. Di dalam perekonomian kita atau ekonomi uang, seseorang yang menjual sesuatu akan mendapatkan imbalan berupa uang. Dengan alat penukar berupa uang, orang akan lebih mudah memenuhi segala keinginannya dan penjualan menjadi lebih mudah dilakukan (Danang Sunyoto, 2012). Oleh karena itu, proses penjualan memiliki peran yang penting di dalam suatu perusahaan, karena suatu perusahaan tidak mampu bertahan lama tanpa adanya pemasukan yang didapat secara berkelanjutan. Adapun kegiatan penjualan produk yang terdapat di PT DRA Component Persada berada dibawah tanggung jawab Departemen *marketing*. Bagian *marketing* tidak hanya menangani masalah penjualan produk tetapi juga menjaga hubungan baik dengan *customer*.

4.9 Dokumen Sistem Penjualan Produk

Sistem penjualan produk di PT DRA Component Persada yaitu *make to order* dimana perusahaan akan melakukan produksi dan pengiriman produk apabila ada permintaan dari *customer*. Kegiatan penjualan produk dimulai dari adanya permintaan produk dari *customer* berupa *form purchase order*. Setelah itu Departemen *marketing* akan memberikan surat balasan *sale confirmation* ke *customer* tersebut melalui *send by email* berupa kata-kata persetujuan dan

kesanggupan melakukan produksi dan waktu pengiriman atas produk yang dipesan. Kemudian Departemen *marketing* memberikan data pesanan *customer* ke Departemen produksi berupa *form purchase order* yang diberikan *customer* agar segera melakukan proses produksi.

Setelah semua proses produksi telah selesai, Departemen *marketing* akan membuat surat *delivery order* yang akan diberikan ke *customer* sebagai dokumen untuk melakukan proses pengiriman. Dalam proses pengiriman produk, Departemen *marketing* akan mengirimkan lampiran dokumen berupa *form purchase order* yang diberikan *customer* dan *delivery order* sebagai surat penyerahan produk. Lalu *Customer* akan menerima produk beserta dengan lampiran dokumen yang telah dikirim. Terdapat 2 arus dokumen yaitu dokumen masukan dan dokumen keluaran dalam sistem penjualan produk pada Departemen *marketing*.

4.9.1 Dokumen Masukan

Dokumen masuk yang terlibat dalam sistem penjualan produk ke bagian *Marketing* adalah sebagai berikut:

1. *Purchase Order Customer*

Purchase Order customer merupakan sebuah *form* yang diberikan oleh *customer* untuk menunjukkan produk yang ingin mereka pesan. *Form PO customer* ini terdiri dari dua rangkap, dokumen pertama akan disimpan sebagai arsip pada bagian *marketing* dan dokumen kedua diserahkan kepada bagian produksi. Untuk satu *customer* dapat memberikan lebih dari satu *form PO customer*. Hal tersebut tergantung dari waktu pemesanannya, karena satu *customer* bisa lebih dari satu kali melakukan pemesanan dan setiap barang diproduksi berdasarkan *PO customer* yang diterima. *Form Purchase Order Customer* dapat dilihat pada Gambar IV.4

berikut:

Purchase Order

PT. DRA COMPONENT PERSADA
Jl. Raya Karawang Timur 1
Desa U-04, Karawang

PO Number: 00001001000000000000

PO Date: 10 Nov 2017

PO To: PT. DRA COMPONENT PERSADA
Jl. Raya Karawang Timur 1
Desa U-04, Karawang

Item	Description	Qty	Unit Price	Unit %	Amount
0017000001	Screen 57.5 2 015 1	1			190,000

Total Order: 211,200

Remarks: 1. payment 30 days after received item and TV
2. DRA JUDRY 1711 100017
3. MARK ON LABEL FORM M00004-0001

Approved By: [Signature]
Date: [Blank]

Reviewed By: [Signature]
Date: [Blank]

Reviewed By: [Signature]
Date: [Blank]

Reviewed By: [Signature]
Date: [Blank]

Reviewed By: [Signature]
Date: [Blank]

1. Vendor 2. Finance 3. Production 4. Sales

Gambar IV.4 *Purchase Order*
(Sumber: PT DRA Component Persada, 2018)

Deskripsi dari *form PO customer* adalah sebagai berikut:

- Asal dokumen : *Customer*
- Tujuan dokumen : Departemen *Marketing* PT DRA Component Persada
- Frekuensi dokumen : Diterima berdasarkan pemesanan *customer*,

± 15 kali/hari.

Sedangkan keterangan *field* pada dokumen *form PO customer* adalah sebagai berikut:

- *PO Number* : Berisi nomor PO yang telah ditetapkan oleh *customer*
- *PO Date* : Berisi tanggal diterimanya PO *customer*.
- *Expected Date* : Berisi tanggal perkiraan waktu pengiriman produk.
- *Item* : Berisi kode barang
- *Description* : Berisi deskripsi produk yang dipesan.
- *Quantity* : Berisi jumlah produk yang dipesan
- *Unit Price* : Berisi harga satuan produk
- *Disc* : Berisi potongan harga yang diberikan.
- *Tax* : Berisi pajak yang telah ditentukan 10%
- *Amount* : Berisi jumlah tagihan yang harus dibayar oleh *customer*.

4.9.2 Dokumen Keluaran

Dokumen keluar yang terlibat dalam sistem penjualan produk ke bagian *Marketing* adalah sebagai berikut:

1. Sale Confirmation

Sale Confirmation merupakan sebuah surat balasan yang dikirim oleh Departemen *Marketing* yang bertujuan sebagai persetujuan atas produk yang dipesan oleh *customer*. *Sale confirmation* ini dikirim melalui *send by email*. *Sale Confirmation* dapat dilihat pada Gambar IV.5 berikut:

PT. DRA Component Persada
 Jl. Raya Cileungsi Setu, Kampung Pasirangin, RT.001/004 Desa Pasirangin,
 Kec. Cileungsi, Kab. Bogor – Jawa Barat.

6 Januari 2017

No :
 Subject: Sale Confirmation tanggal 1 Januari 2017

Purchase Manager
 PT. HARTONO ISTANA TEKNOLOGI
 JL. KHR. Asnawi PO BOX 126,
 Kaliwungu, Kudus - 59332

Dengan Hormat,
 Terima Kasih atas pesanan dengan No. PO 223 pada tanggal 1 Januari 2017 sebagai berikut:

No	Tipe Barang	Qty	Harga/unit	Total Harga	Keterangan

Pesanan sedang dalam proses produksi dan akan di kirim pada tanggal 29 Januari 2017. Kami selalu menjaga kualitas produk hingga proses pengiriman. Kami harap anda merasa puas dengan produk serta pelayanan kami.

Kami sangat berharap, jika Perusahaan anda tetap memperpanjang kerja-sama dengan kami di masa depan.

Catatan:

Regards

Marketing Manager
 PT DRA Component Persada

Gambar IV.5 *Sale Confirmation*
 (Sumber: Hasil Rekonstruksi Penulis, 2018)

Deskripsi dari *Sale Confirmation* adalah sebagai berikut:

- Asal dokumen : Departemen *Marketing* PT DRA Component Persada
- Tujuan dokumen : *Customer*
- Frekuensi dokumen : Dibuat berdasarkan PO *customer*,
 ± 15 kali/hari.

Sedangkan keterangan *field* pada dokumen *Sale Confirmation* adalah sebagai berikut:

- No : Berisi nomor urutan barang yang dipesan
- Tipe Barang : Berisi tipe atau nama produk yang dipesan
- *Quantity* : Berisi jumlah produk yang dipesan
- Harga/unit : Berisi satuan produk/unit

- Total Harga : Berisi total harga produk yang harus dibayar *customer*
- Keterangan : Berisi konfirmasi pengiriman dan keterangan stok Produk yang dipesan

2. Delivery Order

Delivery Order merupakan sebuah *form* yang digunakan sebagai surat jalan atau dokumen penyerahan produk berdasarkan PO *customer* yang akan diberikan ke *customer* dan sebagai bukti pengeluaran barang. *Form delivery order* ini terdiri dari dua rangkap, dokumen pertama akan disimpan sebagai arsip pada bagian *marketing* dan dokumen kedua diserahkan kepada *customer*. Untuk satu PO *customer* bisa dibuatkan lebih dari satu *form delivery order*. Hal tersebut dikarenakan proses pengiriman barang dilakukan secara ber-angsur sebanyak 2 kali yaitu di awal dan sisanya di akhir. *Delivery Order* dapat dilihat pada Gambar

IV.6 berikut:

Gambar IV.6 *Delivery Order*
(Sumber: PT DRA Component Persada, 2018)

Deskripsi dari *Delivery Order* adalah sebagai berikut:

- Asal dokumen : Departemen *Marketing* PT DRA Component Persada
- Tujuan dokumen : *Customer*
- Frekuensi dokumen : Dibuat berdasarkan PO *customer*,
± 90 kali/minggu.

Sedangkan keterangan *field* pada dokumen *Delivery Order* adalah sebagai berikut:

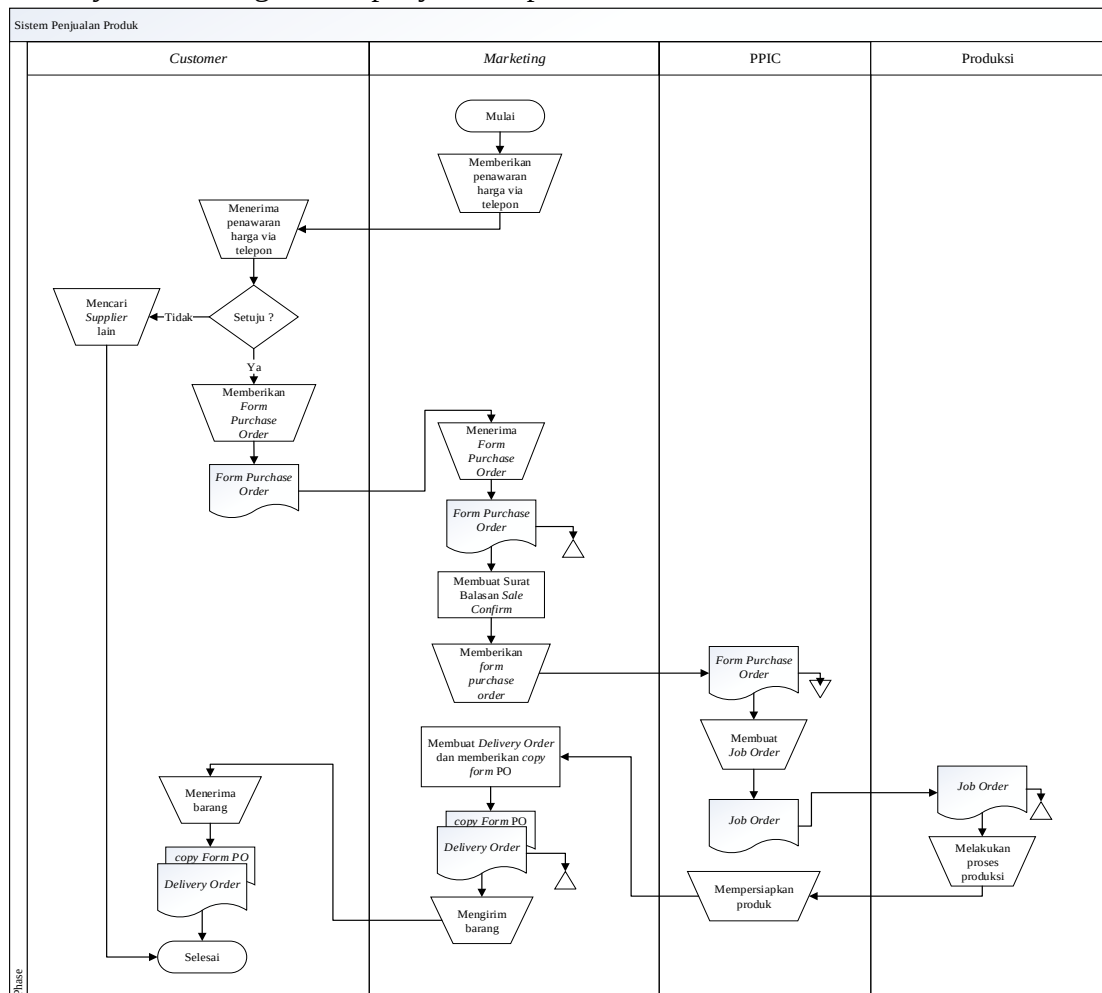
- *Delivery Date* : Berisi tanggal *delivery*
- *DO Number* : Berisi nomor *delivery order*
- *No* : Berisi nomor urutan produk berdasarkan PO *customer*
- *Description* : Berisi deskripsi produk berdasarkan PO *customer*
- *Quantity* : Berisi jumlah produk yang dipesan
- *Packing* : Berisi bahan yang digunakan untuk *packing*

4.10 Prosedur Penjualan Produk

Pada prosedur penjualan produk, pada PT DRA Component Persada mempunyai beberapa tahapan/langkah untuk menjual produk yang dihasilkan, ada beberapa prosedur sistem informasi penjualan produk yang dijelaskan pada *flowmap* pada gambar IV.9:

1. Staf *marketing* memberikan Penawaran Harga kepada *customer* via telepon
2. Jika *customer* setuju maka akan memberikan *Form Purchase Order* ke Departemen *Marketing*, namun jika tidak setuju akan mencari *supplier* lain.
3. Staf *marketing* menerima *Form Purchase Order* dari *customer*.
4. Staf *marketing* memberikan Surat Balasan *Sale Confirmation* kepada *customer* melalui *send email* berupa kata-kata persetujuan atas produk yang dipesannya.
5. Staf *marketing* memberikan data permintaan produk ke Departemen PPIC berupa *Form Purchase Order customer* lalu PO tersebut diolah menjadi *Job order*.

6. Staf PPIC memberikan *form Job Order* ke Departemen Produksi. Dimana *form job order* tersebut berfungsi sebagai permintaan pembuat produk. Lalu staf produksi melakukan produksi berdasarkan *job order* yang diterima.
7. Setelah semua proses produksi telah selesai, staf PPIC akan mempersiapkan produk yang akan dilakukan pengiriman. Kemudian staf *marketing* akan membuat *delivery order* sebagai dokumen untuk melakukan pengiriman.
8. Dalam proses pengiriman produk, staf *marketing* akan mengirimkan lampiran dokumen berupa *copy form purchase order* yang diberikan *customer* dan *delivery order* sebagai surat penyerahan produk



Gambar IV.7 Aliran Prosedur Sistem Informasi Penjualan Produk saat ini

(Sumber: Hasil Analisis, 2018)

4.11 Analisis Permasalahan

Pada sistem penjualan produk yang berjalan di Departemen *Marketing* terdapat beberapa masalah pada proses bisnis yang biasa terjadi. Adapun masalah-masalah tersebut dirumuskan menggunakan diagram PIECES. Tujuan dari analisis PIECES ini adalah untuk mengoreksi atau memperbaiki sistem dari kriteria yang ditetapkan. Berikut ini merupakan diagram PIECES yang memuat masalah yang ada pada proses penjualan produk di PT DRA Component Persada.

Tabel IV.6 Diagram PIECES

Analisis	Uraian Masalah	Sebab	Akibat	Solusi
<i>Performance</i> (Kinerja)	Untuk proses penawaran harga yang dilakukan oleh satu orang memakan waktu seharian seharusnya bisa selesai dalam 1-3 jam.	Belum adanya sistem yang terhubung dengan <i>database</i> yang membantu proses peng- <i>input-an</i> surat penawaran harga	<i>User</i> harus menghubungi <i>customer</i> satu per satu via telepon sehingga proses penawaran harga memakan waktu yang lumayan lama	Pembuatan sistem yang terhubung dengan database yang dapat membantu proses peng- <i>input-an</i> surat penawaran harga
<i>Information</i> (Informasi)	Data penjualan yang kurang terorganisir dengan baik. Contoh: Data yang sama di	Masih menggunakan <i>Microsoft Excel</i>	Terjadinya redundansi data dan kesulitan dalam melakukan pembaruan	Membuat proses peng- <i>input-an</i> data penjualan menjadi

Tabel IV.6 Diagram PIECES (lanjutan)

Analisis	Uraian Masalah	Sebab	Akibat	Solusi
-----------------	-----------------------	--------------	---------------	---------------

	simpan di banyak tempat.		informasi penjualan	Terkomputerisasi dan tersimpan di dalam <i>database</i> .
	Proses pengajuan permintaan produksi masih belum terkomputerisasi dan memakan cukup waktu. Contoh: pengajuan permintaan produksi masih berupa kertas <i>copy-an PO</i>	Pengajuan permintaan produksi masih menggunakan kertas	Proses pengajuan permintaan produksi memakan cukup waktu	Membuat pengajuan permintaan produksi menjadi terkomputerisasi yang terhubung dengan <i>database</i>
	Pada proses peng- <i>input-an</i> Surat Jalan,	Data yang berhubungan dengan penjualan	Saat memasukan nama barang dan <i>customer</i> , <i>user</i> harus	Pembuatan sistem peng- <i>input-an</i> Surat Jalan yang

Tabel IV.6 Diagram PIECES (lanjutan)

Analisis	Uraian Masalah	Sebab	Akibat	Solusi
-----------------	-----------------------	--------------	---------------	---------------

	data di- <i>input</i> secara <i>manual</i>	seperti nama barang dan <i>customer</i> pada <i>database</i> tidak terhubung langsung dengan sistem penjualan produk	melihat <i>database</i> terlebih dahulu kemudian menginputkanny a pada surat jalan	dapat terhubung dengan <i>database</i> .
<i>Economics</i> (Biaya)	-	-	-	-
<i>Control</i> (Kontrol)	Sistem penjualan produk dapat diakses oleh siapa saja	Data penjualan produk dikelola dengan Microsoft Excel, dalam mengelola data tersebut tidak dilindungi hak akses.	Data penjualan produk dapat dilihat dan diakses oleh siapa saja yang tidak berwenang sehingga bisa terjadi manipulasi data	Membuat sistem penjualan produk yang memiliki hak akses.
<i>Efficiency</i> (Efisiensi)	-	-	-	-

Tabel IV.6 Diagram PIECES (lanjutan)

Analisis	Uraian Masalah	Sebab	Akibat	Solusi
-----------------	---------------------------	--------------	---------------	---------------

<i>Service</i> (Layanan)	Lama dalam proses pengelolaan data penjualan. Contoh: <i>User</i> harus bolak balik dalam meng- <i>input</i> data, memindahkan data dan menyimpannya kembali ke lemari arsip	Dalam proses peng- <i>input</i> -an data penjualan masih <i>Microsoft Excel</i>	<i>User</i> harus meng- <i>input</i> -kan data penjualan di setiap proses secara berulang	Membuat sistem menjadi terhubung dengan <i>database</i> sehingga memudahkan dalam pengelolaan dokumen.
-----------------------------	--	---	---	--

(Sumber: Hasil Analisis, 2018)

Pada analisis permasalahan di bidang *economics* dan *efficiency* tidak cukup data yang ditemukan untuk melihat permasalahan yang berhubungan dengan sistem penjualan produk.

BAB V

ANALISIS DAN PEMBAHASAN

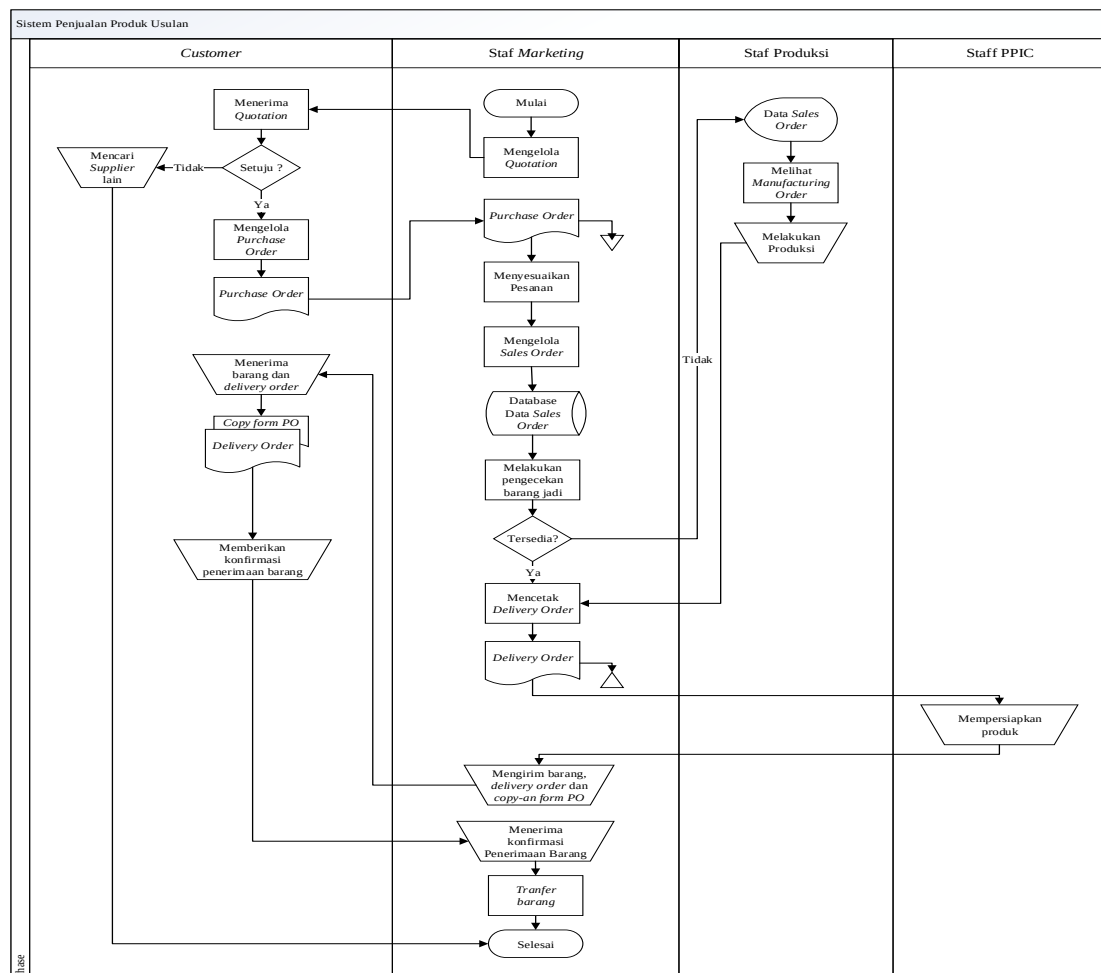
5.1 Prosedur Sistem Penjualan Produk Usulan

Prosedur sistem penjualan produk yang diusulkan pada PT DRA Component Persada:

1. Dimulai dengan staf *marketing* melakukan pengelolaan *quotation* berupa penawaran harga atas produk-produk yang akan dijual kemudian dikirimkan kepada satu atau beberapa *customer* melalui email
2. *Customer* menerima *quotation* tersebut, jika *customer* menyetujui untuk melakukan pembelian maka akan melakukan pengelolaan *Form Purchase Order* berupa *list* produk-produk yang akan dibeli kemudian dikirimkan ke Departemen Marketing, namun jika tidak setuju akan mencari supplier lain.
3. Staf *marketing* menerima *Form Purchase Order* dari *customer* secara manual kemudian melakukan konfirmasi penjualan dengan menyesuaikan pesanan pada *Form quotation* berdasarkan *Form Purchase Order* tersebut yang akan *ter-generate* menjadi *Form Sales Order*
4. Staf *marketing* melakukan pengelolaan *Sales Order* yang akan *ter-generate* menjadi *Form Delivery Order*. Sebelum melakukan pengiriman, staf *marketing* terlebih dahulu melihat data *Sales Order* pada sistem untuk menentukan *sales order* mana yang akan dilakukan pengiriman.
5. Setelah itu staf *marketing* melakukan pengecekan ketersediaan barang. *Sales Order* akan *ter-generate* menjadi *Manufacturing Order* jika tidak ada stok barang yang tersedia
6. Staf produksi dapat melihat *Manufacturing Order* berdasarkan *Sales Order* beserta runtutan proses produksi dan kebutuhan produk yang harus digunakan dan dikerjakan kemudian melakukan produksi secara manual.
7. Setelah semua proses produksi telah selesai dan produk siap dikirim, Staf *marketing* akan mencetak *delivery order* dan staf PPIC akan mempersiapkan produk yang akan dikirim. Dalam proses pengiriman produk secara manual,

staf marketing akan mengirimkan lampiran dokumen berupa copy form purchase order yang diberikan customer dan delivery order sebagai surat penyerahan produk

8. Setelah barang telah sampai kepada customer, customer akan melakukan konfirmasi penerimaan barang melalui telepon kemudian staf marketing akan melakukan transfer barang secara sistem sehingga status penjualan tersebut sudah transferred.



Gambar V.1 Aliran Prosedur Sistem Informasi Penjualan Produk usulan

(Sumber: Hasil Analisis, 2018)

5.2 Analisis Kebutuhan Sistem Informasi

Pada tahap ini akan dilakukan analisis kebutuhan sistem. Analisis kebutuhan ini diambil dari permasalahan yang terjadi pada sistem penjualan produk yang

terjadi di PT DRA Component Persada. Analisis kebutuhan sistem dapat dilihat pada tabel V.1 berikut:

Tabel V.1 Analisis Kebutuhan Sistem

Permasalahan	Kebutuhan User	Functional Requirement	Uraian
Pemrosesan penawaran harga yang dilakukan oleh satu orang memakan waktu sehari-harian seharusnya bisa selesai dalam 1-3 jam.	Pengelolaan data yang terkomputerisasi yang membantu proses penawaran harga	- Mengelola data <i>customer</i> - Mengelola data produk - Mengelola data <i>Quotation</i> atau surat penawaran harga - Mengirimkan <i>quotation</i> kepada <i>customer</i> - konfirmasi <i>quotation</i> - Mengelola data <i>Sales order</i>	- Data <i>customer</i>, produk dan <i>quotations</i> dapat menambah, mengubah, menghapus dan mencari data. - Konfirmasi <i>quotations</i> dilakukan setelah terjadi kesepakatan dengan <i>customer</i>. <i>Quotations</i> berdasarkan PO <i>customer</i> akan ter-<i>generate</i> menjadi <i>Sales Order</i>
Proses pengajuan permintaan produksi belum	Pengelolaan data yang mudah, cepat dan	Melihat <i>Manufacturing Order</i>	Data <i>sales order</i> akan ter- <i>generate</i> menjadi

Tabel V.1 Analisis Kebutuhan Sistem (Lanjutan)

Permasalahan	Kebutuhan User	Functional Requirement	Uraian
---------------------	-----------------------	-------------------------------	---------------

terkomputerisasi dan memakan waktu karena pada proses tersebut masih berupa <i>copy-an form PO customer</i> dan harus diolah menjadi <i>Job Order</i>	terintegrasi antara Departemen <i>Marketing</i> dan Departemen Produksi		<i>Manufacturing Order</i> (<i>List permintaan produksi</i>) apabila barang yang dibutuhkan tidak tersedia di gudang.
Proses peng- <i>input-an</i> Surat Jalan, data di <i>input</i> secara <i>manual</i>	Pengelolaan surat jalan yang terhubung langsung dengan sistem penjualan produk	- Mengelola <i>delivery order</i> - Mengecek ketersediaan stok	- Data <i>delivery order</i> dapat mengubah jumlah produk
Data-data penjualan produk dapat diakses oleh yang tidak berwenang.	Sistem komputerisasi yang setiap karyawan memiliki akses masing-masing.	Akses <i>Login</i> dengan <i>email</i> dan <i>password</i> .	Memvalidasi <i>email</i> dan <i>password</i> yang dimasukkan dan menampilkan pesan kesalahan jika ada ketidaksesuaian.

(Sumber: Hasil Analisis, 2018)

5.2.1 *Functional Requirement*

Functional Requirement dari sistem penjualan produk yang diusulkan antara lain:

1. Staf *marketing* mengelola data *customer*, produk, *quotation*, *sales order* dan *delivery order*
2. Staf *marketing* melakukan konfirmasi *Quotation*.

3. Staf *marketing* mengirimkan *Quotation* (penawaran harga) dan *delivery order* kepada *customer*.
4. Staf produksi dapat melihat *Manufacturing Order*
5. Staf *marketing* dapat mencetak *Delivery Order*
6. Staf *marketing* dan staf produksi melakukan *login*.

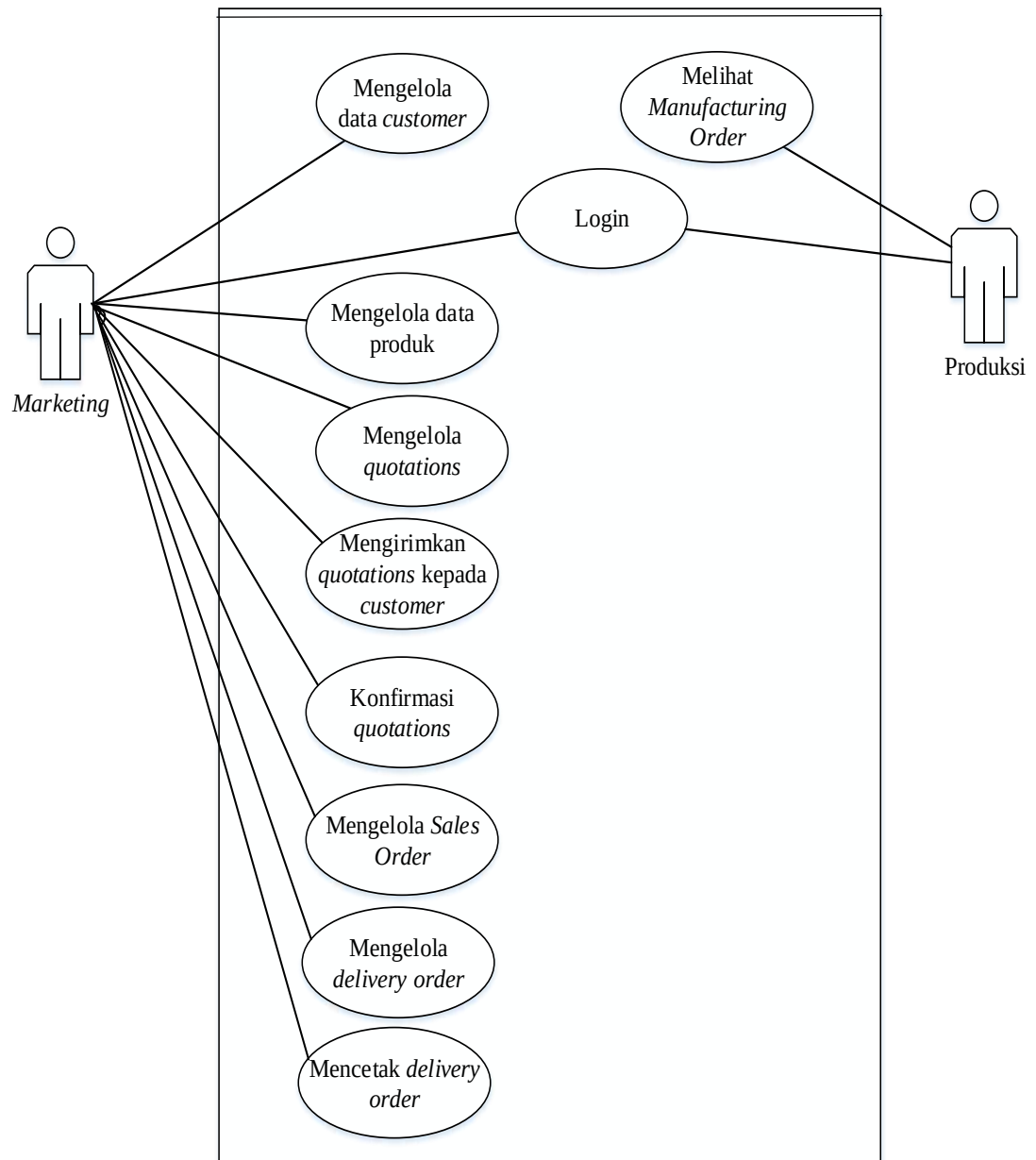
5.2.2 Non Functional Requirement

Berdasarkan hasil pengolahan data dan analisis sistem, maka dibutuhkan perancangan sistem informasi penjualan produk Departemen *Marketing* pada PT DRA Component Persada. Berikut adalah daftar kebutuhan *Non Functional requirement* sistem untuk sistem informasi penjualan produk:

1. *Database* dapat di *backup* sehingga *user* dapat mengurangi resiko kehilangan data dan kerusakan sistem.
2. Adanya *login* menggunakan *email* dan *password* untuk membedakan pengguna termasuk hak akses masing-masing sehingga terlindung dari penggunaan akses yang tidak berwenang.
3. Sistem dapat diakses menggunakan beberapa *web browser* seperti Internet Explore, Google Chrome, dan Mozilla Firefox.
4. Tampilan yang mudah dipahami sehingga tidak memakan waktu dalam memahami atau mempelajari sistem tersebut.

5.3 Use Case Diagram

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Gambar V.1 menjelaskan tentang *use case diagram* usulan sistem penjualan produk.



Gambar V.2 Use Case Diagram Usulan Sistem Penjualan Produk
(Sumber: Hasil Analisis, 2018)

5.3.1 Use Case Description

Use case description yang merupakan gambaran detail dari *use case* yang berisi nama *use case*, deskripsi *use case*, aktor dan *normal flow event*. *Use case description* dari *use case diagram* sistem penjualan produk usulan adalah sebagai berikut:

1. Use Case Description Login

Tabel V.2 *Use Case Description Login*

Nama <i>Use Case</i>	<i>Login</i>
Aktor	Staf <i>Marketing</i> dan Staf Produksi
Deskripsi	<i>Use case</i> ini menggambarkan staf <i>Marketing</i> dan staf
<i>Normal Flow of Events</i>	1. <i>User</i> membuka aplikasi. 2. Sistem menampilkan <i>window login</i>. 3. <i>User</i> memilih <i>database</i>. 4. <i>User</i> memasukan <i>email</i> dan <i>password</i>. 5. <i>User</i> mengklik tombol <i>login</i>. 6. Sistem mengecek ke <i>database</i>, apakah <i>email</i> dan <i>password</i> sesuai. 7. Jika <i>email</i> dan <i>password</i> benar, maka muncul tampilan menu utama. Jika <i>email</i> dan <i>password</i> salah, muncul pesan “<i>Wrong login/password</i>” pada <i>window login</i> dan coba <i>login</i> kembali.

(Sumber: Hasil Analisis, 2018)

2. *Use Case Description Mengelola Data Customer*

Tabel V.3 *Use Case Description Mengelola Data Customer*

Nama <i>Use Case</i>	Mengelola Data <i>Customer</i>
Aktor	Staf <i>Marketing</i>
Deskripsi	<i>Use case</i> ini menggambarkan Staf <i>Marketing</i> dapat mengelola data <i>customer</i> di dalam sistem.

Tabel V.3 *Use Case Description Mengelola Data Customer*

<i>Normal Flow of Events</i>	1. <i>User</i> masuk ke tampilan menu utama. 2. <i>User</i> memilih menu <i>Sales</i>. 3. <i>User</i> memilih submenu <i>Customer</i>. 4. <i>User</i> dapat memilih aksi tambah, cari, ubah dan hapus data.
------------------------------	--

	5. Jika memilih aksi tambah data, <i>user</i> akan mengklik tombol <i>create</i> maka sistem akan menampilkan form tambah data. 6. <i>User</i> memasukkan data, lalu mengklik tombol <i>save</i>. Sistem akan menyimpan data ke dalam <i>database</i>. 7. Jika memilih aksi cari data maka <i>user</i> memasukan nama <i>customer</i> pada kolom pencarian. Sistem akan menampilkan data yang dimaksud. 8. Jika memilih aksi ubah data maka <i>user</i> memilih data yang akan diubah, lalu mengklik tombol <i>edit</i>. 9. <i>User</i> mengubah data lalu mengklik tombol <i>save</i>. Sistem akan mengubah data di dalam <i>database</i>. 10. Jika memilih aksi hapus data maka <i>user</i> memilih data yang ingin dihapus lalu mengklik tombol <i>more</i>, kemudian pilih <i>delete</i>. Sistem akan menghapus data dari <i>database</i>.
--	--

(Sumber: Hasil Analisis, 2018)

3. Use Case Description Mengelola Data Produk

Tabel V.4 Use Case Description Mengelola Data Produk

Nama Use Case	Mengelola Data Produk
Aktor	Staf Marketing
Deskripsi	Use case ini menggambarkan Staf Marketing dapat

Tabel V.4 Use Case Description Mengelola Data Produk

	mengelola data produk di dalam sistem.
Normal Flow of Events	1. <i>User</i> masuk ke tampilan menu utama. 2. <i>User</i> memilih menu <i>Sales</i>. 3. <i>User</i> memilih submenu <i>Product</i>. 4. <i>User</i> dapat memilih aksi tambah, cari, ubah dan hapus data

	5. Jika memilih aksi tambah data, <i>user</i> akan mengklik tombol <i>create</i> maka sistem akan menampilkan form tambah data 6. <i>User</i> memasukkan data, lalu mengklik tombol <i>save</i>. Sistem akan menyimpan data ke dalam <i>database</i>. 7. Jika memilih aksi cari data maka <i>user</i> memasukan nama produk pada kolom pencarian. Sistem akan menampilkan data yang dimaksud. 8. Jika memilih aksi ubah data maka <i>user</i> memilih data yang akan diubah, lalu mengklik tombol <i>edit</i>. 9. <i>User</i> mengubah data lalu mengklik tombol <i>save</i>. Sistem akan mengubah data di dalam <i>database</i>. 10. Jika memilih aksi hapus data maka <i>user</i> memilih data yang ingin dihapus lalu mengklik tombol <i>more</i>, kemudian pilih <i>delete</i>. Sistem akan menghapus data dari <i>database</i>.
--	--

(Sumber: Hasil Analisis, 2018)

4. Use Case Description Mengelola data Quotation

Tabel V.5 Use Case Description Mengelola data Quotation

Nama Use Case	Mengelola data <i>Quotations</i> atau surat penawaran harga
Aktor	Staf <i>Marketing</i>

Tabel V.5 Use Case Description Mengelola data Quotation

Deskripsi	<i>Use case</i> ini menggambarkan Staf <i>Marketing</i> dapat mengelola data <i>quotation</i> di dalam sistem.
<i>Normal Flow of Events</i>	1. <i>User</i> masuk ke tampilan menu utama. 2. <i>User</i> memilih menu <i>Sales</i>. 3. <i>User</i> memilih submenu <i>Quotation</i>. 4. <i>User</i> dapat memilih aksi tambah, cari, ubah dan hapus data.

	5. Jika memilih aksi tambah data, <i>user</i> akan mengklik tombol <i>create</i> tambah data. 6. <i>User</i> memasukkan data, lalu mengklik tombol <i>save</i>. Sistem akan menyimpan data ke dalam <i>database</i>. 7. Jika memilih aksi cari data maka <i>user</i> memasukan nomor <i>quotation</i> pada kolom pencarian. Sistem akan menampilkan data yang dimaksud. 8. Jika memilih aksi ubah data maka <i>user</i> memilih data yang akan diubah, lalu mengklik tombol <i>edit</i>. 9. <i>User</i> mengubah data lalu mengklik tombol <i>save</i>. Sistem akan mengubah data di dalam <i>database</i>. 10. Jika memilih aksi hapus data maka <i>user</i> memilih data yang ingin dihapus lalu mengklik tombol <i>more</i>, kemudian pilih <i>delete</i>. Sistem akan menghapus data dari <i>database</i>.
--	--

(Sumber: Hasil Analisis, 2018)

5. Use Case Description Mengirimkan Quotation kepada Customer

Tabel V.6 Use Case Description Mengirimkan Quotation kepada Customer

Nama Use Case	Mengirimkan Quotation kepada Customer
Aktor	Staf Marketing

Tabel V.6 Use Case Description Mengirimkan Quotation kepada Customer

Deskripsi	<i>Use case</i> ini menggambarkan Staf Marketing dapat mengirimkan <i>quotations</i> kepada <i>customer</i> berdasarkan data <i>request for quotations</i> (surat permintaan harga) dari <i>customer</i>
Normal Flow of Events	1. <i>User</i> masuk ke tampilan menu utama 2. <i>User</i> memilih menu <i>Sales</i> 3. <i>User</i> memilih submenu <i>Quotations</i> 4. <i>User</i> memilih <i>Quotation</i> yang memiliki status “<i>Draf Quotations</i>”. 5. <i>User</i> memilih tombol <i>Send by email</i>.

(Sumber: Hasil Analisis, 2018)

6. Use Case Description Konfirmasi Quotations

Tabel V.7 Use Case Description Konfirmasi Quotations

Nama Use Case	Konfirmasi Quotations
Aktor	Staf Marketing
Deskripsi	Use case ini menggambarkan Staf Marketing dapat melakukan konfirmasi quotations berdasarkan kesepakatan antara customer dan staf marketing. Quotations akan ter-generate menjadi sales order.
Normal Flow of Events	1. User masuk ke tampilan menu utama. 2. User memilih menu Marketing. 3. User memilih submenu Quotations. 4. User memilih Quotations yang telah disetujui antara staf marketing dan customer 5. User memilih tombol Confirm Sale. 6. Quotations yang telah dikonfirmasi dapat dilihat pada submenu Sales Order.

(Sumber: Hasil Analisis, 2018)

7. Use Case Description Melihat Manufacturing Order

Tabel V.8 Use Case Description Melihat Manufacturing Order

Nama Use Case	Melihat Manufacturing Order
Aktor	Staf Produksi
Deskripsi	Use case ini menggambarkan Staf Produksi dapat melihat manufacturing order yang ter-generate dari sales order
Normal Flow of Events	1. User masuk ke tampilan menu utama. 2. User memilih menu Manufacturing. 3. User memilih submenu Manufacturing Order. 4. User melihat Manufacturing Order.

(Sumber: Hasil Analisis, 2018)

8. Use Case Description Mengelola Delivery Order

Tabel V.9 *Use Case Description* Mengelola *Delivery Order*

Nama <i>Use Case</i>	Mengelola <i>Delivery Order</i>
Aktor	Staf <i>Marketing</i>
Deskripsi	<i>Use case</i> ini menggambarkan Staf <i>Marketing</i> dapat mengelola <i>delivery order</i> di dalam sistem.
<i>Normal Flow of Events</i>	1. <i>User</i> masuk ke tampilan menu utama. 2. <i>User</i> memilih menu <i>Sales</i>. 3. <i>User</i> memilih submenu <i>Sales Order</i>. 4. <i>User</i> memilih <i>Sales Order</i> yang memiliki status “<i>Sales Order</i>”. 5. <i>User</i> memilih tombol <i>View Delivery Order</i>. 6. <i>User</i> memilih tombol <i>Recheck Availability</i>. 7. Jika produk tidak tersedia maka <i>sales order</i> ter-<i>generate</i> menjadi <i>manufacturing order</i>. 8. <i>User</i> memilih tombol <i>Force Availability</i> 9. <i>User</i> memilih tombol <i>transfer</i>

Tabel V.9 *Use Case Description* Mengelola *Delivery Order* (Lanjutan)

	10. <i>User</i> memilih tombol <i>Apply</i> 11. Status <i>Sales Order</i> akan berubah menjadi <i>Transferred</i>.
<i>Alternatif Flow of Events</i>	7a. Jika pengiriman dilakukan lebih 1 kali maka data jumlah produk yang akan dikirim dapat diubah sesuai dengan permintaan <i>customer</i> 7b. <i>User</i> mengubah jumlah produk yang akan dikirim.

(Sumber: Hasil Analisis, 2018)

9. *Use Case Description* Mengelola *Sales Order*Tabel V.10 *Use Case Description* Mengelola *Sales Order*

Nama <i>Use Case</i>	Mengelola <i>Sales Order</i>
Aktor	Staf <i>Marketing</i>

Deskripsi	<i>Use case</i> ini menggambarkan Staf <i>Marketing</i> dapat mengelola <i>Sales Order</i> di dalam sistem.
<i>Normal Flow of Events</i>	1. <i>User</i> masuk ke tampilan menu utama. 2. <i>User</i> memilih menu <i>Sales</i>. 3. <i>User</i> memilih submenu <i>Sales Order</i>. 4. <i>User</i> dapat memilih aksi tambah, cari, ubah dan hapus data. 5. Jika memilih aksi tambah data, <i>user</i> akan mengklik tombol <i>create</i> tambah data. 6. <i>User</i> memasukkan data, lalu mengklik tombol <i>save</i>. Sistem akan menyimpan data ke dalam <i>database</i>. 7. Jika memilih aksi cari data maka <i>user</i> memasukan nomor <i>sales order</i> pada kolom pencarian. Sistem akan menampilkan data yang dimaksud. 8. Jika memilih aksi ubah data maka <i>user</i> memilih

Tabel V.10 *Use Case Description* Mengelola *Sales Order* (Lanjutan)

	- data yang akan diubah, lalu mengklik tombol <i>edit</i>. 9. <i>User</i> mengubah data lalu mengklik tombol <i>save</i>. Sistem akan mengubah data di dalam <i>database</i>. 10. Jika memilih aksi hapus data maka <i>user</i> memilih data yang ingin dihapus lalu mengklik tombol <i>more</i>, kemudian pilih <i>delete</i>. Sistem akan menghapus data dari <i>database</i>.
--	--

(Sumber: Hasil Analisis, 2018)

10. *Use Case Description* Mencetak *Delivery Order*

Tabel V.11 *Use Case Description* Mencetak *Delivery Order*

Nama <i>Use Case</i>	Mencetak <i>Delivery Order</i>
Aktor	Staf <i>Marketing</i>
Deskripsi	<i>Use case</i> ini menggambarkan Staf <i>Marketing</i> dapat mencetak dokumen <i>delivery order</i> .

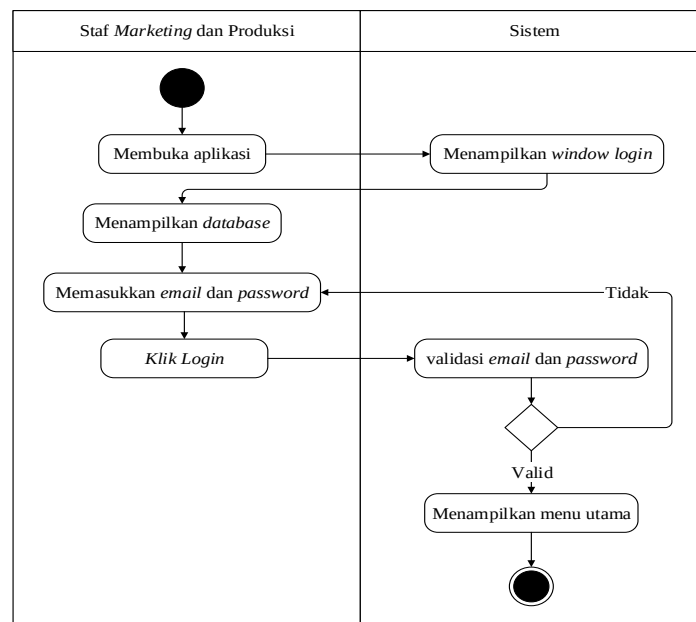
<i>Normal Flow of Events</i>	1. <i>User</i> masuk ke tampilan menu utama. 2. <i>User</i> memilih menu <i>Sales</i>. 3. <i>User</i> memilih submenu <i>Sales Order</i>. 4. <i>User</i> memilih <i>Sales Order</i> yang memiliki status “<i>Sales Order</i>”. 5. <i>User</i> memilih tombol <i>View Delivery Order</i>. 6. <i>User</i> memilih tombol <i>Print</i>. 7. <i>User</i> memilih <i>Delivery Order</i>
------------------------------	---

(Sumber: Hasil Analisis, 2018)

5.4 Activity Diagram

Activity Diagram menggambarkan urutan aktivitas proses bisnis pada suatu sistem. *Activity Diagram* sistem penjualan produk di PT DRA Component Persada yang diusulkan antara lain sebagai berikut:

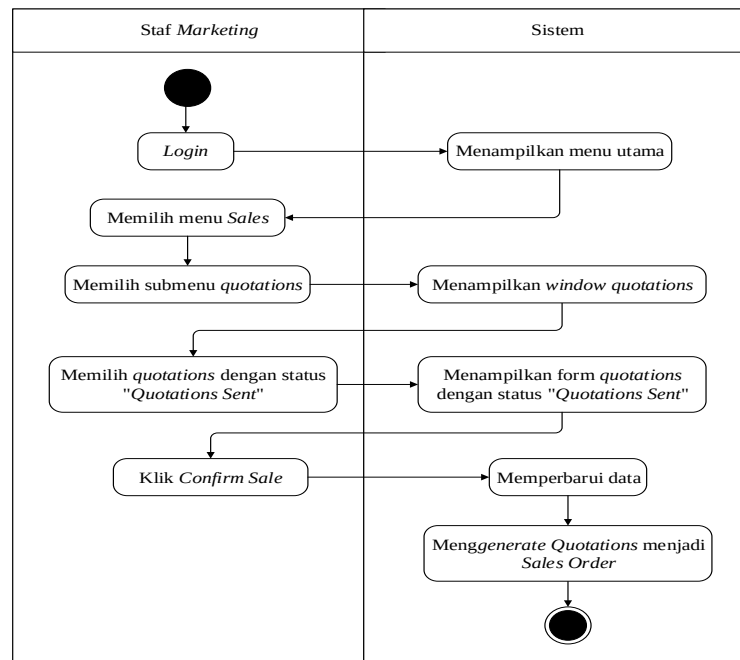
1. Activity Diagram Login



Gambar V.3 Activity Diagram Login

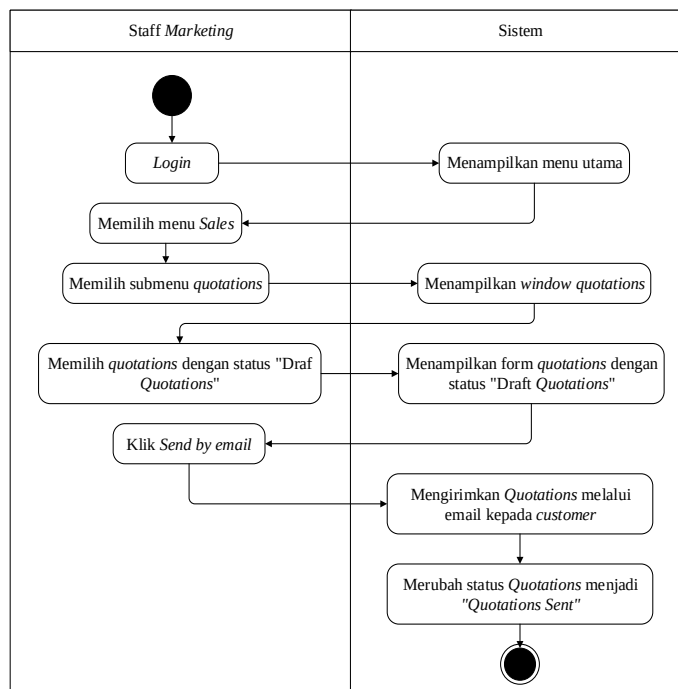
(Sumber: Hasil Analisis, 2018)

2. Activity Diagram Konfirmasi Quotation



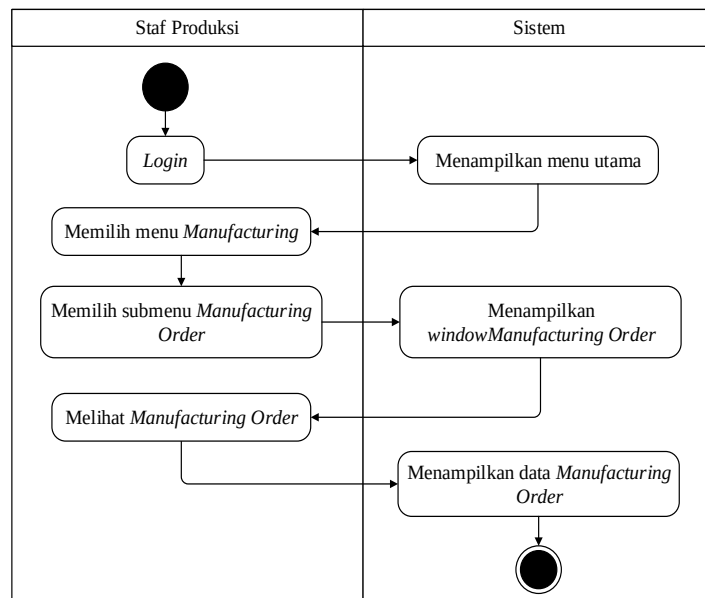
Gambar V.4 Activity Diagram Konfirmasi Quotation
(Sumber: Hasil Analisis, 2018)

3. Activity Diagram Mengirimkan Quotation



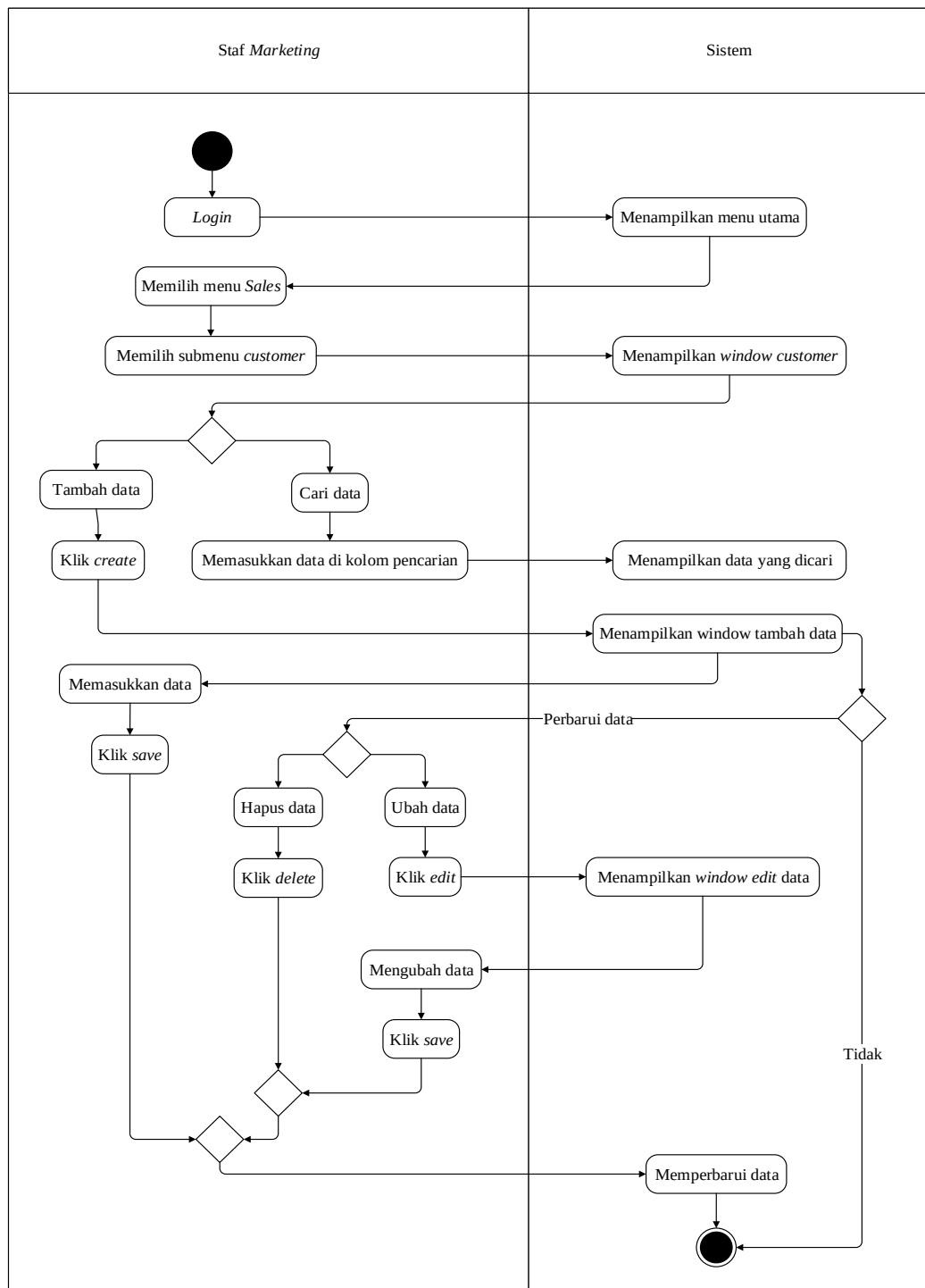
Gambar V.5 Activity Diagram Mengirimkan Quotation
(Sumber: Hasil Analisis, 2018)

4. Activity Diagram Melihat Manufacturing Order



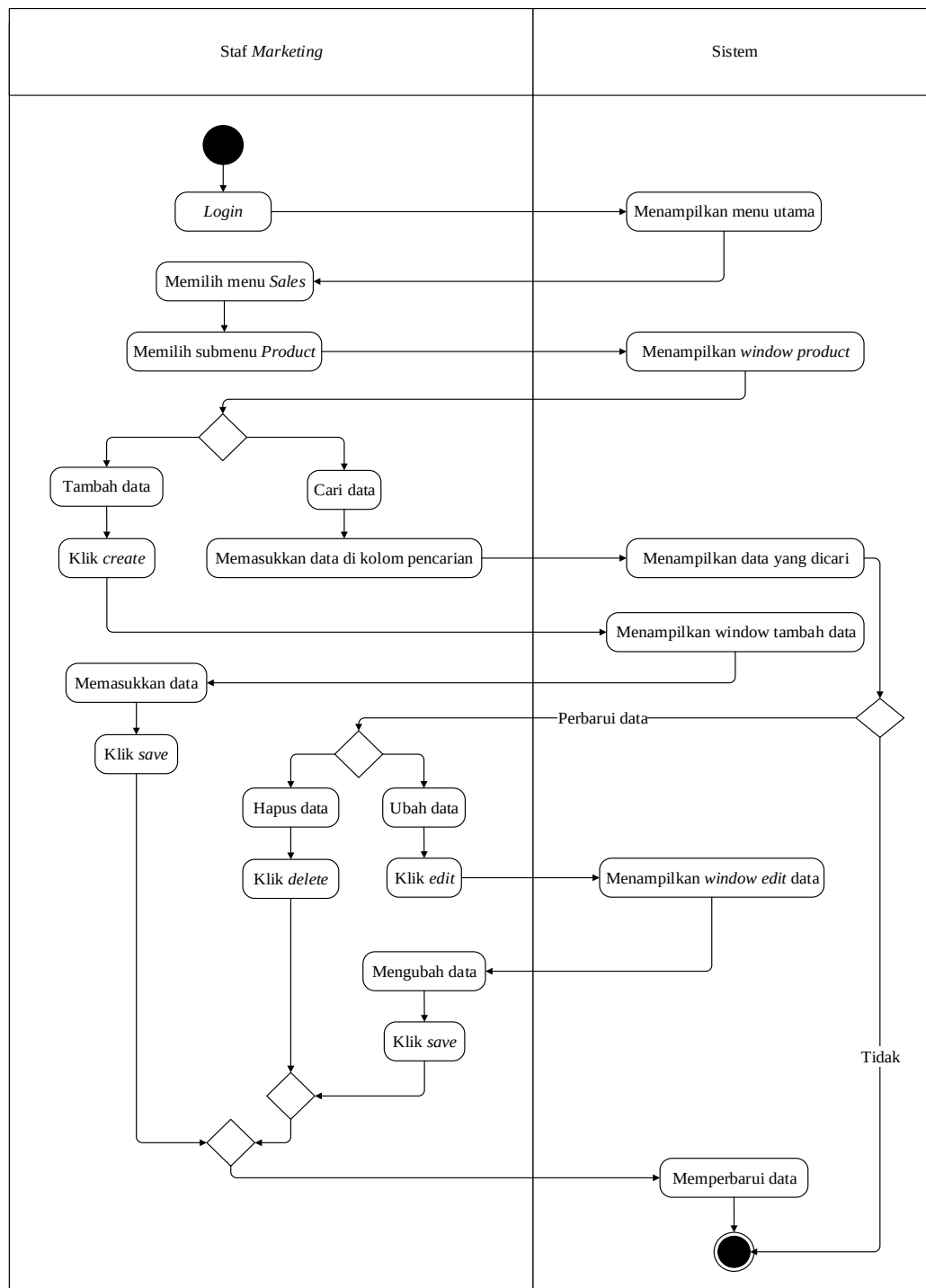
Gambar V.6 Activity Diagram Melihat Manufacturing Order
(Sumber: Hasil Analisis, 2018)

5. Activity Diagram Mengelola Data Customer



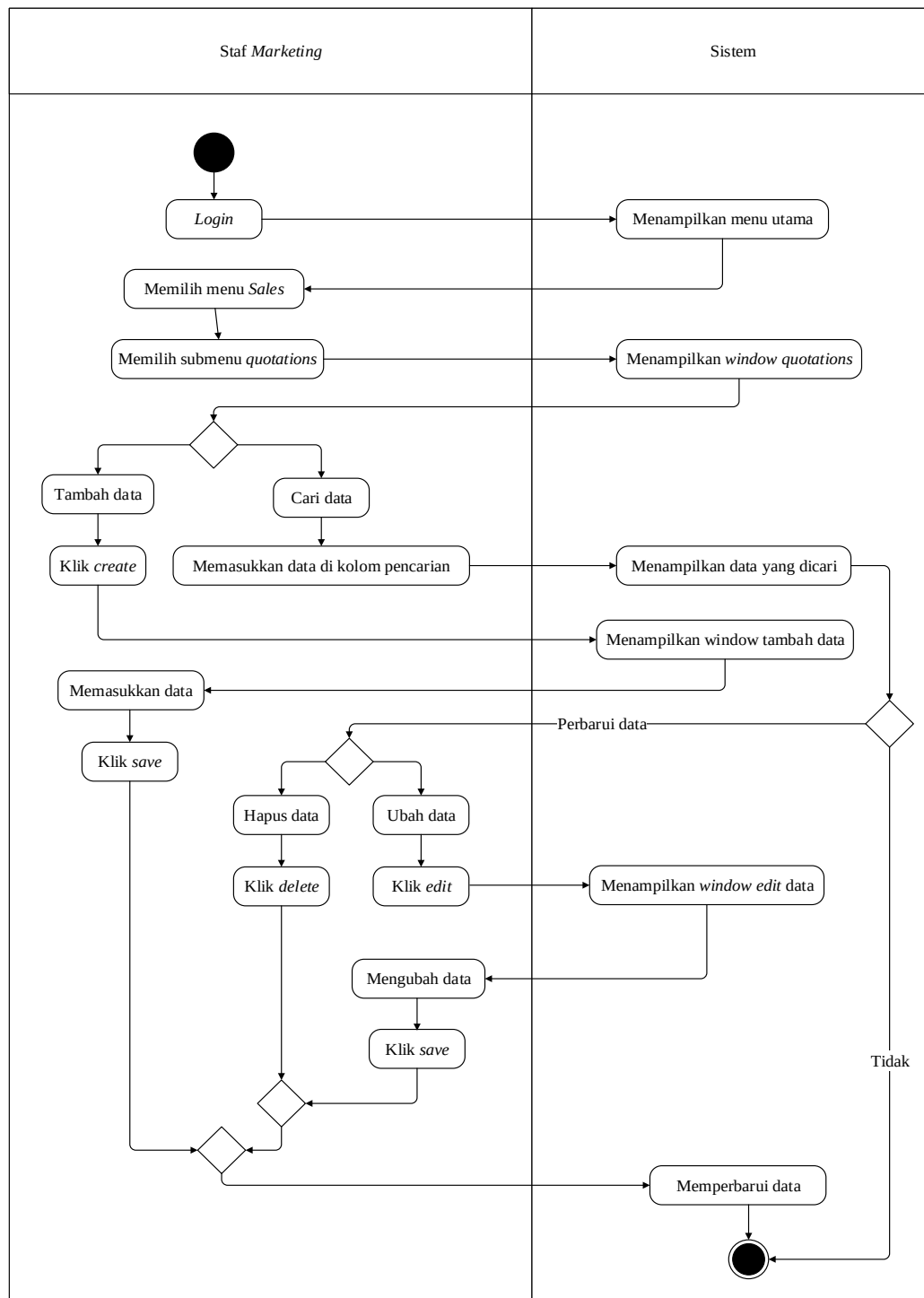
Gambar V.7 Activity Diagram Mengelola Data Customer
(Sumber: Hasil Analisis, 2018)

6. Activity Diagram Mengelola Data Produk



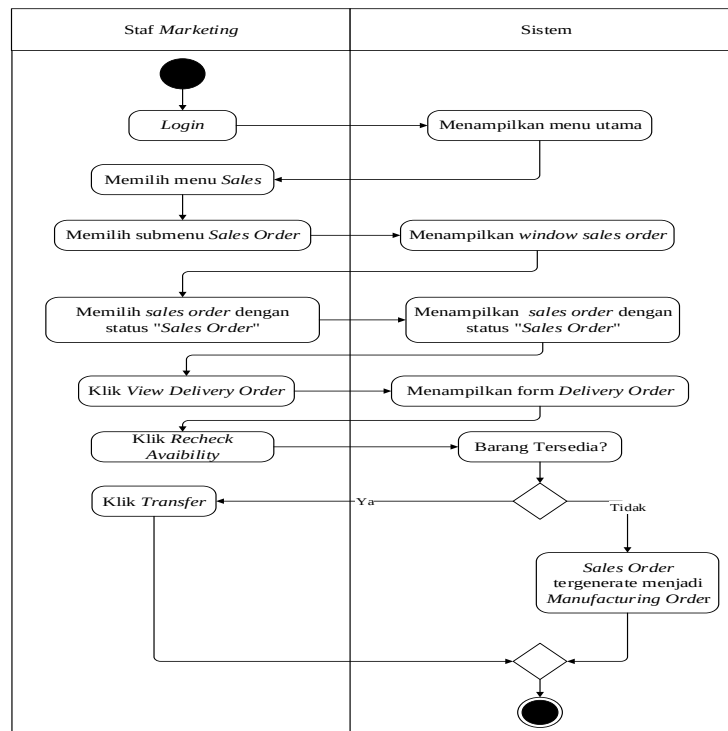
Gambar V.8 Activity Diagram Mengelola Data Produk
(Sumber: Hasil Analisis, 2018)

7. Activity Diagram Mengelola data Quotation atau surat penawaran harga



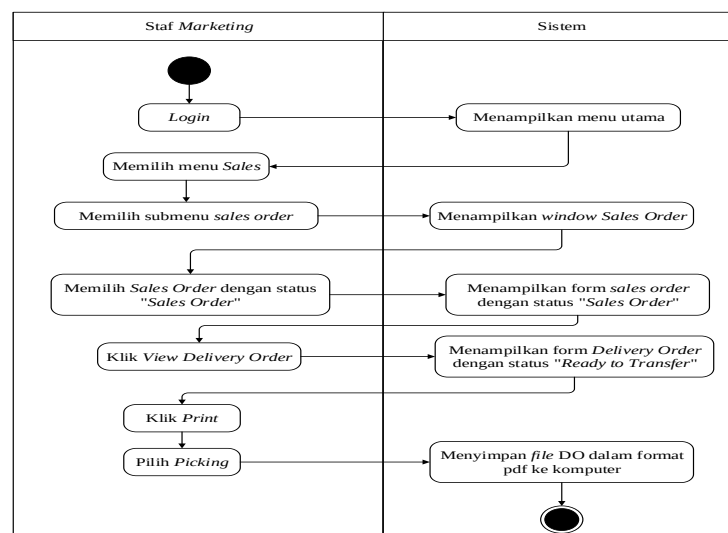
Gambar V.9 Activity Diagram Mengelola data Quotation
(Sumber: Hasil Analisis, 2018)

8. Activity Diagram Mengelola Delivery Order



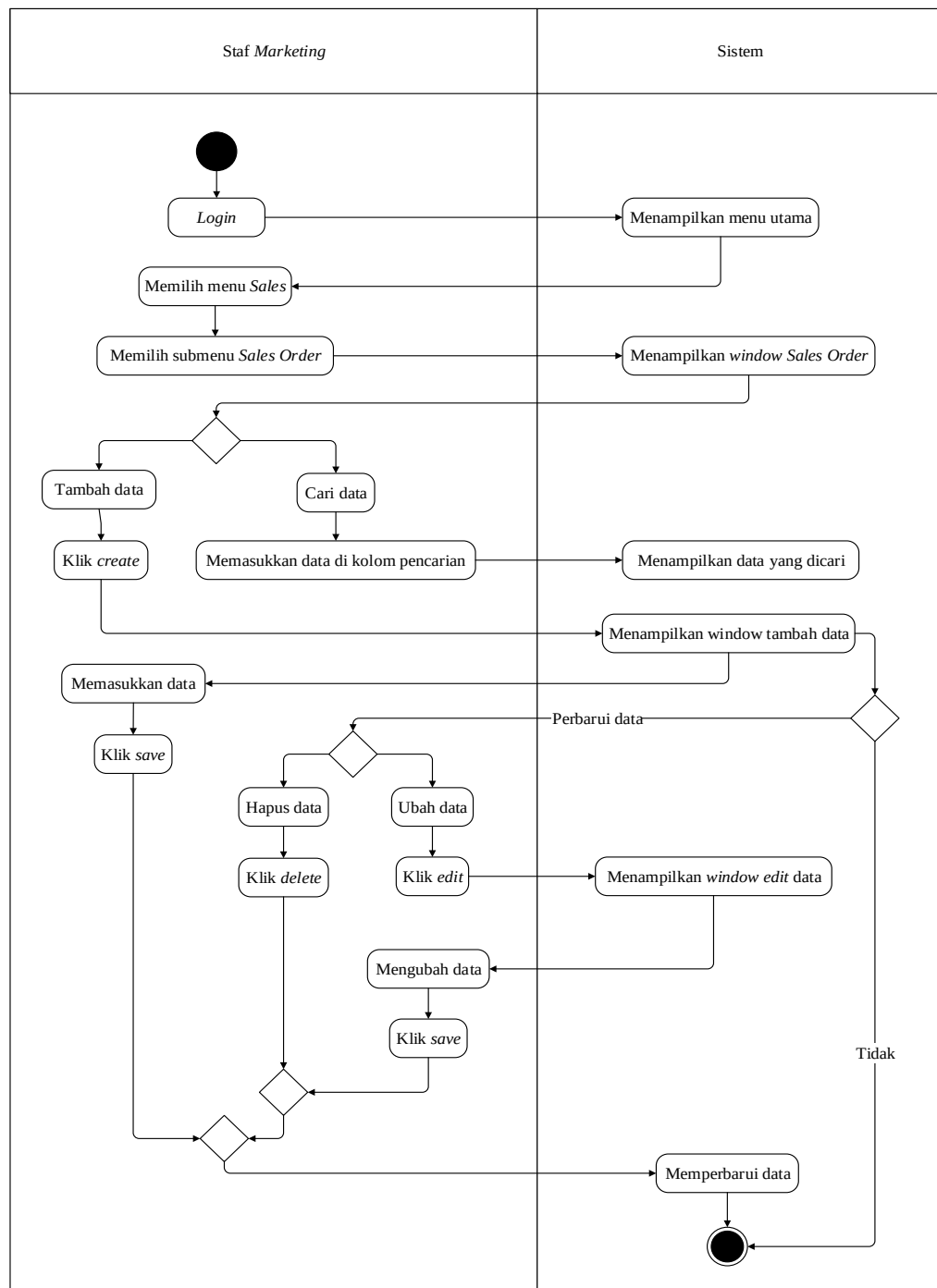
Gambar V.10 Activity Diagram Mengelola Delivery Order
(Sumber: Hasil Analisis, 2018)

9. Activity Diagram Mencetak Delivery Order



Gambar V.11 Activity Diagram Mencetak Delivery Order
(Sumber: Hasil Analisis, 2018)

10. Activity Diagram Mengelola Sales Order

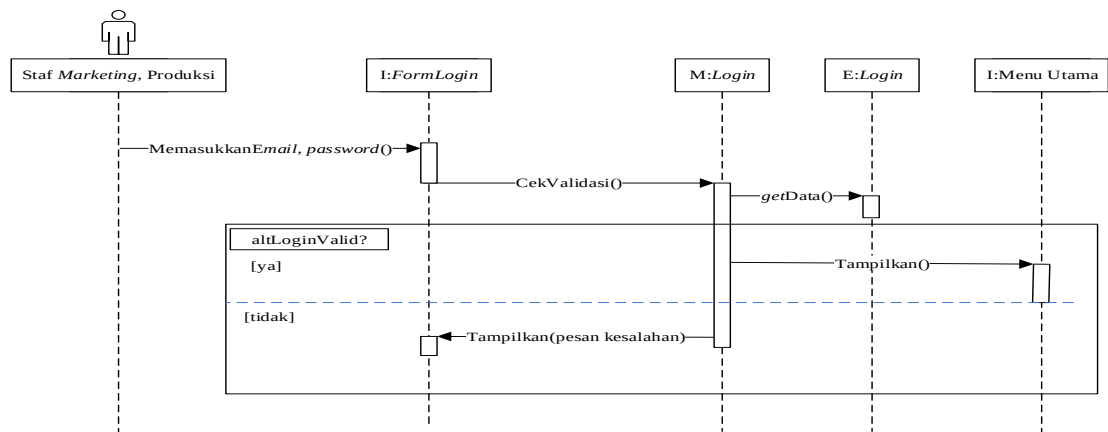


Gambar V.12 Activity Diagram Mengelola Sales Order
(Sumber: Hasil Analisis, 2018)

5.5 Sequence Diagram

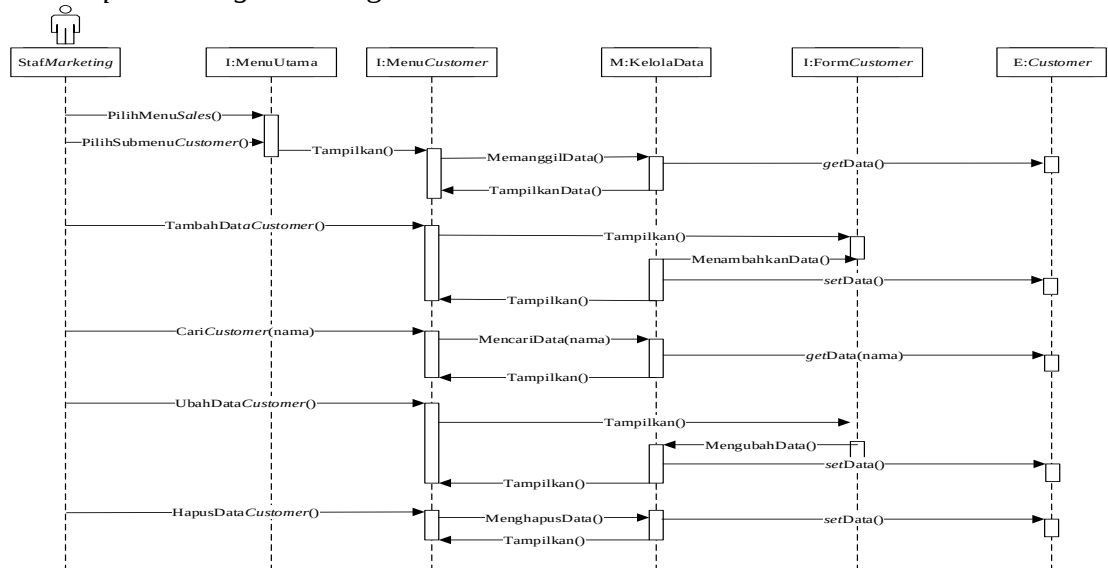
Sequence Diagram menggambarkan interaksi yang terjadi antar objek pada suatu *Use Case Diagram*. Berikut ini adalah *sequence diagram* sistem ERP penjualan produk pada PT DRA Component Persada:

1. Sequence diagram login



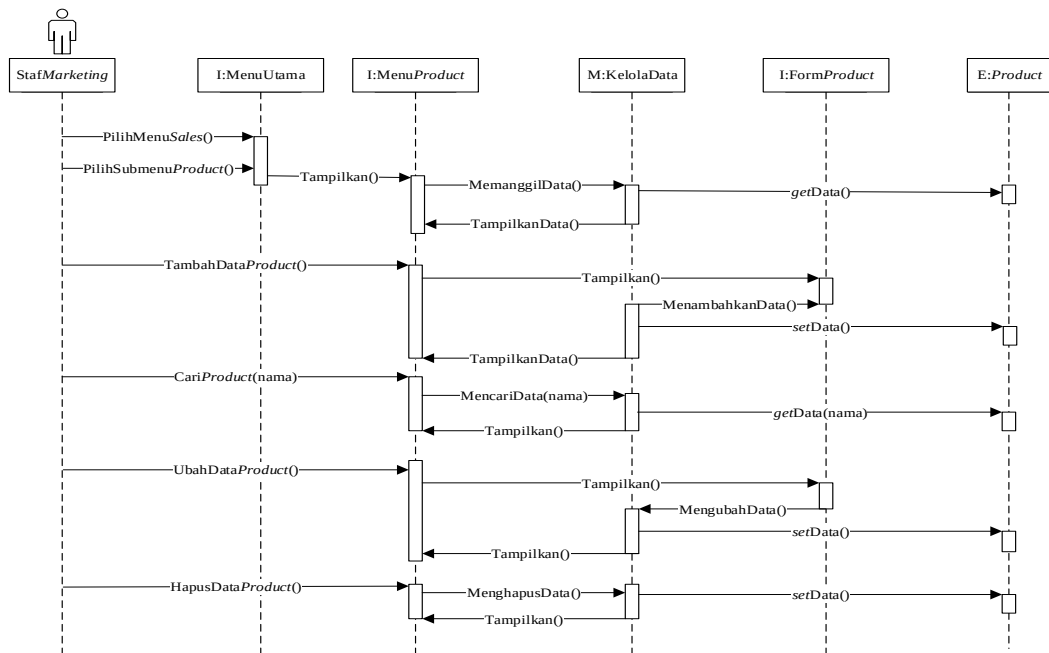
Gambar V.13 Sequence Diagram Login
(Sumber: Hasil Analisis, 2018)

2. Sequence diagram mengelola data customer



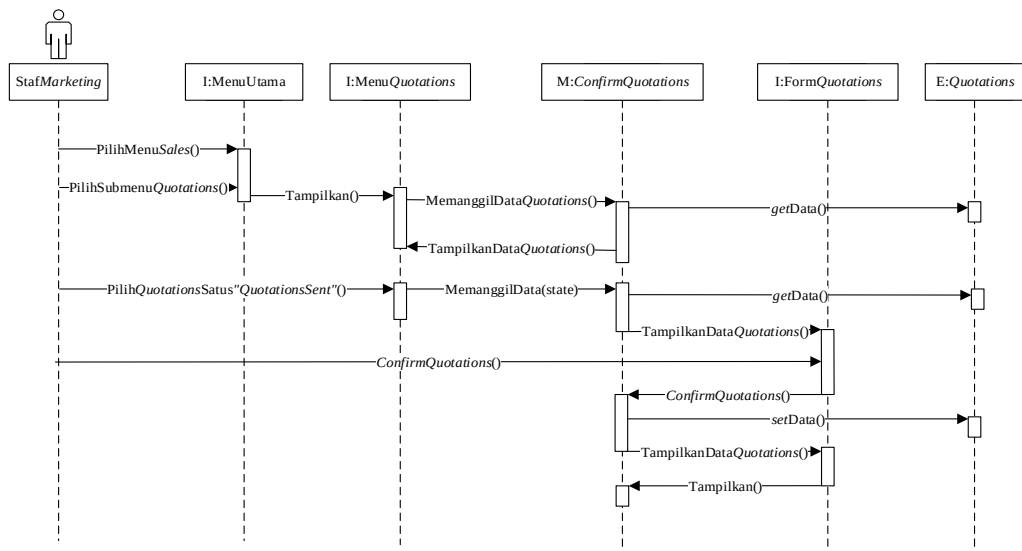
Gambar V.14 Sequence Diagram Mengelola Data customer
(Sumber: Hasil Analisis, 2018)

3. Sequence diagram mengelola data produk



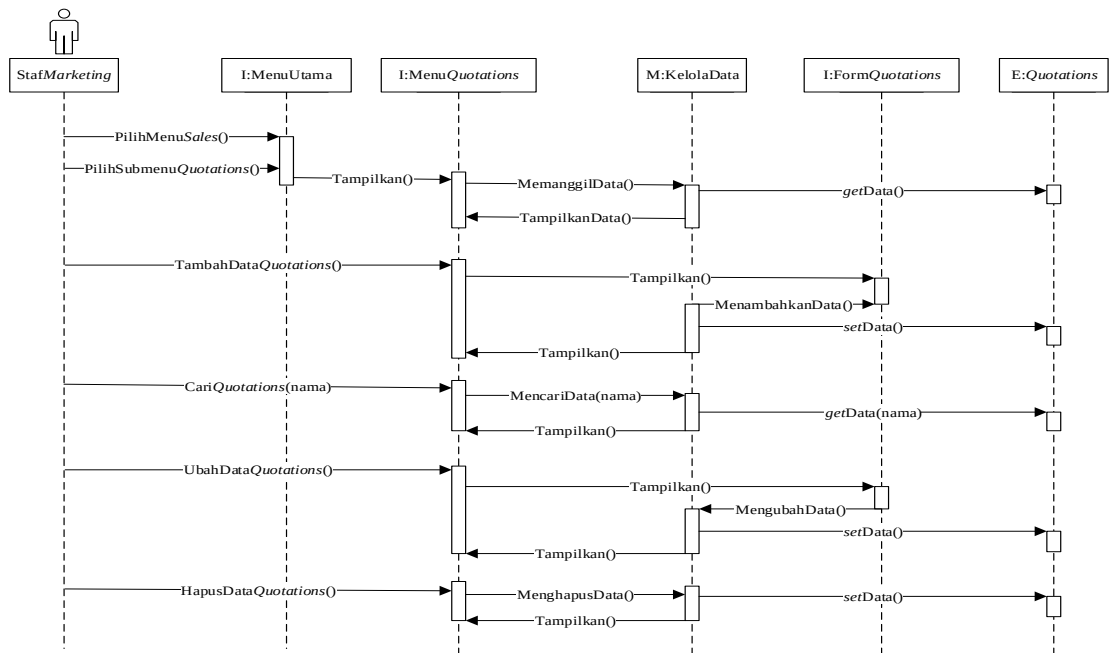
Gambar V.15 *Sequence Diagram* Mengelola Data produk
(Sumber: Hasil Analisis, 2018)

4. *Sequence diagram* Konfirmasi Quotation



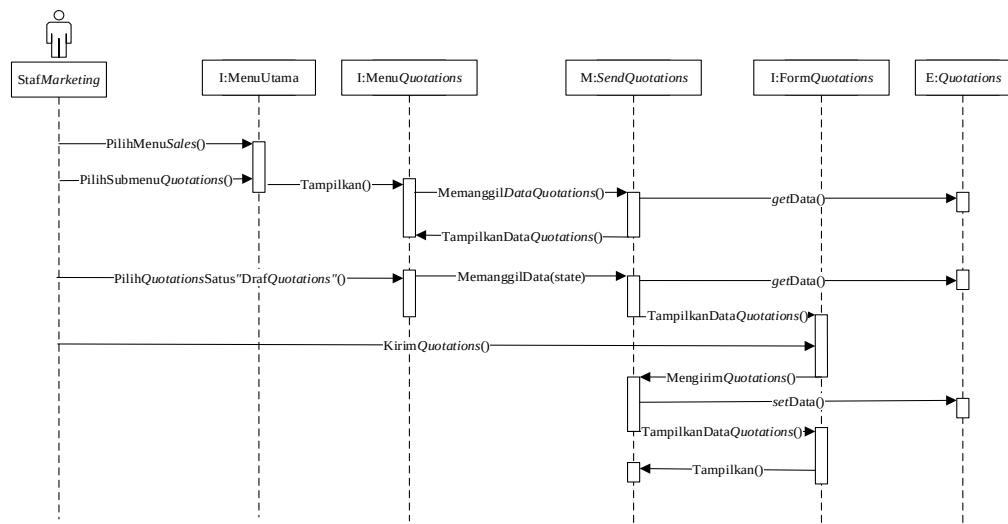
Gambar V.16 *Sequence Diagram* Konfirmasi Quotations
(Sumber: Hasil Analisis, 2018)

5. *Sequence diagram* Mengelola data Quotations atau surat penawaran harga



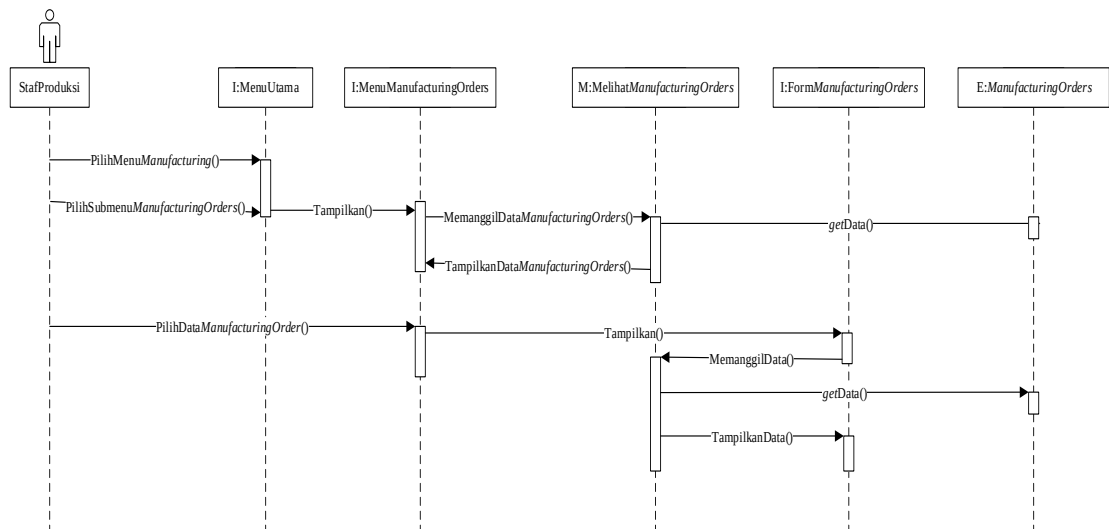
Gambar V.17 Sequence Diagram Mengelola data Quotations
(Sumber: Hasil Analisis, 2018)

6. Sequence diagram Mengirimkan Quotations



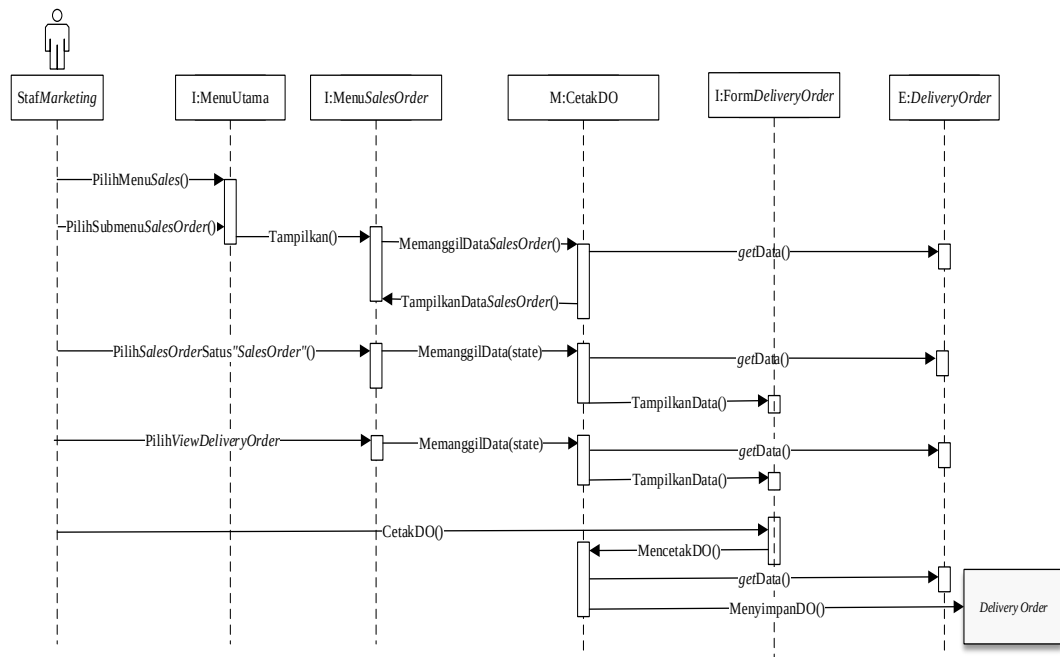
Gambar V.18 Sequence Diagram Mengirimkan Quotations
(Sumber: Hasil Analisis, 2018)

7. Sequence diagram Melihat Manufacturing Order



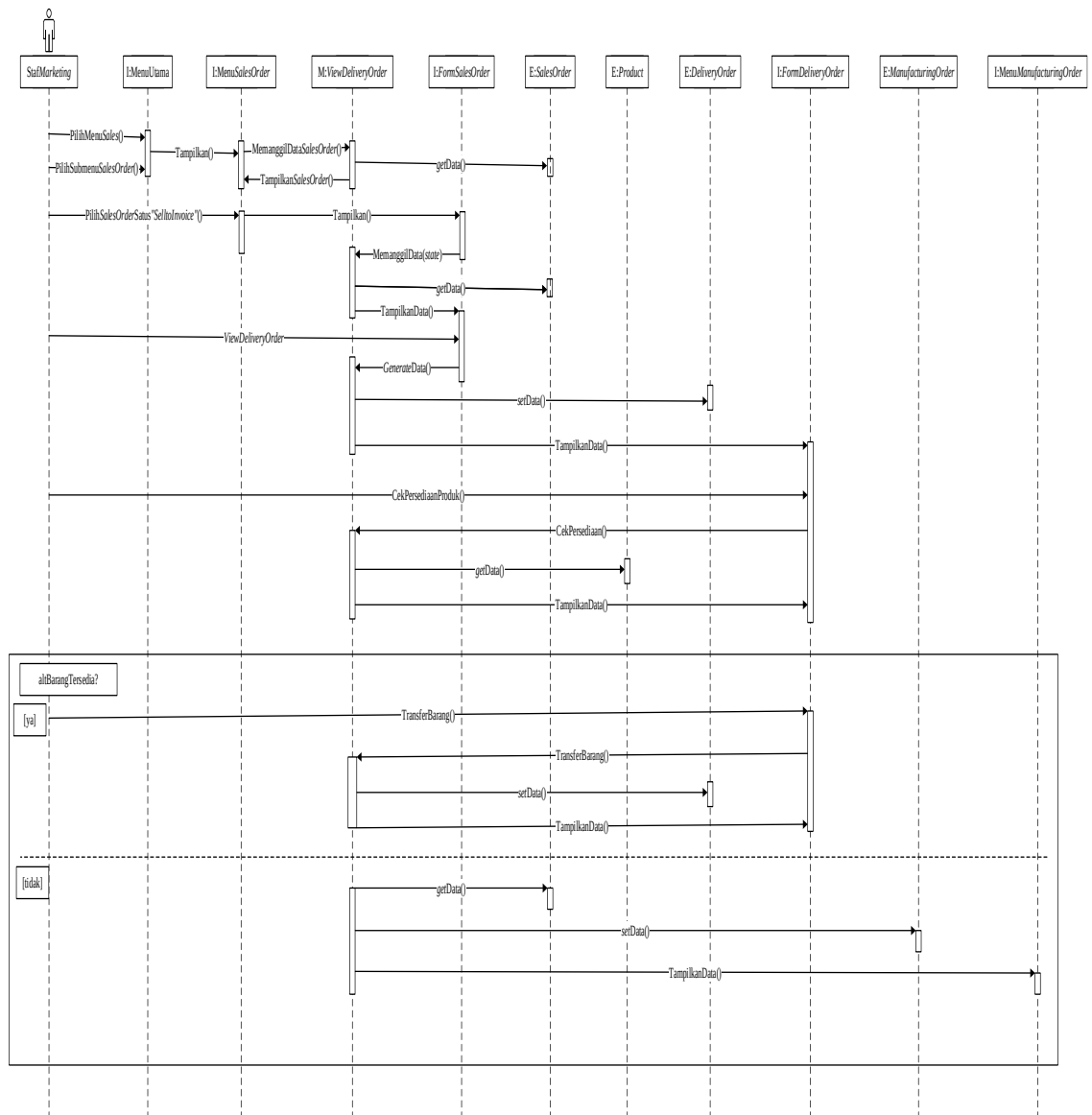
Gambar V.19 Sequence Diagram Melihat Manufacturing Order
(Sumber: Hasil Analisis, 2018)

8. Sequence diagram Mencetak Delivery Order



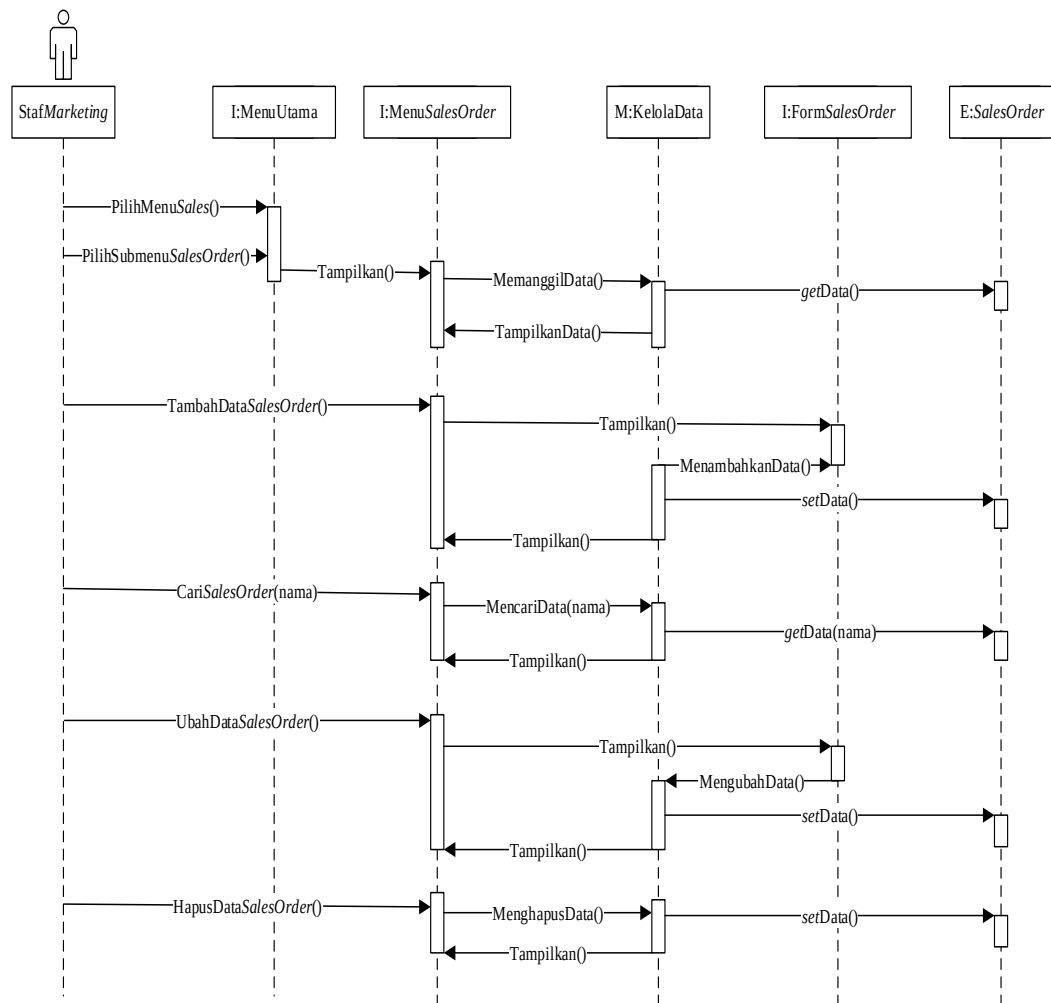
Gambar V.20 Sequence Diagram Mencetak Delivery Order
(Sumber: Hasil Analisis, 2018)

9. Sequence diagram Mengelola Delivery Order



Gambar V.21 Sequence Diagram Mengelola Delivery Order
(Sumber: Hasil Analisis, 2018)

10. Sequence diagram Mengelola Sales Order

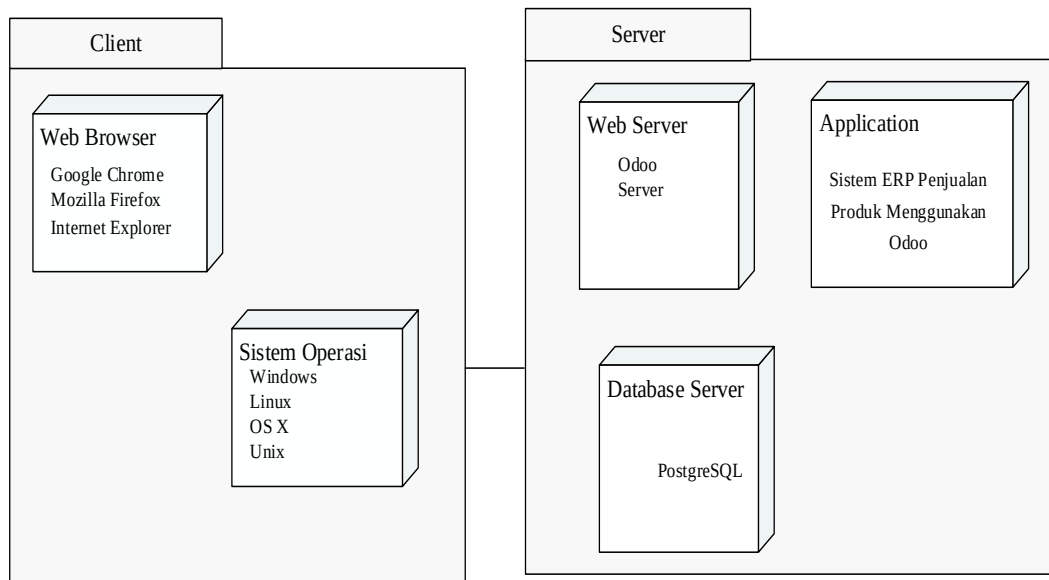


Gambar V.22 Sequence Diagram Mengelola Sales Order
(Sumber: Hasil Analisis, 2018)

5.6 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem. Berikut ini merupakan *class diagram* sistem ERP Penjualan Produk pada PT DRA Component Persada:

merupakan *deployment diagram* sistem ERP Penjualan Produk Usulan pada PT DRA Component Persada yang diusulkan:

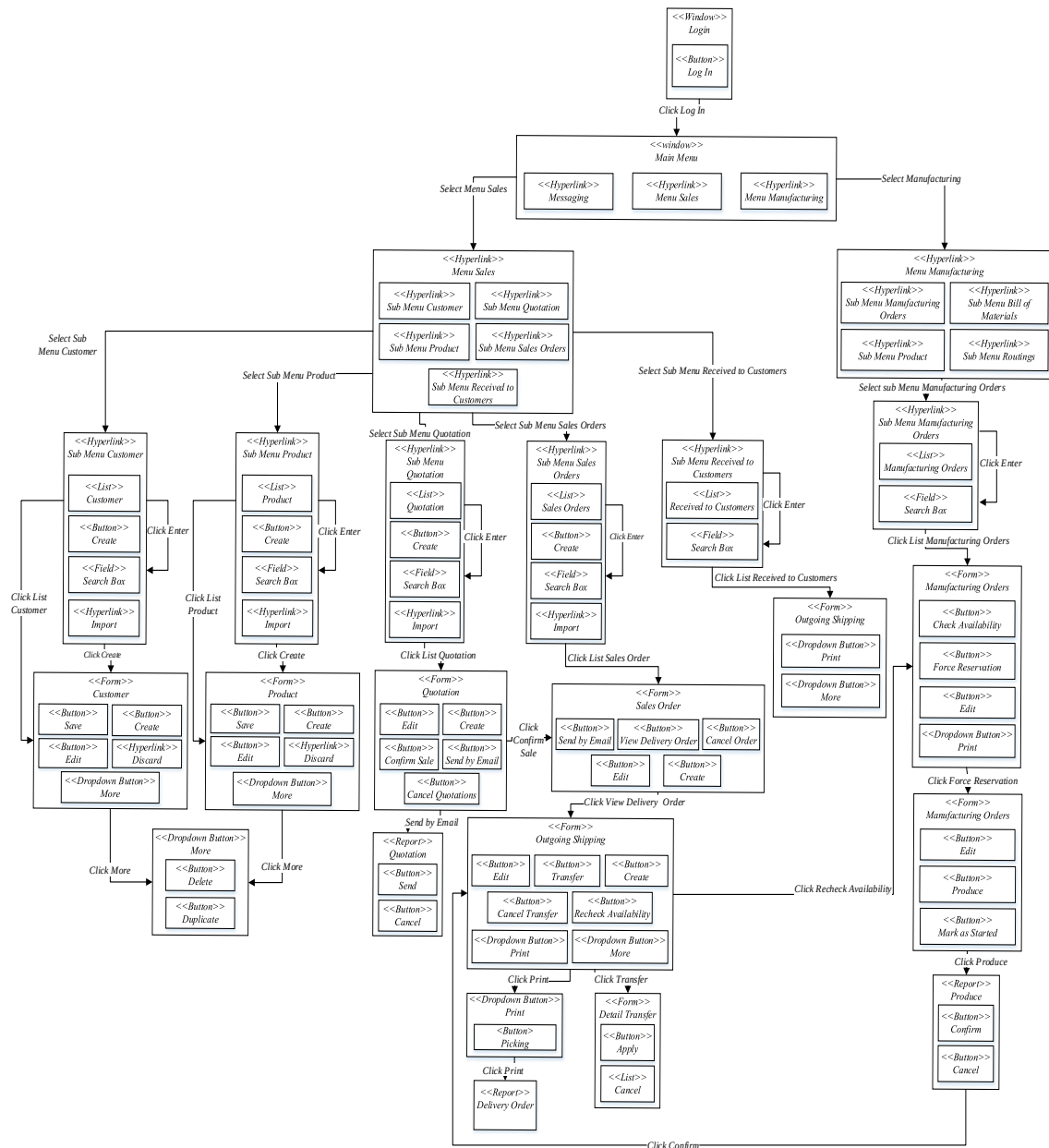


Gambar V.24 *Deployment Diagram* Sistem Penjualan Produk Usulan

(Sumber: Hasil Analisis, 2018)

5.8 *Window Navigation Diagram*

Windows Navigation Diagram digunakan untuk menggambarkan desain struktur navigasi dari suatu sistem. Berikut ini merupakan *window navigation diagram* sistem ERP Penjualan Produk usulan pada PT DRA Component Persada yang diusulkan:



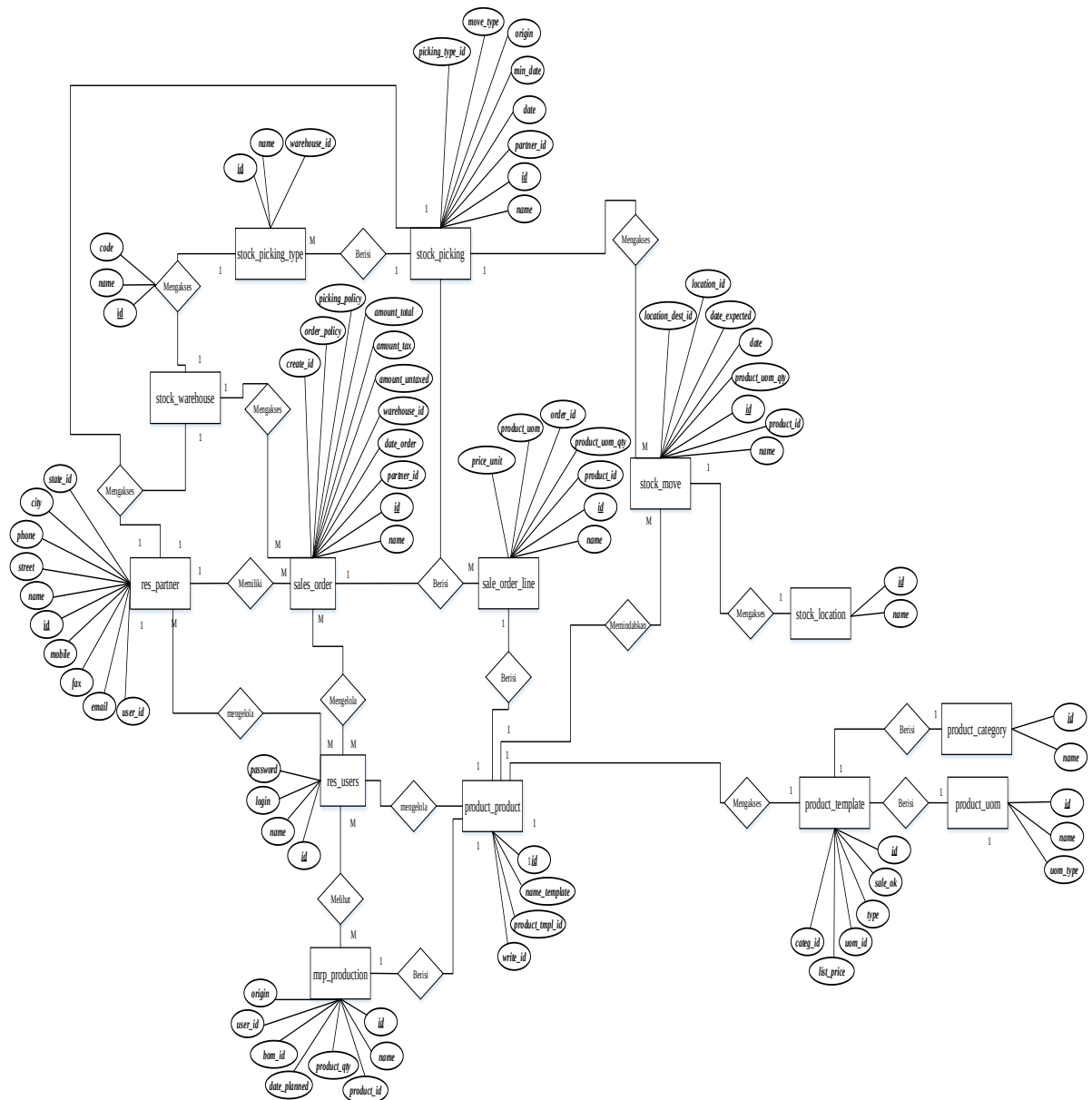
Gambar V.25 Windows Navigation Sistem Penjualan Produk Usulan

(Sumber: Hasil Analisis, 2018)

5.9 Pemodelan Data Sistem Usulan

Pemodelan data sistem usulan akan dimodelkan menggunakan *entity relationship diagram*.

Entity Relationship Diagram digunakan untuk memodelkan kebutuhan dari organisasi yang direpresentasikan untuk mendesain *database* sistem terkait. Berikut adalah *entity relationship diagram* sistem ERP penjualan produk susulan pada PT DRA Component Persada:



Gambar V.26 Entity Relationship Diagram Sistem Penjualan Produk

Sumber: Hasil Analisis (2018)

5.10 Black Box Testing

Pengujian sistem ERP penjualan produk usulan dilakukan menggunakan metode black box testing. Pengujian dilakukan untuk memastikan bahwa sistem berjalan sesuai dengan yang telah dirancang. Adapun black box testing dari sistem ERP pembelian bahan baku usulan sebagai berikut (terlampir).

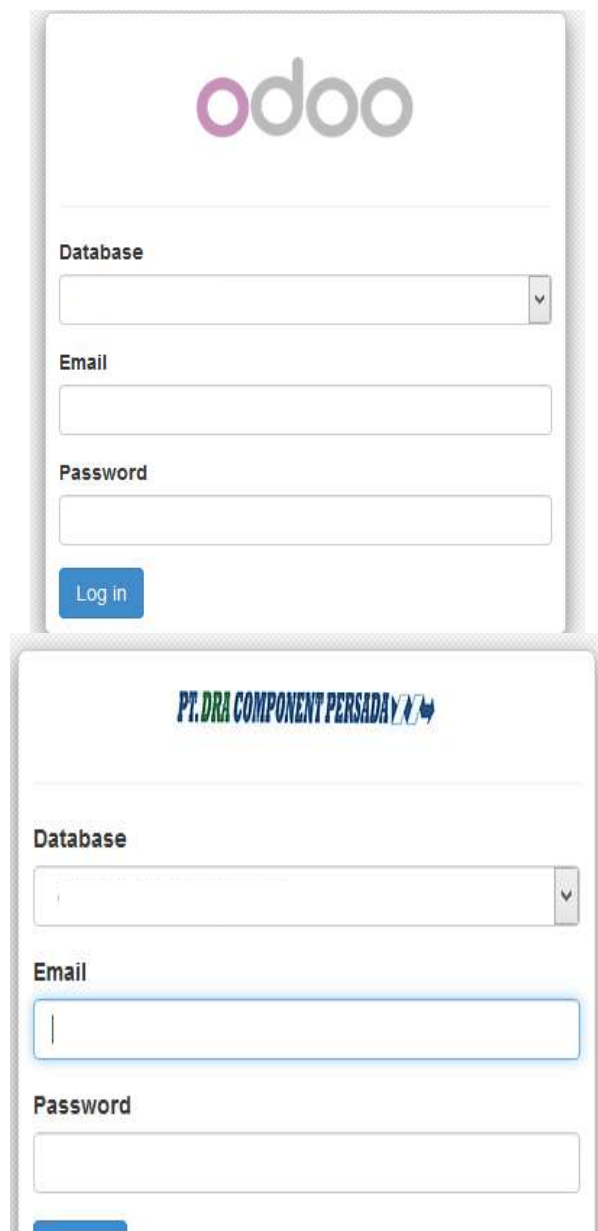
5.11 Perancangan Sistem Usulan

Pada perancangan sistem usulan akan dijelaskan mengenai perancangan *interface* sistem usulan serta perancangan transaksi sistem usulan dari sistem penjualan produk pada PT DRA Component Persada

5.11.1 Perancangan *Interface* Sistem Usulan

Pada perancangan sistem penjualan produk sebagai berikut:

1. Form *Login*



The image displays two screenshots of login interfaces. The top screenshot shows the Odoo login form, which includes a header with the Odoo logo, a 'Database' dropdown menu, 'Email' and 'Password' text input fields, and a blue 'Log in' button. The bottom screenshot shows the login form for PT. DRA COMPONENT PERSADA, featuring a header with the company logo, a 'Database' dropdown menu, 'Email' and 'Password' text input fields, and a partially visible blue button at the bottom.

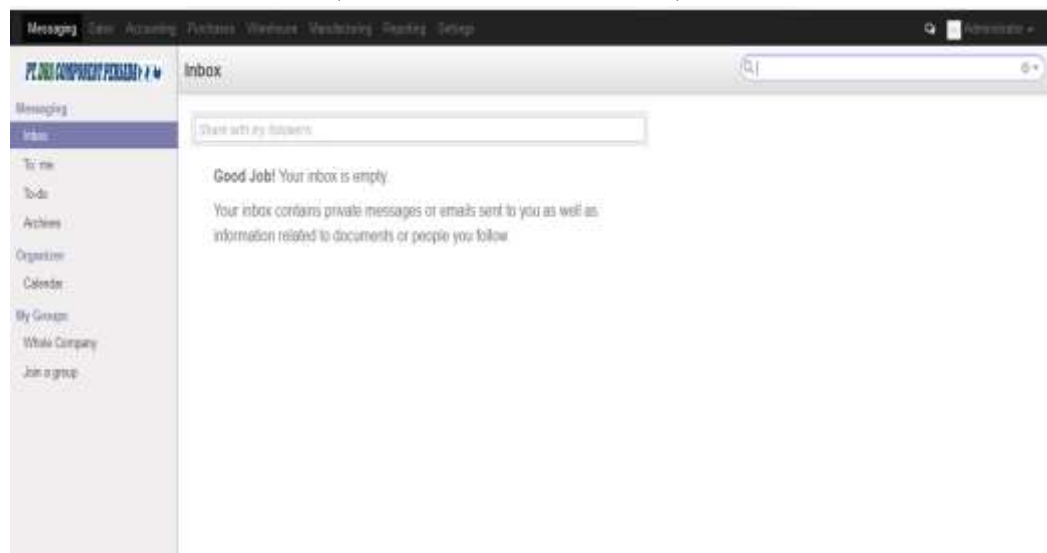
Gambar V.28 Perancangan *Login*
(Sumber: Hasil Analisis, 2018)

2. Menu Utama

Menu Utama pada *default* Odoo, modul-modul belum ada yang ter-*install*



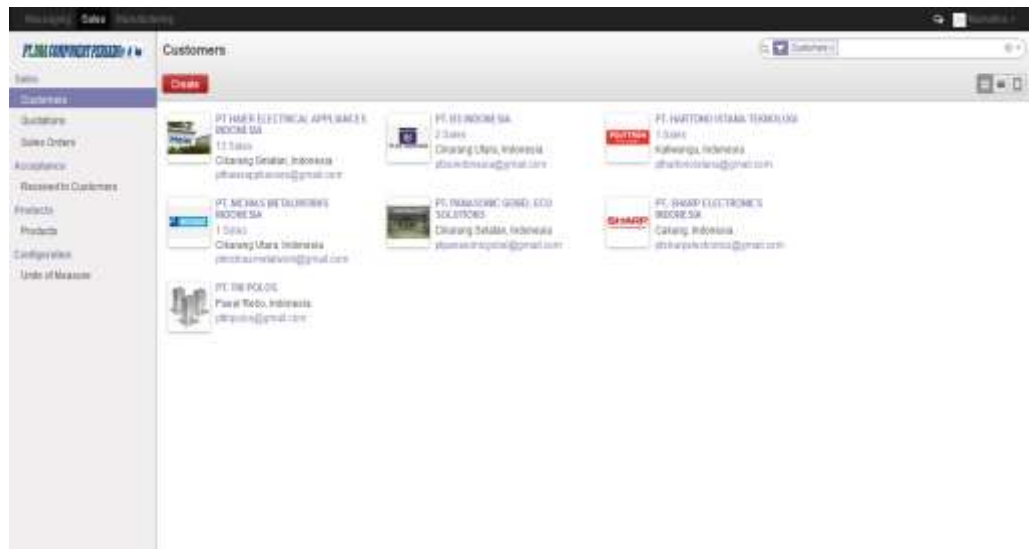
Gambar V.29 *Default* Menu Utama
(Sumber: Localhost/Odoo, 2018)



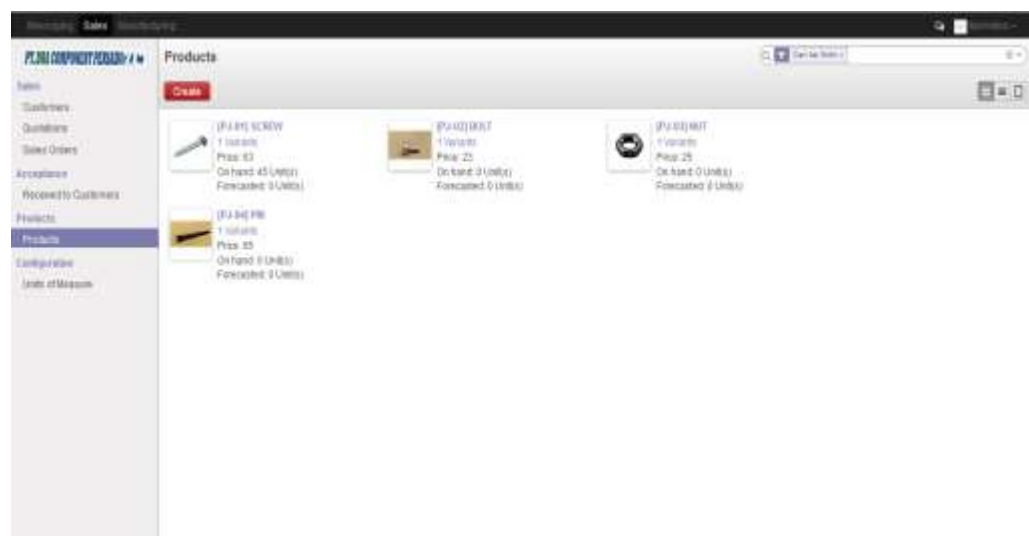
Gambar V.30 Perancangan Menu Utama
(Sumber: Hasil Analisis, 2018)

Modul yang dibutuhkan pada sistem penjualan produk usulan ini adalah:

1. Modul *Sales Management*. Setelah modul ini ter-*install* akan menghasilkan menu *Sales*. Adapun submenu yang terdapat pada menu *Sales* antara lain:
 - a. Submenu *Customer*



Gambar V.31 Submenu *Customer*
(Sumber: Hasil Analisis, 2018)



- b. Submenu *Product*

Gambar V.32 Submenu *Product*
(Sumber: Hasil Analisis, 2018)

c. Submenu *Quotation*

Quotation Number	Date	Customer	Salesperson	Total	Status
80012	8/8/2018 08:19:50	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	821.80	Quotation Sent

Gambar V.33 Submenu *Quotations*
(Sumber: Hasil Analisis, 2018)

d. Submenu *Sales Order*

Order Number	Date	Customer	Salesperson	Total	Status
SO004	8/8/2018 17:42:08	PT. BS INDONESIA	Marketing	18,360.00	Cancel
SO001	8/8/2018 03:30:27	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	7,523.00	Sales Order
SO003	8/8/2018 02:52:25	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	2,172.00	Sales Order
SO019	8/8/2018 21:26:28	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	3,465.00	Cancel
SO018	8/8/2018 19:20:14	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	8,938.00	Cancel
SO017	8/8/2018 08:12:39	PT. BS INDONESIA	Marketing	1,381.00	Cancel
SO015	8/8/2018 06:57:23	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	1,038.90	Sales Order
SO014	8/8/2018 06:52:15	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	901.90	Sales Order
SO013	8/8/2018 08:37:36	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	1,581.90	Sales Order

Gambar V.34 Submenu *Sales Order*
(Sumber: Hasil Analisis, 2018)

2. Modul *Manufacturing Management*. Setelah modul ini ter-install akan menghasilkan menu *Manufacturing*. Pada menu ini hanya menggunakan sub menu *Manufacturing Orders* untuk melihat *Manufacturing Order* yang ter-generate dari sales order.

Sub menu *Manufacturing Order*

Reference	Scheduled Date	Product	Product Quantity	Product Unit of Measure	Routing	Total Hours	Total Cycles	Source Document	Status
MO00010	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000010	Canc
MO00011	2013-02-10 11:21:55	[WFF-4%]Routing SCREW	1	Units	heading-01	0.00	1.00	SO0001000011	Canc
MO00012	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000012	Canc
MO00013	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000013	Canc
MO00014	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000014	Canc
MO00015	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000015	Canc
MO00016	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000016	Canc
MO00017	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000017	Canc
MO00018	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000018	Canc
MO00019	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000019	Canc
MO00020	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000020	Canc
MO00021	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000021	Canc
MO00022	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000022	Canc
MO00023	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000023	Canc
MO00024	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000024	Canc
MO00025	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000025	Canc
MO00026	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000026	Canc
MO00027	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000027	Canc
MO00028	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000028	Canc
MO00029	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000029	Canc
MO00030	2013-02-10 11:21:55	[WFF-4%]Headings SCREW	1	Units	heading-01	0.00	1.00	SO0001000030	Canc

Gambar V.35 Submenu *Manufacturing Order*
(Sumber: Hasil Analisis, 2018)

5.11.2 Perancangan Transaksi Sistem Usulan

Sementara untuk perancangan alur desain sistem transaksinya adalah sebagai berikut:

1. Membuat data master *customer*

Gambar V.36 Default form *customer*
(Sumber Localhost/Odoo, 2018)

Gambar V.37 Perancangan form *customer*
(Sumber: Hasil Analisis, 2018)

Pada *default* form *customer* masih muncul *field tags*, *website*, *title* dan *contact reference* sedangkan pada perancangan form *customer* sudah tidak ditampilkan. Hal ini dikarenakan PT DRA Component Persada tidak memiliki data mengenai hal tersebut dan pada perancangan form *customer* pilihan *supplier* sudah tidak ditampilkan. Hal ini dikarenakan form tersebut merupakan form *customer* sehingga tidak perlu ada pilihan *supplier*

2. Membuat data master *product*

Gambar V.38 Default form *product*
(Sumber: Localhost/Odoo, 2018)

Gambar V.39 Perancangan form *product*
(Sumber: Hasil Analisis, 2018)

Pada *default form product* masih muncul *field EAN13 Barcode*, *Internal Reference* dan grup *Variants* sedangkan pada perancangan form *product* sudah tidak ditampilkan. Hal ini dikarenakan PT DRA Component Persada tidak memiliki data mengenai hal tersebut.

3. Membuat form *Quotations*

Pada proses ini, bagian *marketing* dapat membuat form *quotations* berupa penawaran harga atas produk-produk yang akan dijual kepada *customer*. Form ini dapat digunakan untuk proses penawaran harga yang sebelumnya dilakukan hanya via telepon.

The screenshot displays the Odoo Quotation form for SO024. The interface includes a sidebar with navigation options like Sales, Customers, Quotations, and Orders. The main form area shows the following details:

- Customer:** PT PRISCA/PTON, Jl. Sekeloa Timur PT No.39, Kawasan Industri Jababara, Pajajaran, Cikarang Utara 505, Indonesia.
- Date:** 08/02/2018 17:42:00
- Reference/Exception:** Warehouse
- Product Code:**

The **Order Lines** table contains the following data:

Product	Description	Quantity	Unit of Measure	Unit Price	Tax%	Subtotal
PJ-A6 SCREW	PJ-A11 SCREW	100	Units	83	PPh Kalkulasi 12%	8.300,00
PJ-A5 BOLT	PJ-A21 BOLT	100	Units	23	PPh Kalkulasi 12%	2.300,00
PJ-A3 NUT	PJ-A21 NUT	100	Units	23	PPh Kalkulasi 12%	2.300,00
PJ-A4 PIN	PJ-A41 PIN	100	Units	83	PPh Kalkulasi 12%	8.300,00

Totals:

- Untaxed Amount: Rp 17.800,00
- Taxes: Rp 1.750,00
- Total: Rp 19.550,00**

Gambar V.40 Default form *quotation*
(Sumber: Localhost/Odoo, 2018)

Quotation SO024

Customer: PT DRA COMPONENT PERSADA
Jl. Jendral Sudirman No. 17 No. 25
Kawasan Industri Jakarta, Jakarta
Cikarang Utara 555
Indonesia

Sale: 06072018 17:42:08
Warehouse: Pabrik Candi

Product	Description	Quantity	Unit of Measure	Unit Price	Taxes	Subtotal
PJ-01 SCREW	PJ-01 SCREW	100	Unit(s)	60	PPN 11%	6.300.00
PJ-02 BOLT	PJ-02 BOLT	100	Unit(s)	23	PPN 11%	2.300.00
PJ-03 NUT	PJ-03 NUT	100	Unit(s)	25	PPN 11%	2.500.00
PJ-04 PIN	PJ-04 PIN	100	Unit(s)	55	PPN 11%	5.500.00

Unterschied Amount: Rp 17.000.00
Taxes: Rp 1.700.00
Total: Rp 19.360.00

Gambar V.41 Perancangan form *quotation*
(Sumber: Hasil Analisis, 2018)

Pada *default form quotations* masih muncul *Reference/Description* sedangkan pada perancangan form *quotations* sudah tidak ditampilkan. Hal ini dikarenakan PT DRA Component Persada tidak memiliki data mengenai hal tersebut dan tombol *print* juga tidak ditampilkan karena pada *button more* sudah ada tombol *print quotations*.

4. *Quotations* akan dikirimkan ke *customer*

Quotation SO024

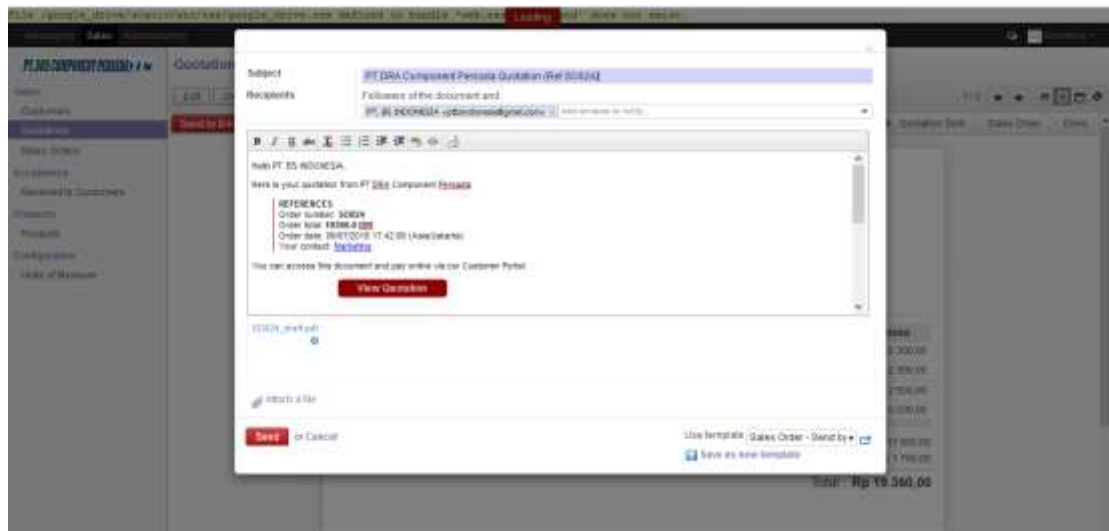
Customer: PT DRA COMPONENT PERSADA
Jl. Jendral Sudirman No. 17 No. 25
Kawasan Industri Jakarta, Jakarta
Cikarang Utara 555
Indonesia

Sale: 06072018 17:42:08
Warehouse: Pabrik Candi

Product	Description	Quantity	Unit of Measure	Unit Price	Taxes	Subtotal
PJ-01 SCREW	PJ-01 SCREW	100	Unit(s)	60	PPN 11%	6.300.00
PJ-02 BOLT	PJ-02 BOLT	100	Unit(s)	23	PPN 11%	2.300.00
PJ-03 NUT	PJ-03 NUT	100	Unit(s)	25	PPN 11%	2.500.00
PJ-04 PIN	PJ-04 PIN	100	Unit(s)	55	PPN 11%	5.500.00

Unterschied Amount: Rp 17.000.00
Taxes: Rp 1.700.00
Total: Rp 19.360.00

Gambar V.42 Form *Send Quotations by Email*
(Sumber: Hasil Analisis, 2018)



Gambar V.43 Send Quotations by Email

(Sumber: Hasil Analisis, 2018)

Form quotations dapat dikirimkan ke *customer* melalui *e-mail* maupun datang secara langsung kepada *customer* dengan mencetak form quotations ini.

5. Konfirmasi Penjualan

Customer akan menerima quotations, jika *customer* menyetujui untuk melakukan pembelian maka *customer* akan mengirimkan purchase order berupa list produk-produk yang akan dibeli. Setelah menerima purchase order dari *customer*, bagian marketing dapat melakukan pencarian form quotations pada database ini yang sesuai dengan *customer* terkait, dimana form quotations tersebut sudah dibuat sebelumnya. Setelah form quotations yang terkait sudah ditemukan, bagian marketing dapat melakukan pengeditan data quotations dengan memilih

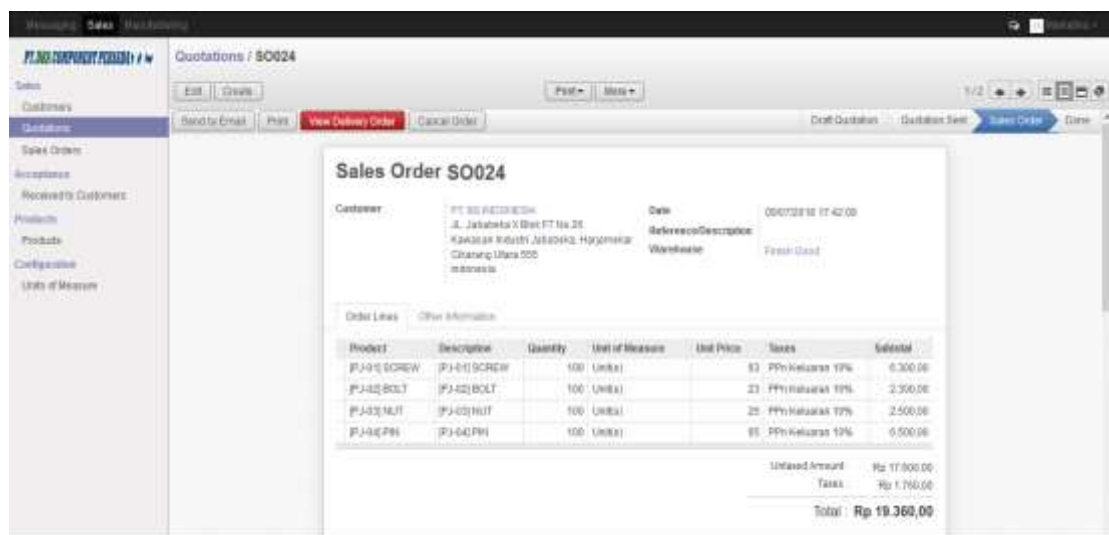
Product	Description	Quantity	Unit of Measure	Unit Price	Notes	Subtotal
PU-01 SCREW	PU-01 SCREW	100	UNIT	43	PPH 10%	4.300,00
PU-02 BOLT	PU-02 BOLT	100	UNIT	21	PPH 10%	2.100,00
PU-03 NUT	PU-03 NUT	100	UNIT	20	PPH 10%	2.000,00
PU-04 PIN	PU-04 PIN	100	UNIT	90	PPH 10%	9.000,00
Unlabeled Amount						Rp 17.800,00
Taxes						Rp 1.700,00
Total						Rp 19.360,00

tombol *edit* untuk mengubah produk agar sesuai dengan pesanan *customer*. Apabila data *quotations* telah sesuai dengan pesanan *customer*, bagian *marketing* dapat menekan tombol *confirm sale* yang menyatakan bahwa penjualan tersebut telah disetujui dan disepakati

Gambar V.44 Form *confirm sale*
(Sumber: Hasil Analisis, 2018)

6. Sales Order

Setelah bagian *marketing* menekan *confirm sale*, sistem akan secara otomatis men-generate form *sales order* yang baru dengan nomer dokumen yang sama



Sales Order SO024

Customer: PT. ABC COMPANY PT. XYZ
Jl. Jakarta X Blok IT No. 25
Kawasan Industri Jakarta Utara, Indonesia
Cibitung Utara 500
Indonesia

Date: 05/07/2018 10:17:42:00
Reference/Description: Warehouse
Product: Fresh Food

Product	Description	Quantity	Unit of Measure	Unit Price	Taxes	Subtotal
PJ-01 SCREW	PJ-01 SCREW	100	Unit(s)	53	PPN Kenaikan 10%	5.300.00
PJ-02 BOLT	PJ-02 BOLT	100	Unit(s)	23	PPN Kenaikan 10%	2.300.00
PJ-03 NUT	PJ-03 NUT	100	Unit(s)	25	PPN Kenaikan 10%	2.500.00
PJ-04 PIN	PJ-04 PIN	100	Unit(s)	55	PPN Kenaikan 10%	5.500.00
Unladed Amount						Rp 17.600.00
Taxes						Rp 1.760.00
Total						Rp 19.360.00

dengan *quotations* yang terkait. Namun, bagian *marketing* juga dapat membuat form *sales order* secara langsung tanpa mengacu pada *quotations*.

Gambar V.45 Default form *sales order*

(Sumber: Localhost/Odoo, 2018)

Sales Order SO024

Customer: PT BIRACORONA
Jl. Jendral A. Yani 17 No. 26
Kawasan Industri Jember, Jember
Jember 60132

Date: 08/02/19 17:40:00
Warehouse: Fresh Good

Product	Description	Quantity	Unit of Measure	Unit Price	Taxes	Subtotal
PJ-01 SCREW	PJ-01 SCREW	100	Unit	63	PPH Kalkulasi 10%	6.300,00
PJ-02 BOLT	PJ-02 BOLT	100	Unit	23	PPH Kalkulasi 10%	2.300,00
PJ-03 NUT	PJ-03 NUT	100	Unit	25	PPH Kalkulasi 10%	2.500,00
PJ-04 PIN	PJ-04 PIN	100	Unit	60	PPH Kalkulasi 10%	6.000,00

Unlabeled Amount: Rp 17.900,00
Taxes: Rp 1.790,00
Total: Rp 19.360,00

Gambar V.46 Perancangan form *sales order*
(Sumber: Hasil Analisis, 2018)

Pada perancangan form *sales order* tombol *print* tidak ditampilkan karena pada button more sudah ada tombol *print*. Bagian *marketing* dapat mencetak *sales order* sebagai arsip yang akan disimpan maupun tidak.

7. Form *Outgoing Shipping*

Form *Outgoing Shipping* ini ter-generate langsung dari form *sales order*. Sebelum melakukan pengiriman, bagian *marketing* terlebih dahulu melihat data *sales order* pada sistem untuk menentukan *sales order* yang mana yang akan dilakukan pengiriman produk. Setelah itu bagian *marketing* dapat melakukan *recheck availability* atau mengecek ketersediaan barang.

Pada *default form Outgoing Shipping* tombol *force availability* masih muncul, proses ini dilakukan secara manual untuk mengadakan ketersediaan barang sedangkan pada perancangan form *Outgoing Shipping* tidak ditampilkan karena proses yang berjalan pada perusahaan *make to order* sehingga form *sales order* akan *ter-generate* menjadi form *manufacturing order* jika stok barang tidak tersedia.

Product	Quantity	Unit of Measure	Destination Location	Availability	Status
PJ-4-BOREW	100	Unit(s)	Partner Locations/Customer	0.0 Unit(s)	Waiting Another Move
PJ-4-BOLT	100	Unit(s)	Partner Locations/Customer	0.0 Unit(s)	Waiting Another Move
PJ-4-NUT	100	Unit(s)	Partner Locations/Customer	0.0 Unit(s)	Waiting Another Move
PJ-4-PIN	100	Unit(s)	Partner Locations/Customer	0.0 Unit(s)	Waiting Another Move

Gambar V.47 Default form *Outgoing Shipping*
(Sumber: Localhost/Odoo, 2018)

Product	Quantity	Unit of Measure	Destination Location	Availability	Status
PJ-4-BOREW	100	Unit(s)	Partner Locations/Customer	0.0 Unit(s)	Waiting Another Move
PJ-4-BOLT	100	Unit(s)	Partner Locations/Customer	0.0 Unit(s)	Waiting Another Move
PJ-4-NUT	100	Unit(s)	Partner Locations/Customer	0.0 Unit(s)	Waiting Another Move
PJ-4-PIN	100	Unit(s)	Partner Locations/Customer	0.0 Unit(s)	Waiting Another Move

Gambar V.48 Perancangan form *Outgoing Shipping*
(Sumber: Hasil Analisis, 2018)

8. Manufacturing Order

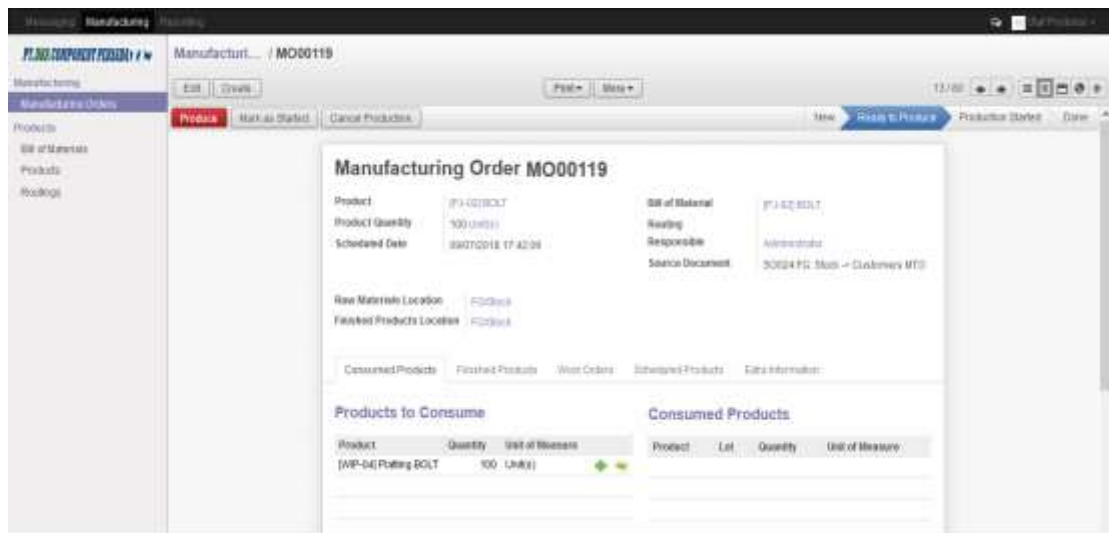
Form *sales order* akan ter-generate menjadi *Manufacturing Order* jika stok barang tidak tersedia. Proses ini ditindaklanjuti oleh bagian produksi. Pada *default form manufacturing order*, bagian produksi masih bisa melakukan *check availability* dan *cancel production* sedangkan pada perancangan form *manufacturing order* bagian produksi hanya dapat melihat form *manufacturing order* tanpa melakukan proses produksi. Hal ini dikarenakan pembahasan tidak mencakup proses produksi sehingga yang dibutuhkan hanya tombol *force*

reservation.

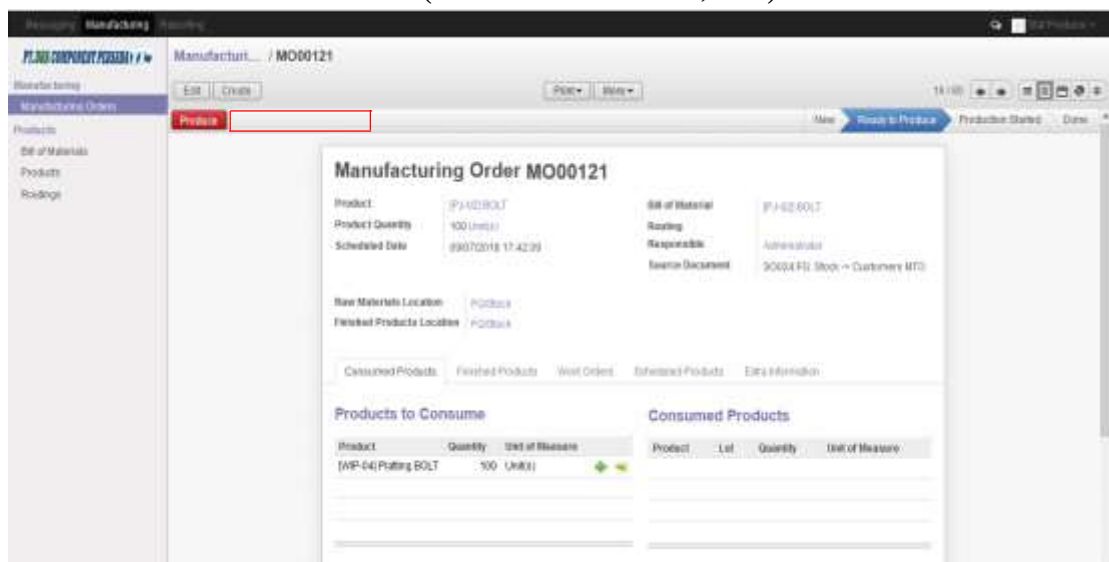
Gambar V.49 Default form *Manufacturing Order*
(Sumber: Localhost/Odoo, 2018)

Gambar V.50 Perancangan form *Manufacturing Order*
(Sumber: Hasil Analisis, 2018)

Setelah menekan tombol *force reservation*, pada default *produce* tombol *mark as started* masih muncul sedangkan pada perancangan sudah tidak ditampilkan. Hal ini dikarenakan tidak membahas proses produksi sehingga yang dibutuhkan hanya tombol *produce*.



Gambar V.51 Default produce
(Sumber: Localhost/Odoo, 2018)



Gambar V.52 Perancangan produce
(Sumber: Hasil Analisis, 2018)

9. Melakukan pengiriman

Pengiriman barang dilakukan secara manual dengan mencetak *delivery order* atau surat jalan.

PT. DRA COMPONENT PERSADA  Dikun Research & Analyst

PT DRA Component Persada
Jl. Raya Cileungsi Setu RT
001RW 04 Pasirangin
Bogor ,B
Indonesia

Customer Address:
PT. BS INDONESIA
Jl. Jababeka X Blok F7 No.26
Kawasan Industri Jababeka, Harjamekar
Cikarang Utara 555
Indonesia
021-4444254
021-4444254

Finish Good: Delivery Orders: FG/OUT/00012

Order (Origin)	State	Commitment Date	Scheduled Date
SO028	Transferred	09/10/2018 13:24:54	09/09/2018 17:42:08

Product	Quantity	Source	Barcode
[PJ-01] SCREW	100 Unit(s)	FG/Stock	
[PJ-02] BOLT	100 Unit(s)	FG/Stock	
[PJ-03] NUT	100 Unit(s)	FG/Stock	
[PJ-04] PIN	100 Unit(s)	FG/Stock	

Gambar V.53 Default Delivery Order Report
(Sumber: Localhost/Odoo, 2018)

PT. DRA COMPONENT PERSADA  Dikun Research & Analyst

PT DRA Component Persada
Jl. Raya Cileungsi Setu RT
001RW 04 Pasirangin
Bogor ,B
Indonesia

Customer Address:
PT. BS INDONESIA
Jl. Jababeka X Blok F7 No.26
Kawasan Industri Jababeka, Harjamekar
Cikarang Utara 555
Indonesia
021-4444254
021-4444254

Finish Good: Delivery Orders: FG/OUT/00012

Order (Origin)	State	Commitment Date	Scheduled Date
SO034	Transferred	09/10/2018 13:25:22	09/09/2018 17:42:08

Product	Quantity	Source	Barcode
[PJ-01] SCREW	100 Unit(s)	FG/Stock	
[PJ-02] BOLT	100 Unit(s)	FG/Stock	
[PJ-03] NUT	100 Unit(s)	FG/Stock	
[PJ-04] PIN	100 Unit(s)	FG/Stock	

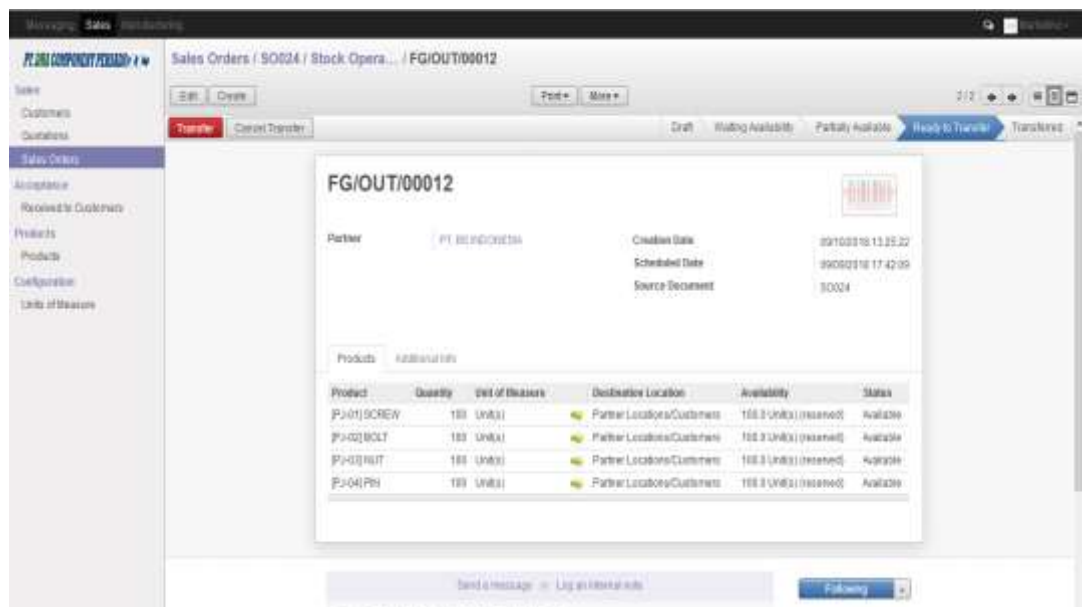
Received By _____ Security _____ Deliver By _____ Approved By _____

(_____) (_____) (_____) Marketing

Gambar V.54 Perancangan Delivery Order Report
(Sumber: Hasil Analisis, 2018)

10. Melakukan *transfer* barang pada sistem

Setelah barang sudah sampai ke *customer*, *customer* akan menghubungi staf *marketing* via telepon untuk memberikan konfirmasi bahwa barang yang dipesan telah diterima lalu staf *marketing* akan melakukan *transfer* barang pada sistem. Pada proses ini bagian *marketing* dapat melakukan *transfer* barang *full delivery* maupun *partial delivery* sesuai dengan kenyataan pada saat melakukan pengiriman barang kepada *customer* yang dituju.



Gambar V.55 *Transfer* barang pada sistem
(Sumber: Hasil Analisis, 2018)

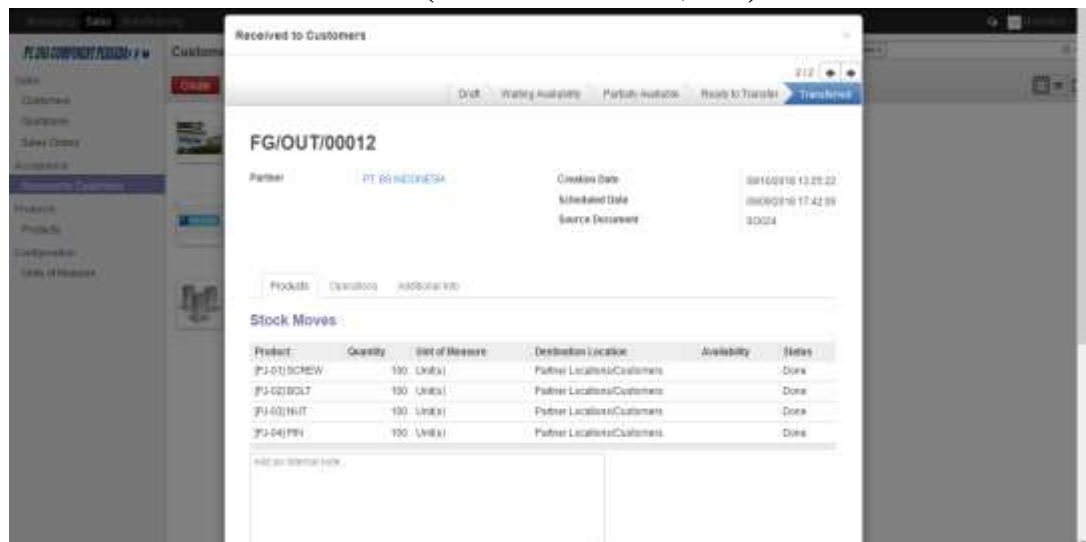
11. Penerimaan barang

Setelah melakukan *transfer* barang pada sistem, form yang telah ter-*transfer* akan muncul pada menu item *Acceptance – Received to Customers* dengan status



transferred

Gambar V.56 Perancangan penerimaan barang
(Sumber: Hasil Analisis, 2018)



Gambar V.57 Perancangan form *received to customers*
(Sumber: Hasil Analisis, 2018)

5.12 Konfigurasi Sistem Penjualan Produk Usulan

Konfigurasi dilakukan guna melakukan penyesuaian antara proses penjualan produk berjalan dengan proses penjualan produk usulan menggunakan Odoo. Adapun konfigurasi yang dilakukan yaitu:

1. Konfigurasi Modul *Sales Management*



Gambar V.58 Konfigurasi Modul *Sales Management*

(Sumber: Hasil Analisis, 2018)

Untuk konfigurasi menu *Sales* terdapat pada menu *Setting > Configuration > Sales*. Pada proses pembuatan *invoice* diatur setelah dan berdasarkan pembuatan *delivery orders*, sementara pada proses *quotation* dan *sales order* pembuatan *delivery orders* dibuat secara otomatis dari *sales order* dan dapat melakukan penjualan produk dengan jenis *unit of measure* yang berbeda.

2. Konfigurasi Data *Master*

5.13 Data Product

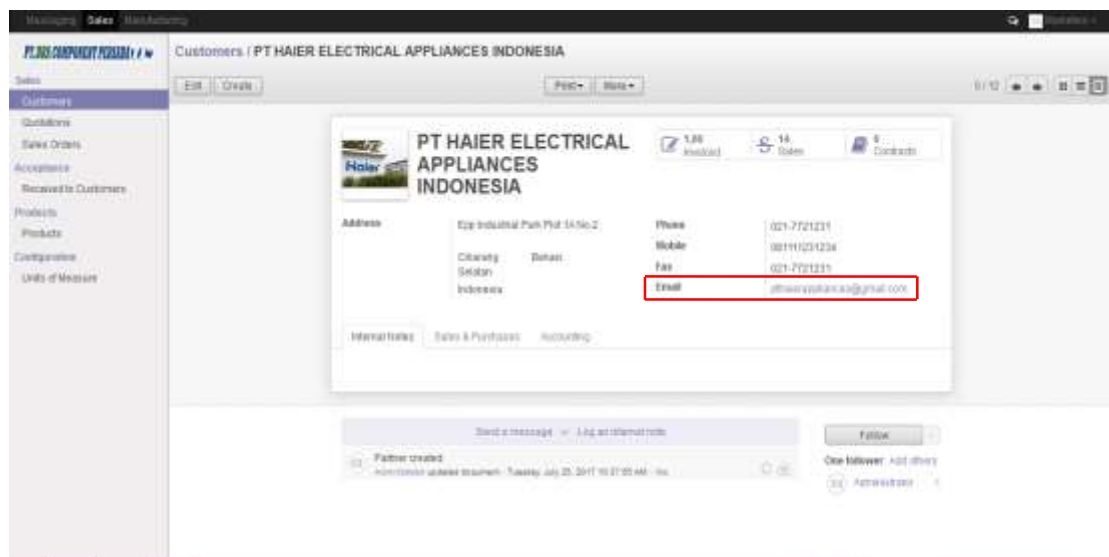
Pada data *master product* diatur menjadi *can be sold* yang artinya bahwa produk tersebut dapat dijual dan *routes* atau alur perpindahan data produk. Adapun *product* diatur dengan *routes manufacture* dan *make to order*. *Manufacture* menunjukkan bahwa *product* yang diproduksi merupakan produk manufaktur dan dapat membuat kebutuhan *product* pada *sales order* ter-generate menjadi *manufacturing order*. Sedangkan *Make to order* menunjukkan bahwa *product* dibutuhkan berdasarkan kegiatan



Gambar V.59 Konfigurasi Data *Master Product*
(Sumber: Hasil Analisis, 2018)

5.14 Data Customer

Pada data *master customer* diatur email-nya sehingga *customer* dapat menerima *Quotation* melalui *email* dari sistem.



Gambar V.40 Konfigurasi Data *Master Customer*
(Sumber: Hasil Analisis, 2018)

3. Konfigurasi Hak User

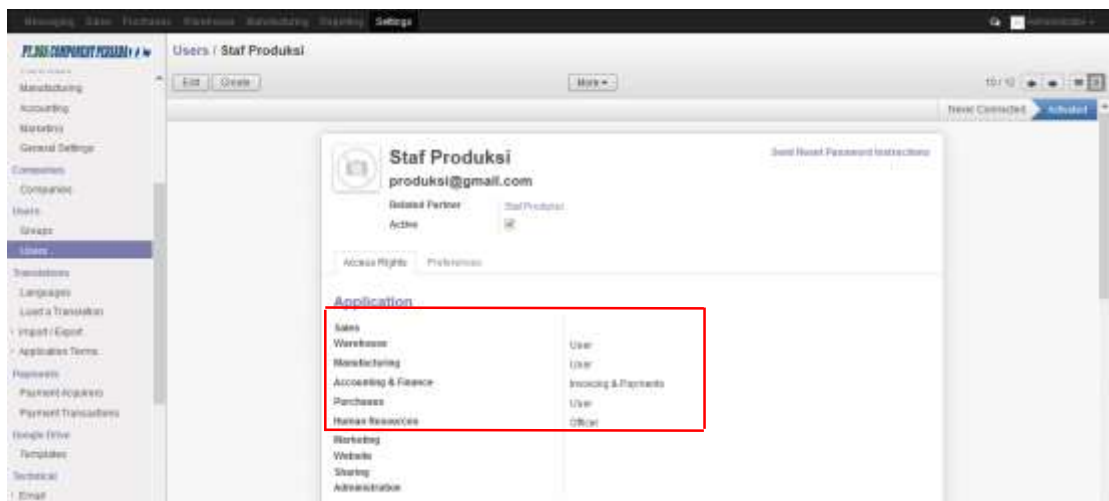
a. Staf Marketing



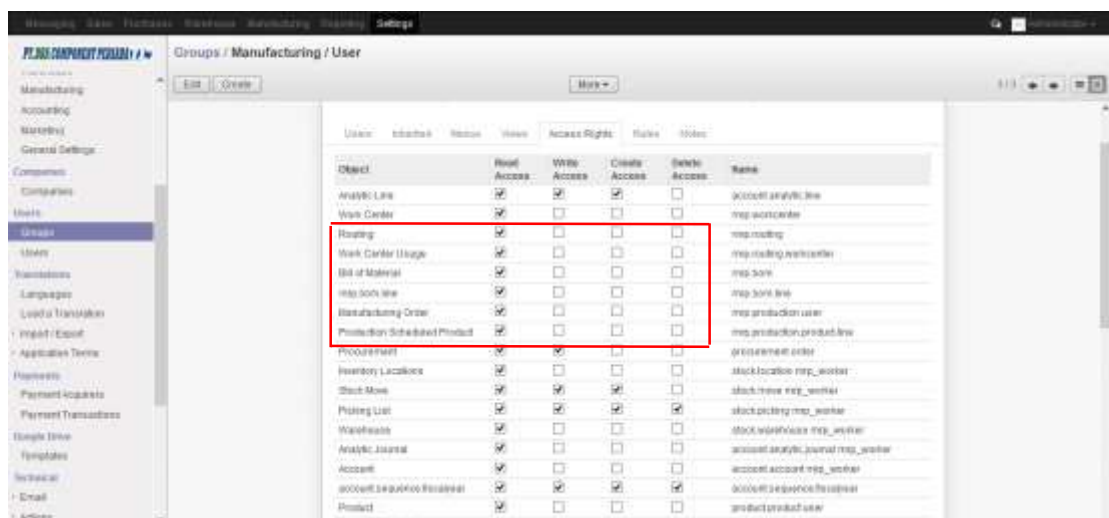
(Sumber: Hasil Analisis, 2018)

Gambar V.41 Konfigurasi Hak Akses Staf Marketing
(Sumber: Hasil Analisis, 2018)

b. Staf Produksi



Gambar V.42 Konfigurasi Hak Akses Staf Produksi
(Sumber: Hasil Analisis, 2018)



Gambar V.43 Konfigurasi Hak Akses *Group Manufacturing User*
(Sumber: Hasil Analisis, 2018)

Pada bagian integrasi antara aktor *marketing* dengan bagian *manufacturing*, *marketing* hanya dapat melihat *manufacturing order* tidak dapat melakukan aksi selain hal tersebut.

BAB VI

PENUTUP

1.2 Kesimpulan

Berdasarkan hasil penelitian selama proses pengumpulan dan pengolahan data, analisis dan desain sistem, serta perancangan sistem usulan maka dapat diambil kesimpulan sebagai berikut:

1. Pada proses pengajuan permintaan produksi (*manufacturing order*) yang dikelola oleh Bagian *Marketing* dapat dibuat secara otomatis karena ter-*generate* oleh *sales order* yang dikerjakan oleh Bagian *Marketing*, sehingga diharapkan Bagian Produksi dapat langsung mengetahui jumlah permintaan produksi.
2. Sistem penjualan produk menggunakan Odoo menjadi terkomputerisasi dan telah menerapkan media penyimpanan basis data yang terintegrasi, sehingga diharapkan dapat membantu dalam proses pengelolaan dan peng-*input*-an data.
3. Sistem memberikan keamanan terhadap data dimana setiap *user* memiliki akses yang berbeda. Sehingga mengurangi resiko terhadap penyalahgunaan data perusahaan.

1.3 Saran

Adapun saran yang perlu diberikan untuk menerapkan sistem ERP pada pembelian bahan baku ini adalah sebagai berikut:

1. Diharapkan dapat mengimplementasikan hasil penelitian berupa sistem ERP menggunakan Odoo pada sistem penjualan produk di Departemen *Marketing* pada PT DRA Component Persada.
2. Diharapkan bagi pembaca tugas akhir ini dapat melakukan pengembangan selanjutnya terhadap sistem ERP menggunakan Odoo yaitu pengembangan Modul *Purchasing* dan *manufacturing* sehingga integrasi sistem di PT DRA Component Persada segera terwujud.

DAFTAR PUSTAKA

- Arsip PT DRA Component Persada.
- Brady dan Loonam. 2010. *Exploring the Use of Entity Relationship Diagram as a Technique to Support Grounded Theory Inquiry*. Qualitative Research in Organization and Management.
- Dennis, Alan. 2010. *Analysis and Design, 5th Edition*. New York: John Wiley.
- Enterprise, Jubilee. 2017. *Mastering Python*. Jakarta: PT Elex Media Komputindo.
- H.M, Jogyianto. 2010. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi Offset.
- Johnny Ibrahim, Prof. Dr dan Jonaedi Efendi, Dr. *Metode Penelitian Hukum*. Cetakan kedua. Depok: Prenadamedia Group
- Kamus Besar Bahasa Indonesia. 2016. *Implementasi*. <http://kbbi.web.id/implementasi>. Diakses 9 Mei 2018.
- Kamus Besar Bahasa Indonesia. 2016. *Pengelolaan*. <http://kbbi.web.id/pengelolaan>. Diakses 9 Mei 2018.
- Mujilan, Agustinus. 2013. *Analisis Dan Perancangan Sistem: Perspektif Kompetensi Akuntansi*. Edisi Pertama. Madiun: Universitas Widya Mandala
- Mujilan, Agustinus. 2015. *Sistem Informasi Akutansi: Teori dan Wawasan dalam Dunia Elektronik*. Edisi Ketiga. Madiun: Universitas Widya Mandala
- Kusumawati, Dewi M.Kom. *Basis Data dengan PostgreSQL*. Yogyakarta: Graha Ilmu.
- Kotler, dan Keller. 2012. *Manajemen Pemasaran*. Edisi 12. Jakarta: Erlangga
- Ragil, Wukil. 2010. *Pedoman Sosialisasi Prosedur Standar*. Jakarta: Mitra Wacana Media
- Rizky, Soetam. 2011. *Konsep Dasar Rekayasa Perangkat Lunak (Software Reengineering)*. Jakarta: Prestasi Pustaka Publisher.
- Rusdiana, Dr. H.A dan Irfan, Moch. 2014. *Sistem Informasi Manajemen*. Cetakan Pertama. Bandung: CV Pustaka Setia.

- Sarosa, Samiaji. 2017. *Metodologi Pengembangan Sistem Informasi*. Jakarta: Indeks Jakarta
- Satzingeretal, Jackson dan Burd. 2012. *System Analysis and Design in a Changing World*. USA: Course Technology.
- Sujarweni, V Wiratna. 2015. *Sistem Akuntansi*. Cetakan Pertama. Yogyakarta: Pustaka Baru Press.
- Sunyoto, Danang, Drs. 2012. *Dasar-Dasar Manajemen Pemasaran*. Cetakan Pertama. Yogyakarta: CAPS
- Susanto, Azhar Prof. Dr. 2017. *Sistem Informasi Akutansi: Pemahaman Konsep secara Terpadu*. Edisi Pertama. Bandung: Lingga Jaya
- Susanto, Azhar Prof. Dr. 2017. *Teknologi Informasi Untuk Aplikasi Bisnis dan Akuntansi*. Cetakan Pertama. Pertama. Bandung: Lingga Jaya

Situs:

- Santosa, Budi Agung. 2010. *Menimbang-nimbang Produk ERP (Enterprise Resource Planning)*. <http://blog.alphamedia.co.id/2010/02/menimbang-nimbang-produk-erp.html>. (Tanggal Akses: 09 Mei 2018)
- Wikipedia, Definisi Open Source
https://wikipedia.org/wiki/Sumber_terbuka (Tanggal Akses: 07 Mei 2018)
- Wikipedia, Definisi Odoo
<https://wikipedia.org/wiki/Odoo> (Tanggal Akses; 06 Mei 2018).
- Postgresql.org, Tipe Data PostgreSQL
<https://www.postgresql.org/docs/9.2/static/datatype.html> (Tangal Akses: 08 Mei 2018)

Jurnal:

- Annisa Utami, Nia Ambarsari, dan R.Wahjoe Witjaksono. 2015. Penerapan Sistem *Sales Management* Menggunakan OpenERP Pada PT XYZ Dengan Metode Spiral

- Adam Ramadhan, Luciana Andrawina, R.Wahjznoe Witjaksono. 2016. Perancangan Sistem *Procurement* Dengan *Soft System Methodology* Di Rumah Sakit Muhammadiyah Bandung
- Aldi Mustafri, Deden Witarsyah, R.Wahjoe Witjaksono. 2016. Pengembangan Odoo Modul *Warehouse* Pada Gudang PT Tarumex Menggunakan Metode *Rapid Application Development*
- Dika Anugerah P, Ari Yanuar R, dan R.Wahjoe Witjaksono. 2016. Penerapan Sistem *Sales Management* Menggunakan OpenERP Dengan Metode *Rapid Application Development* (Study Kasus: PT. Genta Trikaya)
- Risa Mutiara, Yuli Adam P, dan Muhammad Azani. 2017. Implementasi Aplikasi *Enterprise Resource Planning* Odoo Modul *Sales Management* Menggunakan Metode *Rapid Application Development* Di UD. Permatasari

LAMPIRAN A
BUKTI WAWANCARA

Tanggal	Jam	Pewawancara	Yang diwawancarai (Jabatannya)	Lokasi Wawancara	Topik Wawancara
18-11-2017	13:30	Azmi Nurul Azizah	Bapak Sugeng (Kepala Divisi Produksi)	Ruangan <i>Meeting</i>	- Proses Pengajuan Permintaan Produksi
25-11-2017	14:00	Azmi Nurul Azizah	Bapak Muchlis (<i>General Manager</i>)	Ruangan <i>Meeting</i>	- Proses Penawaran Harga dan penerimaan <i>Purchase Order</i>
					- Proses penjualan
					- Proses pembuatan <i>Sales Confirmation</i> untuk <i>customer</i>
					- Proses pembuatan <i>Delivery Order</i> dan penerimaan barang sampai ke <i>customer</i>

TRANSKRIP WAWANCARA

Nama Narasumber : Bapak Sugeng (Kepala Divisi Produksi)

Tanggal : 18 November 2017

Jam : 14.00 – 15.00 WIB

Tempat Wawancara : Ruangan *Meeting*

Topik Wawancara : Proses pengajuan permintaan produksi

Pewawancara : Proses produksi di PT DRA ini seperti apa pak?

Narasumber : Proses produksi di sini pake sistem *make to order*, jadi kalau ada pesanan dari *customer* baru dah kami melakukan produksi.

Pewawancara : Berarti kalau gitu, tidak menstok produk jadi ya pak?

Narasumber : Iya betul kami tidak menstok nya.

Pewawancara : Kalau untuk proses permintaan produksinya seperti apa pak?

Narasumber : Ya kalau ada pesanan dari *customer*, bagian *marketing* mengajukan permintaan produksinya cuma ngasih *copy-an form PO* dari *customer* aja

Pewawancara : Lembar tersebut ditujukan ke siapa pak?

Narasumber : Lembar itu ditujukan ke bagian PPIC. Nah setelah menerima *PO customer*, staf PPIC akan mengolah *PO* tersebut menjadi form *Job Order* dan menyerahkannya ke kita bagian Produksi sebagai permintaan produksi.

Pewawancara : Kira-kira untuk sebulan berapa kali lembar tersebut dibuat pak?

Narasumber : wah tidak tentu juga sih, tapi kalau berdasarkan perkiraan sih yaa paling sekitar kurang lebih 15 lembar mungkin ya perhari.

Pewawancara : Lumayan banyak juga ya pak. Itu setiap hari pak?

Narasumber : Ya maksudnya perhari 15 lembar itu ya dalam sehari kurang lebih bisa sampai 15 lembar tapi kita tidak tahu hari nya kapan. Misal hari Senin ada pesanan nih dari beberapa *customer* ya itu bisa sampai 15 *PO* dari *customer* berarti kan ada 15 lembar *copy-an* nya buat pengajuan permintaan produksi dan kita nerima *form Job Order* juga 15 lembar tapi belum tentu keesokan hari nya ada pesanan karena kan yang pesanan di hari Senin itu kita proses dulu tapi bisa jadi ada permintaan produksi berturut-turut sih.

Pewawancara : Berarti pengajuan permintaan produksi nya hanya menggunakan kertas itu saja ya pak?

Narasumber : Iya kita melakukan proses produksi berdasarkan *form job order* dalam bentuk kertas dari PPIC

Pewawancara : Tapi bisa jadi terjadi masalah tidak pak dengan menggunakan kertas itu pak?

Narasumber : ya bisa aja terjadi masalah, pertama mungkin bagian *marketing* sendiri lupa ngasih *copy-an* tersebut karena ke selip atau apa kan kita tidak tahu. Kedua proses permintaan produksi tersebut jadi lama karena pas ada pesanan dari *customer* bagian produksi gak langsung tahu pesanan tersebut tapi harus diolah dulu sama PPIC sedangkan di PPIC pun *copy-an form PO customer* masih ditulis tangan di *form job order* yang sudah di fotokopi banyak. Jadi *form job order* hasil tulis ulang aja

Pewawancara : Oh seperti itu ya pak, oke-oke saya sudah mengerti pak. Terima kasih atas informasi dan waktunya ya pak.
Narasumber : Iya sama-sama dek.

Nama Narasumber : Bapak Muchlis (*General Manager*)
Tanggal : 25 November 2017
Jam : 14.00 – 16.30 WIB
Tempat Wawancara : Ruangan *Meeting*
Topik Wawancara : Proses penjualan produk, proses penawaran harga, proses penerimaan *Purchase Order*, proses pembuatan *Sales Confirmation*, proses pembuatan *Delivery Order* dan proses penerimaan barang sampai ke *customer*

Pewawancara : Penjualan produk yang berlangsung di PT DRA seperti apa pak?
Narasumber : Ya biasanya kita memberikan penawaran harga atas produk-produk yang kita jual ke beberapa *customer* dengan menghubungi nya melalui telepon. Kalau penawaran harga yang kita diajukan tersebut disetujui biasanya keesokan harinya *customer* melakukan *order* ke kita dengan mengirimkan *Purchase Order*. Setelah itu kita buat *Sale Confirmation*.

Pewawancara : Oh berarti untuk mendapatkan *PO* dari *customer* harus memberikan penawaran harga dulu ya pak?

Narasumber : Ya tidak juga, penawaran harga kita berikan untuk *customer* baru yang kita jadikan target kerja-sama atau untuk *customer* yang minta penawaran harga. Ya kalau *customer* tetap atau yang udah sering *order* mah biasanya langsung ngasih *PO* saja gitu.

Pewawancara : Oh iya iya pak. Untuk penawaran harga yang dilakukan via telepon seperti itu ada kendalanya tidak pak?

Narasumber : Ya ada, tentu nya proses penawaran harga ini cukup menyita waktu ya karena kan kita harus nelepon *customer* baru satu per satu.

Pewawancara : Berarti untuk penawaran harga ini belum ada form nya pak?

Narasumber : Iya belum ada, sama halnya dengan SC

Pewawancara : Memangnya seperti apa pak SC ?

Narasumber : Ya kan kita dapat PO nih dari *customer* nah trus kita buat SC nya lewat email cuma berupa kata-kata kalau kita bisa dan sanggup untuk melakukan produksi berdasarkan list produk-produk yang ingin dibeli *customer* ataupun konfirmasi bahwa barang yang diminta bahan baku nya habis jadi kan harus melakukan pembelian bahan baku dulu nah kita negosiasi sama *customer* mau gak kalau pengiriman barangnya di undur beberapa hari. Ya seperti itu lah contohnya.

Pewawancara : Oh iya iya pak. Berarti penawaran harga dan SC sama-sama tidak belum ada form nya pak?

Narasumber : Iya betul.

Pewawancara : Selanjutnya setelah SC apa lagi pak?

Narasumber : Nah setelah konfirmasi ke *customer*, bagian *marketing* akan memberikan *copy-an* form PO dari *customer* tadi ke bagian PPIC sebagai pengajuan permintaan produksi atas pesanan *customer* tersebut, nah *copy-an* form PO tersebut diolah menjadi *Job Order* yang nantinya akan diserahkan ke bagian produksi.

Pewawancara : Oh jadi form PO dari *customer* nya tidak proses dulu pak?

Narasumber : Tidak, cuma di fotokopi form PO nya terus langsung di kasih ke bagian PPIC

Pewawancara : Bisa terjadi masalah tidak pak dengan langsung memberikan begitu?

Narasumber : Ya bisa terjadi masalah kaya misalnya kadang staf *marketing* lupa udah ngasih *copy-an* itu atau belum atau lupa naruh dimana atau staf PPIC lupa juga naruh dimana terus tercecer ya kan berarti *marketing* harus ngasih ulang gitu, ya semacam itu lah.

Pewawancara : Selanjutnya pak?

Narasumber : Selanjutnya setelah produk sudah jadi, staf produksi akan memberi tahu kepada staf PPIC kalau produk-produk berdasarkan PO dari *customer* X udah jadi dan staf PPIC akan mempersiapkan produk yang akan dikirim lalu staf PPIC memberi tahu kepada staf *marketing* bahwa barang telah siap dikirim.

Pewawancara : Setelah diberi tahu produknya telah selesai, setelah itu apa lagi pak?

Narasumber : Nah setelah itu staf *marketing* membuat surat jalan atau *delivery order* berdasarkan dari *PO customer* tersebut sebagai surat pengiriman barang. Lalu dilakukan lah tuh pengiriman barang beserta dengan surat jalan atau *delivery order*.

Pewawancara : Kira-kira dalam sebulan berapa lembar pembuatan *delivery order* pak?

Narasumber : Ya tidak tentu, kan tergantung pengirimannya karena 1 PO dari *customer* bisa jadi dilakukan 2 kali pengiriman sesuai keinginan *customer* nya nah setiap

pengirimannya kan kita buat *delivery order* sebagai surat jalan bersama barang yang dikirim. Tapi kalau di perkirakan sih kurang lebih 90 lembar lah per minggu.

Pewawancara : Oh seperti itu ya pak. Oiya pak, untuk pemberitahuan bahwa barang telah sampai ke *customer* seperti apa pak?

Narasumber : Setelah melakukan pengiriman barang biasanya *customer* akan menelepon kita untuk memberi informasi bahwa barang yang dipesannya sudah sampai ke sana, seperti itu.

Pewawancara : Oh iya pak, saya paham. Kalau begitu terima kasih ya pak atas informasi dan waktunya.

Narasumber : Iya sama-sama ya

LAMPIRAN B
BUKTI OBSERVASI

Tanggal	Jam	Pengamat	Yang diamati (Jabatannya)	Lokasi Observasi	Kegiatan Diamati	Contoh Dok.dikumpulkan
04-11- 2017	13:30	Azmi Nurul Azizah	Bapak Sugeng (Kepala Divisi Produksi)	Ruangan <i>Meeting</i>	Proses pengamatan Pengajuan Permintaan Produksi	
11-11- 2017	15:30	Azmi Nurul Azizah	Bapak Muchlis (<i>General Manager</i>)	Ruangan <i>Meeting</i>	Proses pengamatan Penawaran ke <i>customer</i>	
					Proses Pembuatan <i>Sales Confirmation</i>	<i>Sales Confirmation</i> (lihat Bab IV hal 80)
					Proses pengamatan Penerimaan <i>form PO</i> dari <i>customer</i>	<i>Form Purchase Order</i> dari <i>customer</i> (lihat Bab IV hal 79)
18-11- 2017	14.00	Azmi Nurul Azizah	Bapak Muchlis (<i>General Manager</i>)	Ruangan <i>Meeting</i>	Proses Pembuatan <i>Delivery Order</i>	<i>Form Delivery Order</i> (Lihat Bab IV hal
					Proses pengamatan Penerimaan Barang sampai ke <i>customer</i>	

Purchase Order

Kawasan Industri Mitra Karawang
Jl. Mitra Karawang Timur II
Blok D-24, Karawang

PO Number
DLPO170110002/MODEM-002 ✓

Vendor
PT. DRA COMPONENT PERSADA
Wisma Harben Blok 2E Jl Raya Pasar
Minggus Pandoren Jakarta Selatan DKI
Jakarta
12780 12780

Ship To
Kawasan Industri Mitra Karawang
Jl. Mitra Karawang Timur II
Blok D-24, Karawang

PO Date
10 Nop 2017

Currency
Rupiah

Terms
Net 30

FOB
Destination

Ship Via
by supplier

Expected Date
12 Nop 2017

Rate
1

Item	Description	Qty	Unit Price	Disc %	Tax	Amount
50170B860H	Screw ST 2.8*8.5 ✓					192.000

Say Dua ratus sebelas ribu dua ratus

Description
1. payment 30 days after received item and IV
2. DELIVERY 17/11/2017
3. MARK ON LABEL H00M MODEM-002 ✓

Sub Total : 192.000
Discount : 0
PPN (VAT) : 19.200
Estimated : 0
Total Order : 211.200

Requested By
Rahm
Date: 

Reviewed By
Dept. Manager
Date: 

Reviewed By
Finance Dept.
Date:

Approved By
Management
Date:

Acknowledged By
Vendor
Date:

1. Vendor 2. Finance 3. Requester 4. File

Lampiran VIII *Form Purchase Order*

(Sumber: PT DRA Component Persada, 2017)

PT. DRA Component Persada

Jl. Raya Cileungsi Setu, Kampung Pasirangin, RT.001/004 Desa Pasirangin,
Kec. Cileungsi, Kab. Bogor – Jawa Barat.

6 Januari 2017

No :

Subject: Sale Confirmation tanggal 1 Januari 2017

Purchase Manager

PT. HARTONO ISTANA TEKNOLOGI

JL. KHR. Asnawi PO BOX 126,

Kaliwungu, Kudus - 59332

Dengan Hormat,

Terima Kasih atas pesanan dengan No. PO 223 pada tanggal 1 Januari 2017 sebagai berikut:

No	Tipe Barang	Qty	Harga/unit	Total Harga	Keterangan

Pesanan sedang dalam proses produksi dan akan di kirim pada tanggal 29 Januari 2017. Kami selalu menjaga kualitas produk hingga proses pengiriman. Kami harap anda merasa puas dengan produk serta pelayanan kami.

Kami sangat berharap, jika Perusahaan anda tetap memperpanjang kerja-sama dengan kami di masa depan.

Catatan:

Regards

Marketing Manager

PT DRA Component Persada

Lampiran IX Sales Confirmation

(Sumber: Rekonstruksi Penulis dari Hasil Observasi, 2018)

DRA COMPONENT

Delivery Date: 05 Mei 2017

PO Number: 014/PC/KIM/IV/2017

Delivery Order

PT KIRANA JAYA MUKTI
 Jl. Ramin Raya Blok G-10
 Delta Siucek B Industrial Park
 Jaya Mukti Cikarang Pusat

NO.	DESCRIPTION	QTY	PACKING
1	PO NO : 0077/PO/KIM/III/2017 MHWRLT00020 INSERT BOLT M6	51 PCS	1 PLSTK
1	PO NO : 0134/PO/KIR/IV/2017 MHWRLT00020 INSERT BOLT M6	300 PCS	1 PLSTK

Received By: [Signature]
 Security: [Signature]
 Delivery By: [Signature]
 Approved By: [Signature]

Where: Warehouse
 How: [Signature]
 Follow & Check: [Signature]
 Ship: Warehouse

Lampiran X *Delivery Order*
 (Sumber: PT DRA Component Persada, 2017)

LAMPIRAN C

SOURCE CODE

1. Sub Menu *Customer*

```
<?xml version="1.0"?>
<tree string="Contacts">
    <field name="display_name"/>
    <field name="function" invisible="1"/>
    <field name="phone"/>
    <field name="email"/>
    <field name="user_id" invisible="1"/>
    <field name="is_company" invisible="1"/>
    <field name="country_id" invisible="1"/>
    <field name="parent_id" invisible="1"/>
</tree>
```

2. Form *Customer*

```
<?xml version="1.0"?>
<form string="Partners">
    <sheet>
        <field name="image" widget="image" class="oe_left
oe_avatar" options="{&quot;preview_image&quot;;
&quot;image_medium&quot;, &quot;size&quot;: [90, 90]}"/>
        <div class="oe_title oe_left">
            <div class="oe_edit_only">
                <label for="name"/> (
                <field name="is_company"
on_change="onchange_type(is_company)" class="oe_inline"/> <label
for="is_company" string="Is a Company?"/>)
            </div>
            <h1>
                <field name="name" default_focus="1"
placeholder="Name"/>
            </h1>
            <field name="parent_id" placeholder="Company"
domain="[('is_company', '=', True)]" context='{&quot;default_is_company&quot;: True,
&quot;default_supplier&quot;: supplier, &quot;default_customer&quot;: customer}'"
attrs="{&quot;invisible&quot;: [(&quot;is_company&quot;, '=', True), (&quot;parent_id&quot;, '=', False)]}"
on_change="onchange_address(use_parent_address, parent_id)"/>
                <field name="category_id" widget="many2many_tags"
placeholder="Tags..." invisible="1"/>
            </div>
            <div class="oe_right oe_button_box" name="buttons"> </div>

        <group>
            <group>
```

```

False)]]"/>
<div attrs="{ 'invisible': [('parent_id','=', False)]}"
name="div_type">
    <field class="oe_inline" name="type"/>
</div>

<label for="street" string="Address"/>
<div>
    <div class="oe_edit_only">
        <field name="use_parent_address" class="oe_inline"
on_change="onchange_address(use_parent_address, parent_id)"
attrs="{ 'invisible': ['|', ('is_company', '=', True), ('parent_id', '=', False)]}" />
        <label for="use_parent_address" attrs="{ 'invisible':
['|', ('is_company', '=', True), ('parent_id', '=', False)]}" />
    </div>
    <button name="open_parent" type="object"
string="(edit company address)" class="oe_link oe_edit_only"
attrs="{ 'invisible': ['|', ('parent_id', '=',
False), ('use_parent_address', '=', False)]}" />
        <field name="street" placeholder="Street..."
attrs="{ 'readonly': [('use_parent_address', '=', True)]}" />
        <field name="street2" attrs="{ 'readonly':
[('use_parent_address', '=', True)]}" />
        <div class="address_format">
            <field name="city" placeholder="City" style="width:
40%" attrs="{ 'readonly': [('use_parent_address', '=', True)]}" />
            <field name="state_id" class="oe_no_button"
placeholder="State" style="width: 37%" options="{ '&no_open&quot;;
True}" on_change="onchange_state(state_id)" attrs="{ 'readonly':
[('use_parent_address', '=', True)]}" />
            <field name="zip" placeholder="ZIP" style="width:
20%" attrs="{ 'readonly': [('use_parent_address', '=', True)]}" />
        </div>
        <field name="country_id" placeholder="Country"
class="oe_no_button" options="{ '&no_open&quot;; True,
&no_create&quot;; True}" attrs="{ 'readonly':
[('use_parent_address', '=', True)]}" />
    </div>
    <field name="website" widget="url" placeholder="e.g.
www.odoo.com" invisible="1" />
</group>
<group>
    <field name="function" placeholder="e.g. Sales Director"
attrs="{ 'invisible': [('is_company', '=', True)]}" />
    <field name="phone" />

```

```

        <field name="mobile"/>
        <field name="fax"/>
        <field name="user_ids" invisible="1"/>
        <field name="email" widget="email" attrs="{ 'required':
[(('user_ids','!=', []))]}"/>
        <field name="title" options="{ '&quot;no_open&quot;:
True}" invisible="1"/>
    </group>
</group>

<notebook colspan="4">
    <page string="Contacts" attrs="{ 'invisible':
[(('is_company','!=',False), ('child_ids','!=', []))]} autofocus="autofocus"
invisible="1">
        <field name="child_ids" mode="kanban"
context="{ 'default_parent_id': active_id, 'default_street': street,
'default_street2': street2, 'default_city': city, 'default_state_id': state_id,
'default_zip': zip, 'default_country_id': country_id, 'default_supplier':
supplier, 'default_customer': customer, 'default_use_parent_address':
True}">
            <kanban>
                <field name="color"/>
                <field name="name"/>
                <field name="title" />
                <field name="email"/>
                <field name="parent_id"/>
                <field name="is_company"/>
                <field name="function"/>
                <field name="phone"/>
                <field name="street"/>
                <field name="street2"/>
                <field name="zip"/>
                <field name="city"/>
                <field name="country_id"/>
                <field name="mobile"/>
                <field name="fax"/>
                <field name="state_id"/>
                <field name="has_image"/>
                <templates>
                    <t t-name="kanban-box">
                        <t t-set="color" t-
value="kanban_color(record.color.raw_value)"/>
                        <div t-att-class="color + (record.title.raw_value
== 1 ? ' oe_kanban_color_alert' : ')" style="position: relative">

```

```

<a t-if="! read_only_mode" type="delete"
style="position: absolute; right: 0; padding: 4px; display: inline-
block">X</a>

<div class="oe_module_vignette">
<a type="open">
<t t-if="record.has_image.raw_value ===
true">


</t>
<t t-if="record.image and
record.image.raw_value !== false">

</t>
<t t-if="record.has_image.raw_value ===
false and (!record.image or record.image.raw_value === false)">
<t t-if="record.is_company.raw_value
=== true">

</t>
<t t-if="record.is_company.raw_value
=== false">

</t>
</t>
</a>
<div class="oe_module_desc">
<div class="oe_kanban_box_content
oe_kanban_color_bglight oe_kanban_color_border">
<table class="oe_kanban_table">
<tr>
<td class="oe_kanban_title1"
align="left" valign="middle">
<h4><a type="open"><field
name="name"/></a></h4>
<i><div t-
if="record.function.raw_value">
<field
name="function"/></div></i>

```

```

                                <div><a t-
if="record.email.raw_value" title="Mail" t-att-
href=""mailto:' + record.email.value">
                                <field name="email"/>
                                </a></div>
                                <div t-
if="record.phone.raw_value">Phone: <field name="phone"/></div>
                                <div t-
if="record.mobile.raw_value">Mobile: <field name="mobile"/></div>
                                <div t-
if="record.fax.raw_value">Fax: <field name="fax"/></div>
                                </td>
                                </tr>
                                </table>
                                </div>
                                </div>
                                </div>
                                </div>
                                </t>
                                </templates>
                                </kanban>
                                <form string="Contact">
                                <sheet>
                                    <field name="image" widget="image"
class="oe_avatar oe_left" options="{&quot;preview_image&quot;;
&quot;image_medium&quot;}" />
                                    <div class="oe_title">
                                        <label for="name" class="oe_edit_only"/>
                                        <h1><field name="name" style="width:
70%"/></h1>
                                        <field name="category_id"
widget="many2many_tags" placeholder="Tags..." style="width: 70%"/>
                                        </div>
                                        <group>
                                            <field name="function" placeholder="e.g. Sales
Director"/>
                                            <field name="email"/>
                                            <field name="phone"/>
                                            <field name="mobile"/>
                                        </group>
                                        <div>
                                            <field name="use_parent_address"/><label
for="use_parent_address"/>
                                        </div>
                                        <group>
                                            <label for="type"/>

```

```

        <div name="div_type">
            <field class="oe_inline" name="type"/>
        </div>
        <label for="street" string="Address"
attrs="{ 'invisible': [('use_parent_address','=', True)] }"/>
        <div attrs="{ 'invisible':
[('use_parent_address','=', True)]}" name="div_address">
            <field name="street" placeholder="Street..."/>
            <field name="street2"/>
            <div class="address_format">
                <field name="city" placeholder="City"
style="width: 40%"/>
                <field name="state_id"
class="oe_no_button" placeholder="State" style="width: 37%"
options="{ '&quot;no_open&quot;': True}"
on_change="onchange_state(state_id)"/>
                <field name="zip" placeholder="ZIP"
style="width: 20%"/>
            </div>
            <field name="country_id"
placeholder="Country" class="oe_no_button"
options="{ '&quot;no_open&quot;': True, '&quot;no_create&quot;': True }"/>
        </div>
    </group>
    <field name="supplier" invisible="True"/>
    <field name="customer" invisible="True"/>
</sheet>
</form>
</field>
</page>
<page name="internal_notes" string="Internal Notes">
    <field name="comment" placeholder="Put an internal
note..."/>
</page>
<page name="sales_purchases" string="Sales &
Purchases">
    <group>
        <group>
            <field name="user_id"
context="{ 'default_groups_ref': ['base.group_partner_manager'] }"/>
            <field name="company_id"
groups="base.group_multi_company" options="{ 'no_create': True }"/>
        </group>
        <group>
            <field name="customer"/>
            <field name="supplier" invisible="1"/>

```

```

        </group>
        <group>
            <field name="ref" invisible="1"/>
            <field name="lang"/>
            <field name="date"/>
        </group>
        <group>
            <field name="active"/>
        </group>
    </group>
</page>
</notebook>
</sheet>
</form>

```

3. Sub Menu *Product*

```

<?xml version="1.0"?>
<tree string="Product">
    <field name="name"/>
    <field name="categ_id"/>
    <field name="type"/>
    <field name="state"/>
    <field name="uom_id" invisible="1"/>
</tree>

```

4. Form *Product*

```

<?xml version="1.0"?>
<data><xpath expr="//form" position="attributes">
    <attribute name="name">Product Template</attribute>
</xpath>
<field name="active" position="after">
    <field name="ean13" attrs="{ 'invisible':
[( 'product_variant_count', '&gt;', 1)]}" invisible="1"/>
    <field name="default_code" attrs="{ 'invisible':
[( 'product_variant_count', '&gt;', 1)]}" invisible="1"/>
</field>
<xpath expr="//page[@string='Sales']" position="after">
    <page name="variants" string="Variants" invisible="1">
        <div class="oe_right">
            <button class="oe_inline oe_stat_button" string="Variant
Prices" name="116" type="action" icon="fa-strikethrough"/>
            <button class="oe_inline oe_stat_button" name="118"
type="action" icon="fa-sitemap">
                <field string="List of Variants"
name="product_variant_count" widget="statinfo"/>
            </button>
        </div>
    </page>
</xpath>

```

```

    <p class="oe_grey">
      <strong>Warning</strong>: adding or deleting attributes
      will delete and recreate existing variants and lead
      to the loss of their possible customizations.
    </p>
    <field name="attribute_line_ids" widget="one2many_list"
context="{ 'show_attribute': False }">
      <tree string="Variants" editable="bottom">
        <field name="attribute_id"/>
        <field name="value_ids" widget="many2many_tags"
options="{ 'no_create_edit': True }" domain="[( 'attribute_id', '=',
attribute_id)]" context="{ 'default_attribute_id': attribute_id }"/>
      </tree>
    </field>
  </page>
</xpath>
</data>

```

5. Sub Menu Quotation

```

<?xml version="1.0"?>
<tree string="Quotation" fonts="bold:message_unread==True"
colors="grey:state=='cancel';blue:state in ('waiting_date','manual');red:state
in ('invoice_except','shipping_except')">
  <field name="message_unread" invisible="1"/>
  <field name="name" string="Quotation Number"/>
  <field name="date_order"/>
  <field name="partner_id"/>
  <field name="user_id"/>
  <field name="amount_total" sum="Total Tax Included"/>
  <field name="state"/>
</tree>

```

6. Form Quotation

```

<?xml version="1.0"?>
<form string="Sales Order">
  <header>
    <button name="invoice_recreate" states="invoice_except"
string="Recreate Invoice" groups="base.group_user" invisible="1"/>
    <button name="invoice_corrected" states="invoice_except"
string="Ignore Exception" groups="base.group_user" invisible="1"/>
    <button name="action_quotation_send" string="Send by
Email" type="object" states="draft" class="oe_highlight"
groups="base.group_user"/>
    <button name="action_quotation_send" string="Send by
Email" type="object" states="sent,progress,manual"
groups="base.group_user"/>

```

```

        <button name="print_quotation" string="Print"
type="object" states="draft" class="oe_highlight"
groups="base.group_user" invisible="1"/>
        <button name="print_quotation" string="Print"
type="object" states="sent,progress,manual" groups="base.group_user"
invisible="1"/>
        <button name="action_button_confirm" states="draft"
string="Confirm Sale" type="object" groups="base.group_user"/>
        <button name="action_button_confirm" states="sent"
string="Confirm Sale" class="oe_highlight" type="object"
groups="base.group_user"/>
        <button name="action_view_invoice" string="View Invoice"
type="object" class="oe_highlight" attrs="{ 'invisible': [( 'invoice_exists', '=',
False)]}" groups="base.group_user" invisible="1"/>
        <button name="403" string="Create Invoice" type="action"
states="manual" class="oe_highlight" groups="base.group_user"
invisible="1"/>
        <button name="copy_quotation" states="cancel"
string="New Copy of Quotation" type="object"/>
        <button name="cancel" states="draft,sent" string="Cancel
Quotation" groups="base.group_user"/>
        <button name="action_cancel" states="manual,progress"
string="Cancel Order" type="object" groups="base.group_user"/>
        <button name="invoice_cancel" states="invoice_except"
string="Cancel Order" groups="base.group_user" invisible="1"/>
        <field name="state" widget="statusbar"
statusbar_visible="draft,sent,progress,done"
statusbar_colors="{ '&quot;invoice_except&quot;:&quot;red&quot;,&quot;
aiting_date&quot;:&quot;blue&quot; }"/>
    </header>
    <sheet>
    <h1>
        <label string="Quotation " attrs="{ 'invisible': [( 'state', 'not
in', ('draft', 'sent'))]}"/>
        <label string="Sales Order " attrs="{ 'invisible':
[( 'state', 'in', ('draft', 'sent'))]}"/>
        <field name="name" class="oe_inline" readonly="1"/>
    </h1>
    <group>
    <group>
        <field name="partner_id"
on_change="onchange_partner_id(partner_id, context)"
domain="[( 'customer', '=', True)]" context="{ 'search_default_customer': 1,
'show_address': 1}" options="{ '&quot;always_reload&quot;: True }"/>

```

```

        <field name="partner_invoice_id"
groups="sale.group_delivery_invoice_address"
context="{ 'default_type': 'invoice' }" invisible="1"/>
        <field name="partner_shipping_id"
on_change="onchange_delivery_id(company_id, partner_id,
partner_shipping_id, fiscal_position)"
groups="sale.group_delivery_invoice_address"
context="{ 'default_type': 'delivery' }"/>
        <field name="project_id"
context="{ 'partner_id': partner_invoice_id, 'manager_id': user_id,
'default_pricelist_id': pricelist_id, 'default_name': name, 'default_type':
'contract' }" groups="sale.group_analytic_accounting"
domain="[('type', 'in', ['view', 'normal', 'contract'])]"/>
    </group>
    <group>
        <field name="date_order"/>
        <field name="client_order_ref" invisible="1"/>
        <field domain="[('type', '=', 'sale')]" name="pricelist_id"
groups="product.group_sale_pricelist"
on_change="onchange_pricelist_id(pricelist_id, order_line)"/>
        <field name="currency_id" invisible="1"/>
    </group>
</group>
<notebook>
    <page string="Order Lines">
        <field name="order_line">
            <form string="Sales Order Lines">
                <header groups="base.group_user">
                    <button name="404" states="confirmed"
string="Invoice" type="action" icon="terp-document-new"/>
                    <field name="state" widget="statusbar"
statusbar_visible="draft,confirmed,done"
statusbar_colors="{ '&quot;exception&quot;': '&quot;red&quot;', '&quot;cancel
&quot;': '&quot;red&quot; }"/>
                </header>
            </group>
            <group>
                <field name="product_id"
context="{ 'partner_id': parent.partner_id, 'quantity': product_uom_qty,
'pricelist': parent.pricelist_id, 'uom': product_uom, 'company_id':
parent.company_id }" groups="base.group_user"
on_change="product_id_change(parent.pricelist_id, product_id,
product_uom_qty, False, product_uos_qty, False, name, parent.partner_id,
False, True, parent.date_order, False, parent.fiscal_position, False,
context)"/>
                <label for="product_uom_qty"/>

```

```

        <div>
            <field
context="{ 'partner_id':parent.partner_id, 'quantity':product_uom_qty,
'pricelist':parent.pricelist_id, 'uom':product_uom, 'uom_qty_change':True,
'company_id': parent.company_id}" name="product_uom_qty"
class="oe_inline"
on_change="product_id_change(parent.pricelist_id,product_id,product_uom
_qty,product_uom,product_uos_qty,product_uos,name,parent.partner_id,
False, False, parent.date_order, False, parent.fiscal_position, True,
context)"/>
                <field name="product_uom"
groups="product.group_uom" class="oe_inline oe_no_button"
context="{ 'company_id': parent.company_id}"
on_change="onchange_product_uom(parent.pricelist_id,product_id,product
_uom_qty,product_uom,product_uos_qty,product_uos,name,parent.partner_
id, False, False, parent.date_order, parent.fiscal_position, context)"/>
            </div>
            <label for="product_uos_qty"
groups="product.group_uos"/>
            <div groups="product.group_uos">
                <field name="product_uos_qty"
class="oe_inline"/>
                <field name="product_uos"
options="{ '&quot;no_open&quot;': True}" class="oe_inline"/>
            </div>
            <field name="price_unit"/>
            <label for="discount"
groups="sale.group_discount_per_so_line"/>
            <div name="discount"
groups="sale.group_discount_per_so_line">
                <field name="discount" class="oe_inline"/>
            %
        </div>
    </group>
    <group>
        <field name="tax_id"
widget="many2many_tags"
domain="[('parent_id','=',False),('type_tax_use','&lt;&gt;','purchase')]" />
        <field name="th_weight"/>
        <!-- we should put a config wizard for these two
fields -->
        <field name="address_allotment_id"/>
    </group>
</group>
<label for="name"/>
<field name="name"/>

```

```

        <div groups="base.group_no_one">
            <label for="invoice_lines"/>
            <field name="invoice_lines"/>
        </div>
    </form>
    <tree string="Sales Order Lines" editable="bottom">
        <field name="sequence" widget="handle"/>
        <field name="state" invisible="1"/>
        <field name="th_weight" invisible="1"/>
        <field name="product_id"
context="{ 'partner_id':parent.partner_id, 'quantity':product_uom_qty,
'pricelist':parent.pricelist_id, 'uom':product_uom, 'company_id':
parent.company_id}" groups="base.group_user"
on_change="product_id_change(parent.pricelist_id, product_id,
product_uom_qty, False, product_uos_qty, False, name, parent.partner_id,
False, True, parent.date_order, False, parent.fiscal_position, False,
context)"/>
            <field name="name"/>
            <field name="product_uom_qty"
context="{ 'partner_id':parent.partner_id, 'quantity':product_uom_qty,
'pricelist':parent.pricelist_id, 'uom':product_uom, 'company_id':
parent.company_id}" on_change="product_id_change(parent.pricelist_id,
product_id, product_uom_qty, product_uom, product_uos_qty, product_uos,
name, parent.partner_id, False, False, parent.date_order, False,
parent.fiscal_position, True, context)"/>
            <field name="product_uom"
context="{ 'company_id': parent.company_id}"
on_change="onchange_product_uom(parent.pricelist_id, product_id,
product_uom_qty, product_uom, product_uos_qty, product_uos, name,
parent.partner_id, False, False, parent.date_order, parent.fiscal_position,
context)" groups="product.group_uom" options="{&quot;no_open&quot;;
True}"/>
            <field name="product_uos_qty"
groups="product.group_uos" invisible="1"/>
            <field name="product_uos" string="UoS"
groups="product.group_uos" invisible="1"/>
            <field name="price_unit"/>
            <field name="tax_id" widget="many2many_tags"
domain="[('parent_id','=',False),('type_tax_use','&lt;&gt;','purchase')]" />
            <field name="discount"
groups="sale.group_discount_per_so_line"/>
            <field name="price_subtotal"/>
        </tree>
    </field>
    <group class="oe_subtotal_footer oe_right" colspan="2"
name="sale_total">

```

```

        <field name="amount_untaxed" widget="monetary"
options="{ 'currency_field': 'currency_id' }"/>
        <field name="amount_tax" widget="monetary"
options="{ 'currency_field': 'currency_id' }"/>
        <div class="oe_subtotal_footer_separator oe_inline">
            <label for="amount_total"/>
            <button name="button_dummy" states="draft,sent"
string="(update)" type="object" class="oe_edit_only oe_link"/>
        </div>
        <field name="amount_total" nlabel="1"
class="oe_subtotal_footer_separator" widget="monetary"
options="{ 'currency_field': 'currency_id' }"/>
    </group>
    <div class="oe_clear"/>
    <field name="note" class="oe_inline" placeholder="Terms
and conditions..." />
</page>
<page string="Other Information"
groups="base.group_user">
    <group>
        <group name="sales_person"
groups="base.group_user">
            <field name="user_id"
context="{ 'default_groups_ref': ['base.group_user',
'base.group_partner_manager', 'account.group_account_invoice',
'base.group_sale_salesman_all_leads'] }"/>
            <field name="section_id" options="{ 'no_create':
True }" groups="base.group_multi_salesteams"/>
            <field groups="base.group_no_one" name="origin"
invisible="1"/>
        </group>
        <group name="sale_pay">
            <field name="payment_term" options="{ 'no_create':
True }" invisible="1"/>
            <field name="fiscal_position" options="{ 'no_create':
True }" on_change="onchange_fiscal_position(fiscal_position, order_line,
context)" invisible="1"/>
            <field name="company_id" widget="selection"
groups="base.group_multi_company"/>
        </group>
    </group>
    <group>
        <field name="invoiced"/>
        <field name="invoice_exists" invisible="1"/>
    </group>
</group>
</page>

```

```

        </notebook>
    </sheet>
    <div class="oe_chatter">
        <field name="message_follower_ids" widget="mail_followers"
groups="base.group_user"/>
        <field name="message_ids" widget="mail_thread"/>
    </div>
</form>

```

7. Form Send Quotation

```

<?xml version="1.0"?>
<form string="Compose Email">
    <group>
        <!-- truly invisible fields for control and options -->
        <field name="composition_mode" invisible="1"/>
        <field name="model" invisible="1"/>
        <field name="res_id" invisible="1"/>
        <field name="is_log" invisible="1"/>
        <field name="parent_id" invisible="1"/>
        <field name="mail_server_id" invisible="1"/>
        <!-- Various warnings -->
        <field name="use_active_domain" invisible="1"/>
        <div colspan="2" class="oe_form_box_info oe_text_center"
attrs="{ 'invisible': [('use_active_domain', '!=', True)]}">
            <p>
                <strong>All records matching your current search filter
will be mailed,
                    not only the ids selected in the list
view.</strong><br/>
                    If you want to work only with selected ids, please
uncheck the
                        list header checkbox.
            </p>
        </div>
        <!-- visible wizard -->
        <field name="email_from"
attrs="{ 'invisible': [('composition_mode', '!=', 'mass_mail')]}">
            <field name="subject" placeholder="Subject..."
required="True"/>
            <!-- recipients -->
            <label for="partner_ids" string="Recipients"
attrs="{ 'invisible': [('is_log', '=', True)]}" groups="base.group_user"/>
            <div groups="base.group_user" attrs="{ 'invisible': [('is_log',
'=', True)]}">
                <span attrs="{ 'invisible': [('composition_mode', '!=',
'mass_mail')]}">

```

```

        <strong>Email mass mailing</strong> on
        <span attrs="{ 'invisible': [('use_active_domain', '=',
True)]}">the selected records</span>
        <span attrs="{ 'invisible': [('use_active_domain', '=',
False)]}">the current search filter</span>.
    </span>
    <span attrs="{ 'invisible': [('composition_mode', '!=',
'comment')]">Followers of the document and</span>
    <field name="partner_ids"
widget="many2many_tags_email" placeholder="Add contacts to notify..."
context="{ 'force_email': True, 'show_email': True}" attrs="{ 'invisible':
[('composition_mode', '!=', 'comment')]" />
</div>
<!-- mass post -->
<field name="notify" attrs="{ 'invisible': ['!',
('composition_mode', '!=', 'mass_post')]" />
<!-- mass mailing -->
<field name="no_auto_thread"
attrs="{ 'invisible': [('composition_mode', '!=', 'mass_mail')]" />
    <field name="reply_to" placeholder="Email address to
redirect replies..." attrs="{ 'invisible': ['!', ('no_auto_thread', '=', False),
('composition_mode', '!=', 'mass_mail')],
'required': [('no_auto_thread', '=', True), ('composition_mode', '=',
'mass_mail')]" />
</group>
<field name="body" />
<field name="attachment_ids" widget="many2many_binary"
string="Attach a file" />
<footer>
    <button string="Send" name="send_mail" type="object"
class="oe_highlight" />
    or
    <button string="Cancel" class="oe_link" special="cancel" />
</footer>
</form>

```

8. Sub Menu *Sales Order*

```

<?xml version="1.0"?>
<tree string="Sales Orders" fonts="bold:message_unread==True"
colors="grey:state=='cancel';blue:state in ('waiting_date','manual');red:state
in ('invoice_except','shipping_except')">
    <field name="message_unread" invisible="1" />
    <field name="name" string="Order Number" />
    <field name="date_order" />
    <field name="partner_id" />
    <field name="user_id" />

```

```

        <field name="amount_total" sum="Total Tax Included"/>
        <field name="state"/>
    </tree>

```

9. Form Sales Order

```

<?xml version="1.0"?>
<form string="Sales Order">
    <header>
        <button name="invoice_recreate" states="invoice_except"
string="Recreate Invoice" groups="base.group_user" invisible="1"/>
        <button name="invoice_corrected" states="invoice_except"
string="Ignore Exception" groups="base.group_user" invisible="1"/>
        <button name="action_quotation_send" string="Send by
Email" type="object" states="draft" class="oe_highlight"
groups="base.group_user"/>
        <button name="action_quotation_send" string="Send by
Email" type="object" states="sent,progress,manual"
groups="base.group_user"/>
        <button name="print_quotation" string="Print"
type="object" states="draft" class="oe_highlight"
groups="base.group_user" invisible="1"/>
        <button name="print_quotation" string="Print"
type="object" states="sent,progress,manual" groups="base.group_user"
invisible="1"/>
        <button name="action_button_confirm" states="draft"
string="Confirm Sale" type="object" groups="base.group_user"/>
        <button name="action_button_confirm" states="sent"
string="Confirm Sale" class="oe_highlight" type="object"
groups="base.group_user"/>
        <button name="action_view_invoice" string="View Invoice"
type="object" class="oe_highlight" attrs="{ 'invisible': [( 'invoice_exists', '=',
False)]}" groups="base.group_user" invisible="1"/>
        <button name="403" string="Create Invoice" type="action"
states="manual" class="oe_highlight" groups="base.group_user"
invisible="1"/>
        <button name="copy_quotation" states="cancel"
string="New Copy of Quotation" type="object"/>
        <button name="cancel" states="draft,sent" string="Cancel
Quotation" groups="base.group_user"/>
        <button name="action_cancel" states="manual,progress"
string="Cancel Order" type="object" groups="base.group_user"/>
        <button name="invoice_cancel" states="invoice_except"
string="Cancel Order" groups="base.group_user" invisible="1"/>
        <field name="state" widget="statusbar"
statusbar_visible="draft,sent,progress,done"

```

```

statusbar_colors="{&quot;invoice_except&quot;:&quot;red&quot;,&quot;waiting_date&quot;:&quot;blue&quot;}">
</header>
<sheet>
<h1>
<label string="Quotation " attrs="{ 'invisible': [('state','not
in','draft','sent')]}"/>
<label string="Sales Order " attrs="{ 'invisible':
[('state','in','draft','sent')]}"/>
<field name="name" class="oe_inline" readonly="1"/>
</h1>
<group>
<group>
<field name="partner_id"
on_change="onchange_partner_id(partner_id, context)"
domain="[('customer','=',True)]" context="{ 'search_default_customer':1,
'show_address': 1}" options="{&quot;always_reload&quot;: True}">
<field name="partner_invoice_id"
groups="sale.group_delivery_invoice_address"
context="{ 'default_type':'invoice'}" invisible="1"/>
<field name="partner_shipping_id"
on_change="onchange_delivery_id(company_id, partner_id,
partner_shipping_id, fiscal_position)"
groups="sale.group_delivery_invoice_address"
context="{ 'default_type':'delivery'}"/>
<field name="project_id"
context="{ 'partner_id':partner_invoice_id, 'manager_id': user_id,
'default_pricelist_id':pricelist_id, 'default_name':name, 'default_type':
'contract'}" groups="sale.group_analytic_accounting"
domain="[('type','in',['view','normal','contract'])]">
</group>
<group>
<field name="date_order"/>
<field name="client_order_ref" invisible="1"/>
<field domain="[('type','=', 'sale')]" name="pricelist_id"
groups="product.group_sale_pricelist"
on_change="onchange_pricelist_id(pricelist_id,order_line)">
<field name="currency_id" invisible="1"/>
</group>
</group>
<notebook>
<page string="Order Lines">
<field name="order_line">
<form string="Sales Order Lines">
<header groups="base.group_user">

```

```

        <button name="404" states="confirmed"
string="Invoice" type="action" icon="terp-document-new"/>
        <field name="state" widget="statusbar"
statusbar_visible="draft,confirmed,done"
statusbar_colors="{&quot;exception&quot;:&quot;red&quot;,&quot;cancel
&quot;:&quot;red&quot;}" />
        </header>
        <group>
        <group>
            <field name="product_id"
context="{ 'partner_id':parent.partner_id, 'quantity':product_uom_qty,
'pricelist':parent.pricelist_id, 'uom':product_uom, 'company_id':
parent.company_id}" groups="base.group_user"
on_change="product_id_change(parent.pricelist_id, product_id,
product_uom_qty, False, product_uos_qty, False, name, parent.partner_id,
False, True, parent.date_order, False, parent.fiscal_position, False,
context)"/>
            <label for="product_uom_qty"/>
            <div>
                <field
context="{ 'partner_id':parent.partner_id, 'quantity':product_uom_qty,
'pricelist':parent.pricelist_id, 'uom':product_uom, 'uom_qty_change':True,
'company_id': parent.company_id}" name="product_uom_qty"
class="oe_inline"
on_change="product_id_change(parent.pricelist_id,product_id,product_uom
_qty,product_uom,product_uos_qty,product_uos,name,parent.partner_id,
False, False, parent.date_order, False, parent.fiscal_position, True,
context)"/>
                <field name="product_uom"
groups="product.group_uom" class="oe_inline oe_no_button"
context="{ 'company_id': parent.company_id}"
on_change="onchange_product_uom(parent.pricelist_id,product_id,product
_uom_qty,product_uom,product_uos_qty,product_uos,name,parent.partner_
id, False, False, parent.date_order, parent.fiscal_position, context)"/>
            </div>
            <label for="product_uos_qty"
groups="product.group_uos"/>
            <div groups="product.group_uos">
                <field name="product_uos_qty"
class="oe_inline"/>
                <field name="product_uos"
options="{&quot;no_open&quot;: True}" class="oe_inline"/>
            </div>
            <field name="price_unit"/>
            <label for="discount"
groups="sale.group_discount_per_so_line"/>

```

```

        <div name="discount"
groups="sale.group_discount_per_so_line">
        <field name="discount" class="oe_inline"/>
%
    </div>
</group>
<group>
    <field name="tax_id"
widget="many2many_tags"
domain="[('parent_id','=',False),('type_tax_use','&lt;&gt;','purchase')]" />
    <field name="th_weight" />
    <!-- we should put a config wizard for these two
fields -->
        <field name="address_allotment_id" />
    </group>
</group>
<label for="name" />
<field name="name" />
<div groups="base.group_no_one">
    <label for="invoice_lines" />
    <field name="invoice_lines" />
</div>
</form>
<tree string="Sales Order Lines" editable="bottom">
    <field name="sequence" widget="handle" />
    <field name="state" invisible="1" />
    <field name="th_weight" invisible="1" />
    <field name="product_id"
context="{ 'partner_id':parent.partner_id, 'quantity':product_uom_qty,
'pricelist':parent.pricelist_id, 'uom':product_uom, 'company_id':
parent.company_id}" groups="base.group_user"
on_change="product_id_change(parent.pricelist_id, product_id,
product_uom_qty, False, product_uos_qty, False, name, parent.partner_id,
False, True, parent.date_order, False, parent.fiscal_position, False,
context)" />
        <field name="name" />
        <field name="product_uom_qty"
context="{ 'partner_id':parent.partner_id, 'quantity':product_uom_qty,
'pricelist':parent.pricelist_id, 'uom':product_uom, 'company_id':
parent.company_id}" on_change="product_id_change(parent.pricelist_id,
product_id, product_uom_qty, product_uom, product_uos_qty, product_uos,
name, parent.partner_id, False, False, parent.date_order, False,
parent.fiscal_position, True, context)" />
        <field name="product_uom"
context="{ 'company_id': parent.company_id}"
on_change="onchange_product_uom(parent.pricelist_id, product_id,

```

```

product_uom_qty, product_uom, product_uos_qty, product_uos, name,
parent.partner_id, False, False, parent.date_order, parent.fiscal_position,
context)" groups="product.group_uom" options="{&quot;no_open&quot;;
True}"/>
        <field name="product_uos_qty"
groups="product.group_uos" invisible="1"/>
        <field name="product_uos" string="UoS"
groups="product.group_uos" invisible="1"/>
        <field name="price_unit"/>
        <field name="tax_id" widget="many2many_tags"
domain="[('parent_id','=',False),('type_tax_use','&lt;&gt;','purchase')]" />
        <field name="discount"
groups="sale.group_discount_per_so_line"/>
        <field name="price_subtotal"/>
    </tree>
</field>
<group class="oe_subtotal_footer oe_right" colspan="2"
name="sale_total">
    <field name="amount_untaxed" widget="monetary"
options="{ 'currency_field': 'currency_id' }"/>
    <field name="amount_tax" widget="monetary"
options="{ 'currency_field': 'currency_id' }"/>
    <div class="oe_subtotal_footer_separator oe_inline">
        <label for="amount_total"/>
        <button name="button_dummy" states="draft,sent"
string="(update)" type="object" class="oe_edit_only oe_link"/>
    </div>
    <field name="amount_total" nolabel="1"
class="oe_subtotal_footer_separator" widget="monetary"
options="{ 'currency_field': 'currency_id' }"/>
</group>
<div class="oe_clear"/>
    <field name="note" class="oe_inline" placeholder="Terms
and conditions..." />
</page>
<page string="Other Information"
groups="base.group_user">
    <group>
        <group name="sales_person"
groups="base.group_user">
            <field name="user_id"
context="{ 'default_groups_ref': ['base.group_user',
'base.group_partner_manager', 'account.group_account_invoice',
'base.group_sale_salesman_all_leads'] }"/>
            <field name="section_id" options="{ 'no_create':
True}" groups="base.group_multi_saleteams"/>

```

```

        <field groups="base.group_no_one" name="origin"
invisible="1"/>
    </group>
    <group name="sale_pay">
        <field name="payment_term" options="{ 'no_create':
True}" invisible="1"/>
        <field name="fiscal_position" options="{ 'no_create':
True}" on_change="onchange_fiscal_position(fiscal_position, order_line,
context)" invisible="1"/>
        <field name="company_id" widget="selection"
groups="base.group_multi_company"/>
    </group>
    <group>
        <field name="invoiced"/>
        <field name="invoice_exists" invisible="1"/>
    </group>
</group>
</page>
</notebook>
</sheet>
<div class="oe_chatter">
    <field name="message_follower_ids" widget="mail_followers"
groups="base.group_user"/>
    <field name="message_ids" widget="mail_thread"/>
</div>
</form>

```

10. Form Outgoing Shipping

```

<?xml version="1.0"?>
<form string="Transfer">
    <header>
        <button name="action_confirm" states="draft" string="Mark as
Todo" type="object" class="oe_highlight" groups="base.group_user"/>
        <button name="action_assign" states="confirmed"
string="Check Availability" type="object" class="oe_highlight"
groups="base.group_user"/>
        <button name="reserve_pick" string="Recheck Availability"
type="object" class="oe_highlight" groups="base.group_user"
attrs="{ 'invisible': ['|', ('state', 'not in', ('waiting', 'partially_available')),
('pack_operation_exist','=',True)] }"/>
        <button name="force_assign"
states="confirmed,waiting,partially_available" string="Force Availability"
type="object" groups="base.group_user" invisible="1"/>
        <button name="do_enter_transfer_details"
states="assigned,partially_available" string="Transfer"
groups="stock.group_stock_user" type="object" class="oe_highlight"/>
    </header>

```

```

        <button name="do_print_picking" string="Print Picking List"
groups="stock.group_stock_user" type="object" attrs="{ 'invisible': ['|',
('picking_type_code', '=', 'outgoing'), ('state', '!=', 'assigned')]}"/>
        <button name="296" string="Reverse Transfer" states="done"
type="action" groups="base.group_user" invisible="1"/>
        <button name="action_cancel"
states="assigned,confirmed,partially_available,draft,waiting"
string="Cancel Transfer" groups="base.group_user" type="object"/>
        <button name="do_unreserve" string="Unreserve"
groups="base.group_user" type="object" attrs="{ 'invisible':
[('quant_reserved_exist', '=', False)]}"/>
        <button name="bid_received" string="Bid Received" type="object"
states="sent" groups="base.group_user"/>
        <field name="state" widget="statusbar"
statusbar_visible="draft,confirmed,partially_available,assigned,done"
statusbar_colors="{ '&quot;shipping_except&quot;:&quot;red&quot;,&quot;
invoice_except&quot;:&quot;red&quot;,&quot;waiting_date&quot;:&quot;
blue&quot; }"/>
    </header>
    <sheet>
        <div class="oe_right oe_button_box">
            <button name="do_partial_open_barcode"
groups="stock.group_stock_user" type="object"
class="oe_stock_scan_button" attrs="{ 'invisible': ['|',('pack_operation_exist',
'=', True),('state','not in',('assigned', 'partially_available'))]}"></button>
            <button name="open_barcode_interface"
groups="stock.group_stock_user" type="object"
class="oe_stock_scan_button" attrs="{ 'invisible': ['|',('pack_operation_exist',
'=', False),('state','not in',('assigned', 'partially_available'))]}"></button>
        </div>
        <h1>
            <field name="name" class="oe_inline" attrs="{ 'invisible':
[('name','=','/')]}"/>readonly="1"/>
        </h1>
        <group>
            <group>
                <field name="partner_id"/>
                <field name="backorder_id" readonly="1"
attrs="{ 'invisible': [('backorder_id','=',False)]}"/>
            </group>
            <group>
                <field name="date"/>
            </group>
        </group>
    </sheet>

```

```

        <field name="min_date"/>
        <field name="origin" placeholder="e.g. PO0032"
class="oe_inline"/>
        <label for="owner_id"
groups="stock.group_tracking_owner" invisible="1"/>
        <div groups="stock.group_tracking_owner">
            <field name="owner_id"/>
            <button name="action_assign_owner" string="Assign
Owner" type="object" attrs="{ 'invisible': ['|',('pack_operation_exist', '=',
False),('state', 'not in', ('draft','assigned','confirmed'))]}" class="oe_link
oe_edit_only"/>
        </div>
    </group>
</group>
<notebook>
    <page string="Products">
        <separator string="Stock Moves" attrs="{ 'invisible':
[( 'pack_operation_exist', '=', False)]}" />
        <field name="move_lines" context="{ 'address_in_id':
partner_id, 'form_view_ref': 'stock.view_move_picking_form',
'tree_view_ref': 'stock.view_move_picking_tree', 'default_picking_type_id':
picking_type_id, 'default_picking_id': active_id}" />
        <field name="pack_operation_exist" invisible="1"/>
        <field name="note" placeholder="Add an internal note..."
class="oe_inline"/>
    </page>
    <page string="Operations" attrs="{ 'invisible': ['|',
('state', '!=', 'done'), ('pack_operation_ids', '=', [])]}">
        <field name="pack_operation_ids"
context="{ 'default_picking_id': active_id, 'default_location_id': location_id,
'default_location_dest_id': location_dest_id}">
            <tree editable="top">
                <field name="package_id"
groups="stock.group_tracking_lot"/>
                <field name="product_id"
on_change="product_id_change(product_id, product_uom_id,
product_qty)"/>
                <field name="product_uom_id"
groups="product.group_uom"/>
                <field name="lot_id" domain="[('product_id', '=',
product_id)]" context="{ 'product_id': product_id}"
groups="stock.group_production_lot"/>
                <field name="picking_id" invisible="1"/>
                <field name="owner_id"
groups="stock.group_tracking_owner"/>

```

```

        <field name="product_qty" attrs="{ 'required':
[(('product_id', '!=', False))]}"/>
        <field name="location_id" domain="[(('id', 'child_of',
parent.location_id)]"/>
        <field name="location_dest_id" domain="[(('id',
'child_of', parent.location_dest_id)]"/>
        <field name="result_package_id"
groups="stock.group_tracking_lot" context="{ 'location_id':
location_dest_id}"/>
    </tree>
</field>
<p class="oe_grey" groups="stock.group_tracking_lot">
    If there is no product but a source package, this means the
source package was moved entirely. If there is a product and a source
package, the product was taken from the source package.
</p>
</page>
<page string="Additional Info">
    <group string="General Informations">
        <group>
            <field name="move_type"/>
            <field name="picking_type_id"/>
            <field name="picking_type_code" invisible="1"/>
            <field name="quant_reserved_exist" invisible="1"/>
            <field name="location_id" invisible="1"/>
            <field name="location_dest_id" invisible="1"/>
        </group>
        <group>
            <field name="company_id"
groups="base.group_multi_company" widget="selection"/>
            <field name="date_done"
groups="base.group_no_one"/>
            <field name="group_id"/>
            <field name="priority"/>
        </group>
    </group>
</page>
</notebook>
</sheet>
<div class="oe_chatter">
    <field name="message_follower_ids"
widget="mail_followers"/>
    <field name="message_ids" widget="mail_thread"/>
</div>
</form>

```

11. Report *Delivery Order*

```

<?xml version="1.0"?>
<t t-name="stock.report_picking">
  <t t-call="report.html_container">
    <t t-foreach="docs" t-as="o">
      <t t-call="report.external_layout">
        <div class="page">
          <div class="row"><div class="col-xs-4 pull-right">
            <img t-att-
src="" report/barcode/?type=%s&value=%s&width=%s&hei
ght=%s' % ('Code128', o.name, 600, 100)"
style="width:300px;height:50px;"/>
          </div></div>
          <div class="row">
            <div class="col-xs-6">
              <div t-if="o.picking_type_id.code=='incoming' and
o.partner_id">
                <span><strong>Supplier
Address:</strong></span>
              </div>
              <div t-if="o.picking_type_id.code=='internal' and
o.partner_id">
                <span><strong>Warehouse
Address:</strong></span>
              </div>
              <div t-if="o.picking_type_id.code=='outgoing' and
o.partner_id">
                <span><strong>Customer
Address:</strong></span>
              </div>
              <div t-if="o.partner_id" name="partner_header">
                <div t-field="o.partner_id" t-field-
options="{&quot;widget&quot;: &quot;contact&quot;, &quot;fields&quot;:
[&quot;address&quot;, &quot;name&quot;, &quot;phone&quot;,
&quot;fax&quot;], &quot;no_marker&quot;: true}" />
                <p t-if="o.partner_id.vat">VAT: <span t-
field="o.partner_id.vat"/></p>
              </div>
            </div>
            <div class="col-xs-5 col-xs-offset-1">
              <div t-if="o.move_lines and
o.move_lines[0].partner_id and o.move_lines[0].partner_id.id !=
o.partner_id.id">
                <span><strong>Delivery
Address:</strong></span>

```

```

        <div t-field="o.move_lines[0].partner_id" t-field-
options="{&quot;widget&quot;; &quot;contact&quot;;, &quot;fields&quot;;
[&quot;address&quot;;, &quot;name&quot;;, &quot;phone&quot;;,
&quot;fax&quot;;], &quot;no_marker&quot;; true}" />
        </div>
        <div t-if="o.picking_type_id.code != 'internal' and
(not o.move_lines or not o.move_lines[0].partner_id) and
o.picking_type_id.warehouse_id.partner_id">
            <span><strong>Warehouse
Address:</strong></span>
            <div t-
field="o.picking_type_id.warehouse_id.partner_id" t-field-
options="{&quot;widget&quot;; &quot;contact&quot;;, &quot;fields&quot;;
[&quot;address&quot;;, &quot;name&quot;;, &quot;phone&quot;;,
&quot;fax&quot;;], &quot;no_marker&quot;; true}" />
            </div>
        </div>
        </div>
        <h2>
        <span t-field="o.picking_type_id"/>:
        <span t-field="o.name"/>
        </h2>
        <table class="table table-condensed">
        <thead>
        <tr>
            <th><strong>Order (Origin)</strong></th>
            <th><strong>State</strong></th>
            <th><strong>Commitment Date</strong></th>
            <th name="td_sched_date_h"><strong>Scheduled
Date</strong></th>
        </tr>
        </thead>
        <tbody>
        <tr>
            <td>
                <span t-field="o.origin"/>
            </td>
            <td>
                <span t-field="o.state"/>
            </td>
            <td>
                <span t-field="o.date"/>
            </td>
            <td name="td_sched_date">
                <span t-field="o.min_date"/>
            </td>
        </tr>
        </tbody>
        </table>

```

```

        </tr>
    </tbody>
</table>
<br/>
<br/>
<table class="table table-condensed" t-if="not
o.pack_operation_ids">
    <thead>
        <tr>
            <th><strong>Product</strong></th>
            <th class="text-
right"><strong>Quantity</strong></th>
            <t t-if="o.picking_type_id.code !=
'incoming'"><th><strong>Source</strong></th></t>
            <th><strong>Barcode</strong></th>
            <t t-if="o.picking_type_id.code !=
'outgoing'"><th><strong>Destination</strong></th></t>
        </tr>
    </thead>
    <tbody>
        <tr t-foreach="o.move_lines" t-as="move">
            <t t-if="(o.state != 'partially_available') or
(move.reserved_availability > 0)">
                <td><span t-field="move.product_id"/></td>
                <td class="text-right">
                    <t t-if="o.state == 'partially_available'">
                        <span t-field="move.reserved_availability"/>
<span t-field="move.product_id.uom_id" groups="product.group_uom"/>
                    </t>
                    <t t-if="o.state != 'partially_available'">
                        <span t-field="move.product_uom_qty"/>
<span t-field="move.product_uom" groups="product.group_uom"/>
                    </t>
                </td>
                <t t-if="o.picking_type_id.code !=
'incoming'"><td><span t-field="move.location_id"/></td></t>
                <td>
                    <span t-if="move.product_id and
move.product_id.ean13">
                        <img t-att-
src=""/report/barcode/?type=%s&value=%s&width=%s&hei
ght=%s' % ('EAN13', move.product_id.ean13, 600, 100)"
style="width:300px;height:50px"/>
                    </span>
                </td>
            </tr>
        </tbody>
    </table>

```

```

        <t t-if="o.picking_type_id.code !=
'outgoing'"><td><span t-field="move.location_dest_id"/></td></t>
        </tr>
    </tbody>
</table>
<table class="table table-condensed" t-
if="o.pack_operation_ids">
    <thead>
        <tr>
            <th><strong>Product</strong></th>
            <th class="text-
right"><strong>Quantity</strong></th>
            <t t-if="o.picking_type_id.code !=
'incoming'"><th><strong>Source</strong></th></t>
            <th><strong>Barcode</strong></th>
            <t t-if="o.picking_type_id.code !=
'outgoing'"><th><strong>Destination</strong></th></t>
        </tr>
    </thead>
    <tbody>
        <tr t-foreach="o.pack_operation_ids" t-
as="pack_operation">
            <td><span t-field="pack_operation.product_id"/>
                <t t-if="not pack_operation.product_id and
pack_operation.package_id"><span t-
field="pack_operation.package_id"/></t></td>
            <td class="text-right"><span t-
field="pack_operation.product_qty"/> <span t-
field="pack_operation.product_uom_id"
groups="product.group_uom"/></td>
            <t t-if="o.picking_type_id.code !=
'incoming'"><td><span t-field="pack_operation.location_id"/>
                <span t-if="pack_operation.package_id">:</span>
                <span t-field="pack_operation.package_id"/>
                <span t-if="pack_operation.lot_id">:</span>
                <span t-field="pack_operation.lot_id"/>
            </td>
            <td>
                <span t-if="pack_operation.lot_id">
                    <img t-att-
src="" /report/barcode/?type=%s&value=%s&width=%s&hei
ght=%s' % ('Code128', pack_operation.lot_id.name, 600, 100)"
style="width:300px;height:50px"/>
                </span>

```

```

<span t-if="pack_operation.product_id and not
pack_operation.lot_id and pack_operation.product_id.ean13">
    
</span>
<span t-if="pack_operation.package_id and not
pack_operation.product_id">
    
</span>
</td>
<t t-if="o.picking_type_id.code !=
'outgoing'"><td><span t-field="pack_operation.location_dest_id"/>
<span t-
if="pack_operation.result_package_id">:</span>
<span t-
field="pack_operation.result_package_id"/>
</td></t>
</tr>
</tbody>
</table>
<table>
<tr>
<td>Received By</td>
<td>Security</td>
<td>Deliver By</td>
<td>Approved By</td>
</tr>
<tr>
<td style="padding-top:3cm"
width="30%">
(.....)
</td>
<td style="padding-top:3cm"
width="30%">
(.....)
</td>
<td style="padding-top:3cm"
width="30%">
(.....)
</td>
<td style="padding-top:3cm"
width="30%">
(.....)
</td>
</tr>

```

```

                                <span t-field="o.write_uid"/>
                                </td>
                            </tr>
                        </table>
                    </div>
                </t>
            </t>
        </t>
    </t>
</t>

```

12. Detail Transfer

```

<?xml version="1.0"?>
<form string="Transfer details" version="7">
    <field name="picking_source_location_id" invisible="True"/>
    <field name="picking_destination_location_id"
invisible="True"/>
    <group string="Products To Move">
        <div class="oe_grey" groups="stock.group_tracking_lot">
            Setting a product and a source package means that the
product will be taken
            out of the package.
        </div>
    </group>
    <group>
        <field name="item_ids"
context="{ 'default_sourceloc_id':picking_source_location_id,
'default_destinationloc_id':picking_destination_location_id}" nolabel="1">
            <tree string="Inventory Details" editable="bottom">
                <field name="package_id"
groups="stock.group_tracking_lot"/>
                <field name="product_id" required="True"
context="{ 'uom':product_uom_id}"
on_change="product_id_change(product_id,product_uom_id,context)"/>
                <field name="quantity"/>
                <button name="split_quantities" string="Split"
type="object" icon="STOCK_PREFERENCES" attrs="{ 'invisible':
[('quantity', '=', 1)]}"/>
                <field name="product_uom_id"
options="{ '&quot;no_open&quot;: True}" groups="product.group_uom"/>
                <field name="sourceloc_id" domain="[('id', 'child_of',
parent.picking_source_location_id)]"/>
                <field name="destinationloc_id" domain="[('id',
'child_of', parent.picking_destination_location_id)]"/>

```

```

        <field name="result_package_id"
groups="stock.group_tracking_lot" context="{location_id:
destinationloc_id}"/>
        <button name="put_in_pack" string="Pack"
type="object" icon="terp-product" attrs="{invisible: [('result_package_id',
'!=', False)]}" groups="stock.group_tracking_lot"/>
        <field name="lot_id"
groups="stock.group_production_lot" domain="[('product_id','=?',
product_id)]" context="{product_id: product_id}"/>
        </tree>
        </field>
    </group>
    <group string="Packages To Move"
groups="stock.group_tracking_lot">
        <div class="oe_grey">
            The source package will be moved entirely. If you specify
a destination package, the source package will be put in the destination
package.
        </div>
    </group>
    <group groups="stock.group_tracking_lot">
        <field name="packop_ids"
context="{default_sourceloc_id:picking_source_location_id,
'default_destinationloc_id':picking_destination_location_id}" nolabel="1">
            <tree editable="bottom">
                <field name="package_id" required="True"
on_change="source_package_change(package_id)"/>
                <field name="sourceloc_id" domain="[('id', 'child_of',
parent.picking_source_location_id)]"/>
                <field name="destinationloc_id" domain="[('id',
'child_of', parent.picking_destination_location_id)]"/>
                <field name="result_package_id"/>
                <button name="put_in_pack" string="Pack"
type="object" icon="terp-product" attrs="{invisible: [('result_package_id',
'!=', False)]}" />
            </tree>
        </field>
    </group>
    <footer>
        <button name="do_detailed_transfer" string="_Apply"
type="object" class="oe_highlight"/>
        or
        <button string="_Cancel" class="oe_link" special="cancel"/>
    </footer>
</form>

```

13. Sub Menu Received to Customers

```
<?xml version="1.0"?>
<tree colors="blue:state == 'draft';grey:state == 'cancel';red:state not in
('cancel', 'done') and min_date < current_date" string="Picking list">
  <field name="name"/>
  <field name="location_dest_id"/>
  <field name="partner_id"/>
  <field name="date"/>
  <field name="min_date" invisible="1"/>
  <field name="origin"/>
  <field name="group_id" invisible="1"/>
  <field name="backorder_id"/>
  <field name="state"/>
  <field name="priority" invisible="1"/>
  <field name="picking_type_id" invisible="1"/>
</tree>
```

14. Form Received to Customers

```
<?xml version="1.0"?>
<form string="Transfer">
  <header>
    <button name="action_confirm" states="draft" string="Mark as
Todo" type="object" class="oe_highlight" groups="base.group_user"/>
    <button name="action_assign" states="confirmed"
string="Check Availability" type="object" class="oe_highlight"
groups="base.group_user"/>
    <button name="reserve_pick" string="Recheck Availability"
type="object" class="oe_highlight" groups="base.group_user"
attrs="{ 'invisible': ['|', ('state', 'not in', ('waiting', 'partially_available')),
('pack_operation_exist', '=', True)] }"/>
    <button name="force_assign"
states="confirmed,waiting,partially_available" string="Force Availability"
type="object" groups="base.group_user" invisible="1"/>
    <button name="do_enter_transfer_details"
states="assigned,partially_available" string="Transfer"
groups="stock.group_stock_user" type="object" class="oe_highlight"/>
    <button name="do_print_picking" string="Print Picking List"
groups="stock.group_stock_user" type="object" attrs="{ 'invisible': ['|',
('picking_type_code', '=', 'outgoing'), ('state', '!=', 'assigned')] }"/>
    <button name="296" string="Reverse Transfer" states="done"
type="action" groups="base.group_user" invisible="1"/>
    <button name="action_cancel"
states="assigned,confirmed,partially_available,draft,waiting"
string="Cancel Transfer" groups="base.group_user" type="object"/>
```

```

        <button name="do_unreserve" string="Unreserve"
groups="base.group_user" type="object" attrs="{ 'invisible':
[('quant_reserved_exist', '=', False)]}"/>
        <button name="bid_received" string="Bid Received" type="object"
states="sent" groups="base.group_user"/>
        <field name="state" widget="statusbar"
statusbar_visible="draft,confirmed,partially_available,assigned,done"
statusbar_colors="{&quot;shipping_except&quot;:&quot;red&quot;,&quot;
invoice_except&quot;:&quot;red&quot;,&quot;waiting_date&quot;:&quot;
blue&quot;}" />
    </header>
    <sheet>
        <div class="oe_right oe_button_box">
            <button name="do_partial_open_barcode"
groups="stock.group_stock_user" type="object"
class="oe_stock_scan_button" attrs="{ 'invisible': [ '(',('pack_operation_exist',
'=', True),('state','not in',('assigned', 'partially_available'))]} "></button>
            <button name="open_barcode_interface"
groups="stock.group_stock_user" type="object"
class="oe_stock_scan_button" attrs="{ 'invisible': [ '(',('pack_operation_exist',
'=', False),('state','not in',('assigned', 'partially_available'))]} "></button>
        </div>
        <h1>
            <field name="name" class="oe_inline" attrs="{ 'invisible':
[('name','=','/')]}" readonly="1"/>
        </h1>
        <group>
            <group>
                <field name="partner_id"/>
                <field name="backorder_id" readonly="1"
attrs="{ 'invisible': [('backorder_id','=',False)]}" />
            </group>
            <group>
                <field name="date"/>
                <field name="min_date"/>
                <field name="origin" placeholder="e.g. PO0032"
class="oe_inline"/>
                <label for="owner_id"
groups="stock.group_tracking_owner" invisible="1"/>
                <div groups="stock.group_tracking_owner">
                    <field name="owner_id"/>

```

```

        <button name="action_assign_owner" string="Assign
Owner" type="object" attrs="{ 'invisible': ['|',('pack_operation_exist', '=',
False),('state', 'not in', ('draft','assigned','confirmed'))] }" class="oe_link
oe_edit_only"/>
    </div>
</group>
</group>
<notebook>
    <page string="Products">
        <separator string="Stock Moves" attrs="{ 'invisible':
[('pack_operation_exist', '=', False)] }"/>
        <field name="move_lines" context="{ 'address_in_id':
partner_id, 'form_view_ref': 'stock.view_move_picking_form',
'tree_view_ref': 'stock.view_move_picking_tree', 'default_picking_type_id':
picking_type_id, 'default_picking_id': active_id }"/>
        <field name="pack_operation_exist" invisible="1"/>
        <field name="note" placeholder="Add an internal note..."
class="oe_inline"/>
    </page>
    <page string="Operations" attrs="{ 'invisible': ['|',
('state', '!=', 'done'), ('pack_operation_ids', '=', [])] }">
        <field name="pack_operation_ids"
context="{ 'default_picking_id': active_id, 'default_location_id': location_id,
'default_location_dest_id': location_dest_id }">
            <tree editable="top">
                <field name="package_id"
groups="stock.group_tracking_lot"/>
                <field name="product_id"
on_change="product_id_change(product_id, product_uom_id,
product_qty)"/>
                <field name="product_uom_id"
groups="product.group_uom"/>
                <field name="lot_id" domain="[('product_id', '=',
product_id)]" context="{ 'product_id': product_id }"
groups="stock.group_production_lot"/>
                <field name="picking_id" invisible="1"/>
                <field name="owner_id"
groups="stock.group_tracking_owner"/>
                <field name="product_qty" attrs="{ 'required':
[('product_id', '!=', False)] }"/>
                <field name="location_id" domain="[('id', 'child_of',
parent.location_id)]"/>
                <field name="location_dest_id" domain="[('id',
'child_of', parent.location_dest_id)]"/>

```

```

        <field name="result_package_id"
groups="stock.group_tracking_lot" context="{ 'location_id':
location_dest_id}"/>
    </tree>
</field>
<p class="oe_grey" groups="stock.group_tracking_lot">
    If there is no product but a source package, this means the
source package was moved entirely. If there is a product and a source
package, the product was taken from the source package.
</p>
</page>
<page string="Additional Info">
    <group string="General Informations">
        <group>
            <field name="move_type"/>
            <field name="picking_type_id"/>
            <field name="picking_type_code" invisible="1"/>
            <field name="quant_reserved_exist" invisible="1"/>
            <field name="location_id" invisible="1"/>
            <field name="location_dest_id" invisible="1"/>
        </group>
        <group>
            <field name="company_id"
groups="base.group_multi_company" widget="selection"/>
            <field name="date_done"
groups="base.group_no_one"/>
            <field name="group_id"/>
            <field name="priority"/>
        </group>
    </group>
</page>
</notebook>
</sheet>
<div class="oe_chatter">
    <field name="message_follower_ids"
widget="mail_followers"/>
    <field name="message_ids" widget="mail_thread"/>
</div>
</form>

```

15. Sub Menu Manufacturing Order

```

<?xml version="1.0"?>
<tree fonts="bold:message_unread==True" colors="blue:state in
('draft','confirmed');red:date_planned<current_date and state not in
('done','cancel');black:date_planned>current_date;gray:state in
('done','cancel') " string="Manufacturing Orders">

```

```

    <field name="message_unread" invisible="1"/>
    <field name="name"/>
    <field name="date_planned"/>
    <field name="product_id"/>
    <field name="product_qty" sum="Total Qty"/>
    <field name="product_uom" groups="product.group_uom"/>
    <field name="routing_id" groups="mrp.group_mrp_routings"/>
    <field name="hour_total" sum="Total Hours"/>
    <field name="cycle_total" sum="Total Cycles"/>
    <field name="origin"/>
    <field name="state"/>
</tree>

```

16. Form Manufacturing Order

```

<?xml version="1.0"?>
<form string="Manufacturing Orders">
    <header>
        <button name="button_confirm" states="draft"
string="Confirm Production" class="oe_highlight"/>
        <button name="346" states="ready,in_production"
string="Produce" type="action" class="oe_highlight"/>
        <button name="action_assign"
states="confirmed,picking_except" string="Check Availability"
type="object" class="oe_highlight"/>
        <button name="force_production" states="confirmed"
string="Force Reservation" type="object"/>
        <button name="button_produce" states="ready" string="Mark
as Started"/>
        <button name="button_cancel"
states="draft,ready,in_production" string="Cancel Production"/>
        <button name="action_cancel" type="object"
states="confirmed" string="Cancel Production"/>
        <field name="state" widget="statusbar"
statusbar_visible="draft,ready,in_production,done"
statusbar_colors="{&quot;confirmed&quot;;&quot;blue&quot;}" />
    </header>
    <sheet>
        <div class="oe_title">
            <h1>Manufacturing Order <field name="name"
class="oe_inline"/></h1>
        </div>
        <group>
            <group>
                <field name="product_id"
on_change="product_id_change(product_id, product_qty)"
domain="[('bom_ids','!=',False),('bom_ids.type','!=','phantom')]"

```

```

class="oe_inline" context="{&quot;default_type&quot;;
&quot;product&quot;}"/>
    <label for="product_qty"/>
    <div>
        <field name="product_qty" class="oe_inline"
on_change="product_id_change(product_id, product_qty)"/>
        <field name="product_uom"
groups="product.group_uom" class="oe_inline"/>
        <button type="action" name="347" string="Update"
states="confirmed" class="oe_edit_only oe_link"/>
    </div>
    <label for="product_uos_qty"
groups="product.group_uos"/>
    <div groups="product.group_uos">
        <field name="product_uos_qty" class="oe_inline"/>
        <label string="-"
attrs="{ 'invisible': [('product_uos', '=', False)]}" class="oe_inline"/>
        <field name="product_uos" class="oe_inline"/>
    </div>
    <field name="date_planned"/>
</group>
<group>
    <field name="bom_id" domain="[('product_id', '=', product_id),
                                ('product_tmpl_id.product_variant_ids', '=', product_id),
                                ('product_id', '=', False)]" context="{ 'default_product_id': product_id}"
on_change="bom_id_change(bom_id)" required="1"/>
    <field name="routing_id" class="oe_inline"
groups="mrp.group_mrp_routings"/>
    <field name="user_id" context="{ 'default_groups_ref':
['base.group_user', 'base.group_partner_manager',
'mrp.group_mrp_user']}" />
    <field name="origin"/>
</group>
<group groups="stock.group_locations">
    <field name="location_src_id"
domain="[('usage', '=', 'internal')]"
on_change="location_id_change(location_src_id, location_dest_id)"/>
    <field name="location_dest_id"
domain="[('usage', '=', 'internal')]" />
</group>
</group>
<notebook>
    <page string="Consumed Products">
    <group>
        <group string="Products to Consume">

```

```

        <field name="move_lines" nolabel="1"
options="{ 'reload_on_button': true }">
        <tree colors="blue:state == 'draft';black:state in
('ready','assigned','in_production');gray:state in ('cancel','done');red:state in
('confirmed','waiting')" string="Products to Consume">
        <field name="product_id"/>
        <field name="product_uom_qty"
string="Quantity"/>
        <field name="product_uom" string="Unit of
Measure" groups="product.group_uom"/>
        <field name="state" invisible="1"/>
        <button name="350" string="Consume
Products" type="action" icon="gtk-go-forward" context="{ 'consume':
True}" states="assigned"/>
        <button name="294" string="Scrap Products"
type="action" icon="terp-gtk-jump-to-ltr" context="{ 'scrap': True}"
states="draft,waiting,confirmed,assigned"/>
        </tree>
        </field>
        </group>
        <group string="Consumed Products">
        <field name="move_lines2" nolabel="1"
options="{ 'reload_on_button': true }">
        <tree colors="red:scrapped==True;blue:state ==
'draft';black:state in ('confirmed','ready','in_production');gray:state ==
'cancel' " string="Consumed Products" editable="bottom">
        <field name="product_id" readonly="1"/>
        <field name="restrict_lot_id"
context="{ 'product_id': product_id}"
groups="stock.group_production_lot"/>
        <field name="product_uom_qty" readonly="1"/>
        <field name="product_uom" readonly="1"
string="Unit of Measure" groups="product.group_uom"/>
        <field name="state" invisible="1"/>
        <field name="scrapped" invisible="1"/>
        </tree>
        </field>
        </group>
        </group>
</page>
<page string="Finished Products">
        <group>
        <group string="Products to Produce">
        <field name="move_created_ids" nolabel="1"
options="{ 'reload_on_button': true }">
        <tree string="Products to Finish">

```

```

        <field name="product_id"/>
        <field name="product_uom_qty"/>
        <field name="product_uom" string="Unit of
Measure" groups="product.group_uom"/>
        <field name="state" invisible="1"/>
    </tree>
</field>
</group>
<group string="Produced Products">
    <field name="move_created_ids2" nolabel="1"
options="{ 'reload_on_button': true }">
        <tree colors="red:scrapped==True;blue:state ==
'draft';black:state in('confirmed','ready','in_production');gray:state ==
'cancel' " string="Finished Products">
            <field name="product_id" readonly="1"/>
            <field name="product_uom_qty"
readonly="1"/>
            <field name="restrict_lot_id"
groups="stock.group_production_lot"/>
            <field name="product_uom" readonly="1"
string="Unit of Measure" groups="product.group_uom"/>
            <field name="location_dest_id" readonly="1"
string="Destination Loc." widget="selection"
groups="stock.group_locations"/>
            <field name="scrapped" invisible="1"/>
            <field name="state" invisible="1"/>
            <button name="294" string="Scrap Products"
type="action" icon="terp-gtk-jump-to-ltr" states="done,cancel"/>
        </tree>
    </field>
</group>
</group>
</page>
<page string="Work Orders"
groups="mrp.group_mrp_routings">
    <button name="action_compute" states="draft"
string="Compute Data" type="object" icon="terp-stock_format-scientific"/>
    <field name="workcenter_lines"
options="{ 'reload_on_button': true }">
        <form string="Production Work Centers">
            <group col="4">
                <field colspan="4" name="name"/>
                <field name="workcenter_id"/>
                <field name="sequence"/>
                <field name="cycle"/>
                <field name="hour" widget="float_time"/>
            </group>
        </form>
    </field>
</page>

```

```

        </group>
    </form>
    <tree string="Production Work Centers">
        <field name="sequence"/>
        <field name="name"/>
        <field name="workcenter_id"/>
        <field name="cycle"/>
        <field name="hour" widget="float_time"/>
    </tree>
</field>
</page>
<page string="Scheduled Products">
    <button name="action_compute" states="draft"
string="Compute Data" type="object" icon="terp-stock_format-scientific"/>
    <field name="product_lines"/>
</page>
<page string="Extra Information">
    <group>
        <group>
            <group>
                <field name="priority"/>
                <field name="date_start" invisible="1"/>
                <field name="date_finished" invisible="1"/>
            </group>
            <group>
                <field name="company_id"
groups="base.group_multi_company" widget="selection"/>
                <field name="move_prod_id"
groups="stock.group_locations"/>
            </group>
        </group>
    </page>
</notebook>
</sheet>
<div class="oe_chatter">
    <field name="message_follower_ids"
widget="mail_followers"/>
    <field name="message_ids" widget="mail_thread"/>
</div>
</form>

```


LAMPIRAN D

BLACK-BOX TESTING

1. Halaman *Login*

Deskripsi : Melakukan *login* dengan memasukan *email* dan *password*

Penguji : Swandini Indah (1314019)

Test ID	Description	Expected Result	Actual Record	Result
1.1-0.1	Mengklik tombol <i>Login</i> tanpa memasukkan <i>email</i> dan <i>password</i>	Sistem akan menolak akses <i>login</i> dan menampilkan pesan harus memasukan <i>email</i>	Sistem menolak akses <i>login</i> dan menampilkan pesan harus memasukan <i>email</i>	<i>Valid</i>
1.1-0.2	Mengklik tombol <i>Login</i> dengan hanya memasukkan <i>email</i>	Sistem akan menolak akses <i>login</i> dan menampilkan pesan harus memasukan <i>password</i>	Sistem menolak akses <i>login</i> dan menampilkan pesan harus memasukan <i>password</i>	<i>Valid</i>
1.1-0.3	Mengklik tombol <i>Login</i> dengan hanya memasukkan <i>password</i>	Sistem akan menolak akses <i>login</i> dan menampilkan pesan harus memasukan <i>email</i>	Sistem menolak akses <i>login</i> dan menampilkan pesan harus memasukan <i>email</i>	<i>Valid</i>
1.1-0.4	Memasukkan <i>email</i> dengan	Sistem akan menolak akses	Sistem menolak akses <i>login</i> dan	<i>Valid</i>

	benar sedangkan <i>password</i> salah atau sebaliknya, lalu mengklik tombol <i>Login</i>	<i>login</i> dan menampilkan pesan <i>email</i> atau <i>password</i> salah	menampilkan pesan <i>email</i> atau <i>password</i> salah	
1.1-0.5	Memasukkan <i>email</i> dan <i>password</i> dengan benar lalu mengklik tombol <i>Login</i>	Sistem akan menerima akses <i>login</i> dan menampilkan halaman utama	Sistem menerima akses <i>login</i> dan menampilkan halaman utama	<i>Valid</i>

2. Menu *Sales*

Deskripsi : Menampilkan menu *Sales*

Penguji : Swandini Indah (1314019)

<i>Test ID</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
2.1-0.1	Mengklik <i>hyperlink</i> Menu <i>Sales</i>	Sistem akan menampilkan Menu <i>Sales</i>	Sistem akan menampilkan Menu <i>Sales</i>	<i>Valid</i>

3. Menu *Manufacturing*

Deskripsi : Menampilkan menu *manufacturing*

Penguji : Swandini Indah (1314019)

<i>Test ID</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
3.1-0.1	Mengklik <i>hyperlink</i> Menu <i>Manufacturing</i>	Sistem akan menampilkan Menu <i>Manufacturing</i>	Sistem akan menampilkan Menu <i>Manufacturing</i>	<i>Valid</i>

4. Sub Menu *Customer*

Deskripsi : Menampilkan sub menu *customer*

Penguji : Swandini Indah (1314019)

Test ID	Description	Expected Result	Actual Record	Result
4.1-0.1	Mengklik <i>hyperlink</i> Sub Menu <i>Customer</i>	Sistem akan menampilkan Sub Menu <i>Customer</i>	Sistem akan menampilkan Sub Menu <i>Customer</i>	<i>Valid</i>

5. Search Data Customer

Deskripsi : Mencari data *customer*

Penguji : Swandini Indah (1314019)

Test ID	Description	Expected Result	Actual Record	Result
4.2-0.1	Mengklik <i>enter</i> tanpa memasukan nama <i>customer</i> pada <i>search box</i>	Sistem tidak memproses pencarian	Sistem tidak memproses pencarian	<i>Valid</i>
4.2-0.2	Memasukan nama <i>customer</i> pada <i>search box</i> dan tidak mengklik <i>enter</i>	Sistem tidak memproses pencarian	Sistem tidak memproses pencarian	<i>Valid</i>
4.2-0.3	Memasukan nama <i>customer</i> pada <i>search box</i> lalu mengklik <i>enter</i>	Sistem menampilkan data <i>customer</i> yang dicari	Sistem menampilkan data <i>customer</i> yang dicari	<i>Valid</i>

6. List Data Customer

Deskripsi : Menampilkan *list* data *customer*

Penguji : Swandini Indah (1314019)

Test ID	Description	Expected Result	Actual Record	Result
4.3-0.1	Mengklik <i>list</i> data <i>customer</i>	Sistem akan menampilkan <i>form Customer</i>	Sistem akan menampilkan <i>form Customer</i>	<i>Valid</i>

7. *Create Data Customer*

Deskripsi : Menambahkan data *customer*

Penguji : Swandini Indah (1314019)

Test ID	Description	Expected Result	Actual Record	Result
4.4-0.1	Mengklik tombol <i>Create</i>	Sistem akan menampilkan <i>form Tambah</i> Data <i>Customer</i>	Sistem akan menampilkan <i>form Tambah</i> Data <i>Customer</i>	<i>Valid</i>
4.4-0.2	Mengklik tombol <i>Save</i> tanpa mengisi semua kolom	Data tidak dapat disimpan dengan menampilkan pesan <i>field name</i> <i>invalid</i>	Data tidak dapat disimpan dengan menampilkan pesan <i>field name</i> <i>invalid</i>	<i>Valid</i>
4.4-0.3	Mengisi kolom isian lalu mengklik tombol <i>Save</i>	Data dapat disimpan	Data dapat disimpan	<i>Valid</i>

8. *Edit Data Customer*

Deskripsi : Mengubah data *customer*

Penguji : Swandini Indah (1314019)

Test ID	Description	Expected Result	Actual Record	Result
----------------	--------------------	------------------------	----------------------	---------------

4.5-0.1	Mengklik tombol <i>Edit</i> pada salah satu data <i>customer</i>	Data <i>customer</i> dapat diubah sesuai kebutuhan	Data <i>customer</i> dapat diubah sesuai kebutuhan	<i>Valid</i>
4.5-0.2	Mengklik tombol <i>Save</i> dengan kondisi semua kolom tidak diisi	Data tidak dapat diubah	Data tidak dapat diubah	<i>Valid</i>
4.5-0.3	Mengisi kolom isian lalu mengklik tombol <i>Save</i>	Data dapat diubah	Data dapat diubah	<i>Valid</i>

9. *Delete Data Customer*

Deskripsi : Menghapus data *customer*

Penguji : Swandini Indah (1314019)

<i>Test ID</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
4.6-0.1	Mengklik tombol <i>Delete</i> pada data yang dipilih	Data dapat dihapus	Data dapat dihapus	<i>Valid</i>

10. Sub Menu *Product*

Deskripsi : Menampilkan sub menu *product*

Penguji : Swandini Indah (1314019)

<i>Test ID</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
5.1-0.1	Mengklik <i>hyperlink</i> Sub Menu <i>Product</i>	Sistem akan menampilkan	Sistem akan menampilkan Sub Menu <i>Product</i>	<i>Valid</i>

		Sub Menu <i>Product</i>		
--	--	----------------------------	--	--

11. *Search Data Product*

Deskripsi : Mencari data *product*

Penguji : Swandini Indah (1314019)

<i>Test ID</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
5.2-0.1	Mengklik <i>enter</i> tanpa memasukan nama <i>product</i> pada <i>search box</i>	Sistem tidak memproses pencarian	Sistem tidak memproses pencarian	<i>Valid</i>
5.2-0.2	Memasukan nama <i>product</i> pada <i>search box</i> dan tidak mengklik <i>enter</i>	Sistem tidak memproses pencarian	Sistem tidak memproses pencarian	<i>Valid</i>
5.2-0.3	Memasukan nama <i>product</i> pada <i>search box</i> lalu mengklik <i>enter</i>	Sistem menampilkan data <i>product</i> yang dicari	Sistem menampilkan data <i>product</i> yang dicari	<i>Valid</i>

12. *List Data Product*

Deskripsi : Menampilkan *list data product*

Penguji : Swandini Indah (1314019)

<i>Test ID</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
5.3-0.1	Mengklik <i>list data product</i>	Sistem akan menampilkan <i>form product</i>	Sistem akan menampilkan <i>form product</i>	<i>Valid</i>

13. *Create Data Product*

Deskripsi : Menambahkan data *product*

Penguji : Swandini Indah (1314019)

<i>Test ID</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
5.4-0.1	Mengklik tombol <i>Create</i>	Sistem akan menampilkan <i>form</i> Tambah Data <i>product</i>	Sistem akan menampilkan <i>form</i> Tambah Data <i>product</i>	<i>Valid</i>
5.4-0.2	Mengklik tombol <i>Save</i> dengan mengosongkan semua kolom	Data tidak dapat disimpan dengan menampilkan pesan <i>field name invalid</i>	Data tidak dapat disimpan dengan menampilkan pesan <i>field name invalid</i>	<i>Valid</i>
5.4-0.3	Mengisi kolom isian lalu mengklik tombol <i>Save</i>	Data dapat disimpan	Data dapat disimpan	<i>Valid</i>

14. *Edit Data Product*

Deskripsi : Mengubah data *product*

Penguji : Swandini Indah (1314019)

<i>Test ID</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
5.5-0.1	Mengklik tombol <i>Edit</i> pada salah satu data <i>product</i>	Data <i>product</i> dapat diubah sesuai kebutuhan	Data <i>product</i> dapat diubah sesuai kebutuhan	<i>Valid</i>
5.5-0.2	Mengklik tombol <i>Save</i> dengan kondisi	Data tidak dapat diubah	Data tidak dapat diubah	<i>Valid</i>

	semua kolom tidak diisi			
5.5-0.3	Mengisi kolom isian lalu mengklik tombol <i>Save</i>	Data dapat diubah	Data dapat diubah	<i>Valid</i>

15. *Delete Data Product*

Deskripsi : Menghapus data *product*

Penguji : Swandini Indah (1314019)

<i>Test ID</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
5.6-0.1	Mengklik tombol <i>Delete</i> pada data yang dipilih	Data dapat dihapus	Data dapat dihapus	<i>Valid</i>

16. Sub Menu *Quotation*

Deskripsi : Menampilkan sub menu *quotation*

Penguji : Swandini Indah (1314019)

<i>Test ID</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
6.1-0.1	Mengklik hyperlink Menu <i>Quotation</i>	Sistem akan menampilkan Menu <i>Quotation</i>	Sistem akan menampilkan Menu <i>Quotation</i>	<i>Valid</i>

17. Sub Menu *Sales Orders*

Deskripsi : Menampilkan sub menu *sales orders*

Penguji : Swandini Indah (1314019)

<i>Test ID</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
7.1-0.1	Mengklik <i>hyperlink</i> Sub	Sistem akan menampilkan	Sistem akan menampilkan Sub	<i>Valid</i>

	Menu Sales Orders	Sub Menu Sales Orders	Menu Sales Orders	
--	-------------------	-----------------------	-------------------	--

18. Sub Menu *Received to Customers*

Deskripsi : Menampilkan sub menu *received to customers*

Penguji : Swandini Indah (1314019)

Test ID	Description	Expected Result	Actual Record	Result
8.1-0.1	Mengklik <i>hyperlink</i> Sub Menu <i>Received to Customers</i>	Sistem akan menampilkan Sub Menu <i>Received to Customers</i>	Sistem akan menampilkan Sub Menu <i>Received to Customers</i>	<i>Valid</i>

19. Sub Menu *Manufacturing Orders*

Deskripsi : Menampilkan sub menu *manufacturing orders*

Penguji : Swandini Indah (1314019)

Test ID	Description	Expected Result	Actual Record	Result
11. 1-0.1	Mengklik <i>hyperlink</i> Sub Menu <i>Manufacturing Orders</i>	Sistem akan menampilkan Sub Menu <i>Manufacturing Orders</i>	Sistem akan menampilkan Sub Menu <i>Manufacturing Orders</i>	<i>Valid</i>

LAMPIRAN E

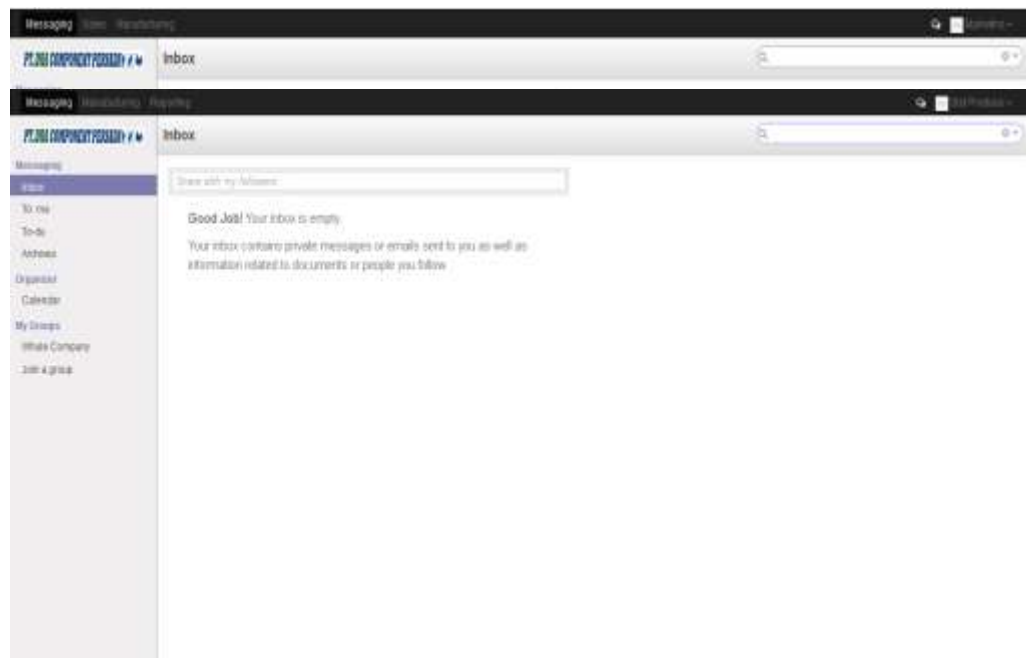
TAMPILAN PROGRAM

1. *Login*



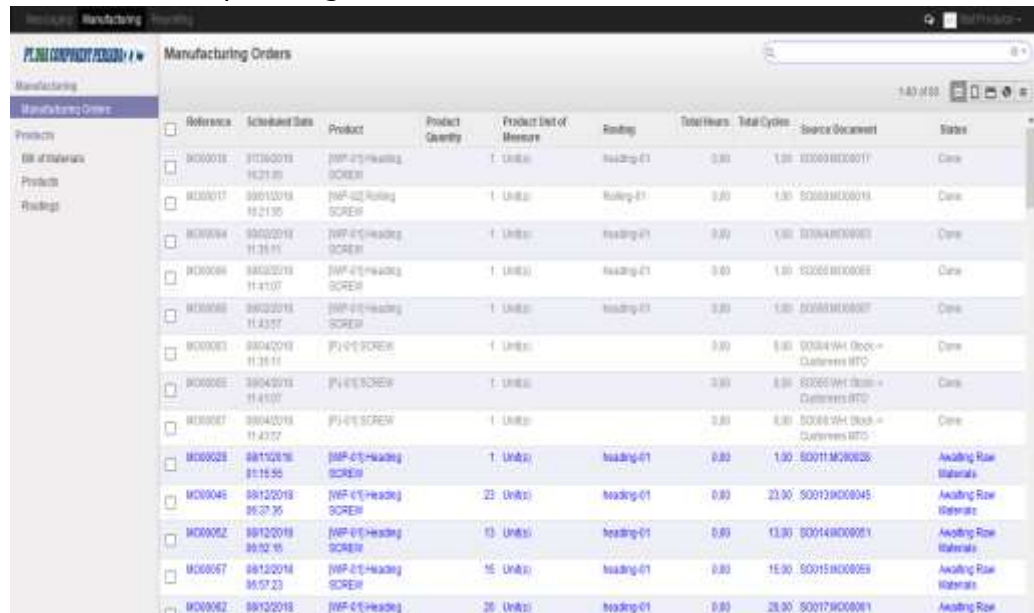
The screenshot shows a login interface for PT. DRA COMPONENT PERSADAY. At the top is the company logo. Below it are three input fields: 'Database' (a dropdown menu), 'Email' (a text field), and 'Password' (a text field). A blue 'Log in' button is positioned below the password field. At the bottom of the form, there is a link that says 'Manage Databases | Powered by Odoo'.

2. Menu Utama Staf *Marketing*



3. Menu Utama Staf *Produksi*

4. Sub Menu *Manufacturing Order*



Reference	Scheduled Date	Product	Product Quantity	Product List of Measure	Routing	Total Hours	Total Cycles	Source Document	Status
MO00015	31/06/2018 16:21:35	WPF-415-Headings SCREW	1 Unit(s)	1 Unit(s)	heading-01	0.00	1.00	SO0001MO00015	Done
MO00017	09/01/2018 16:21:30	WPF-415-Headings SCREW	1 Unit(s)	1 Unit(s)	heading-01	0.00	1.00	SO0001MO00017	Done
MO00018	09/02/2018 11:35:11	WPF-415-Headings SCREW	1 Unit(s)	1 Unit(s)	heading-01	0.00	1.00	SO0001MO00018	Done
MO00019	09/02/2018 11:41:07	WPF-415-Headings SCREW	1 Unit(s)	1 Unit(s)	heading-01	0.00	1.00	SO0001MO00019	Done
MO00020	09/02/2018 11:43:57	WPF-415-Headings SCREW	1 Unit(s)	1 Unit(s)	heading-01	0.00	1.00	SO0001MO00020	Done
MO00021	09/04/2018 11:35:11	PV-415-SCREW	1 Unit(s)	1 Unit(s)		0.00	0.00	SO0001MO00021	Done
MO00022	09/04/2018 11:41:07	PV-415-SCREW	1 Unit(s)	1 Unit(s)		0.00	0.00	SO0001MO00022	Done
MO00023	09/04/2018 11:43:57	PV-415-SCREW	1 Unit(s)	1 Unit(s)		0.00	0.00	SO0001MO00023	Done
MO00024	08/10/2018 05:15:55	WPF-415-Headings SCREW	1 Unit(s)	1 Unit(s)	heading-01	0.00	1.00	SO011MO00024	Awaiting Raw Materials
MO00025	08/12/2018 05:27:35	WPF-415-Headings SCREW	23 Unit(s)	23 Unit(s)	heading-01	0.00	23.00	SO013MO00025	Awaiting Raw Materials
MO00026	08/12/2018 05:32:15	WPF-415-Headings SCREW	13 Unit(s)	13 Unit(s)	heading-01	0.00	13.00	SO014MO00026	Awaiting Raw Materials
MO00027	08/12/2018 05:37:23	WPF-415-Headings SCREW	15 Unit(s)	15 Unit(s)	heading-01	0.00	15.00	SO015MO00027	Awaiting Raw Materials
MO00028	08/12/2018	WPF-415-Headings SCREW	26 Unit(s)	26 Unit(s)	heading-01	0.00	26.00	SO017MO00028	Awaiting Raw Materials

5. Sub Menu *Customer*



Name	Address	Contact Information
PT. HANON ELECTRICAL APPLIANCE S. BOOM SA	13 Jalan Cikarang Selatan, Indonesia	pt.hanon@gmail.com
PT. HANON SA	2 Jalan Cikarang Utara, Indonesia	pt.hanon@gmail.com
PT. HANON USAID TRADING	1 Jalan Klaten, Indonesia	pt.hanon@gmail.com
PT. HANON ELECTRICAL APPLIANCE S. BOOM SA	13 Jalan Cikarang Selatan, Indonesia	pt.hanon@gmail.com
PT. HANON SA	2 Jalan Cikarang Utara, Indonesia	pt.hanon@gmail.com
PT. HANON USAID TRADING	1 Jalan Klaten, Indonesia	pt.hanon@gmail.com
PT. HANON ELECTRICAL APPLIANCE S. BOOM SA	13 Jalan Cikarang Selatan, Indonesia	pt.hanon@gmail.com
PT. HANON SA	2 Jalan Cikarang Utara, Indonesia	pt.hanon@gmail.com
PT. HANON USAID TRADING	1 Jalan Klaten, Indonesia	pt.hanon@gmail.com

6. Form *Customer*

The screenshot shows the 'Customers' section of the PT HAER ELECTRICAL APPLIANCES INDONESIA profile. The page includes a sidebar with navigation links like 'Customers', 'Customers', 'Sales Orders', 'Accounts', 'Received to Customers', 'Products', 'Products', 'Configurations', and 'Units of Measure'. The main content area displays the company name, a logo, and contact information. The contact details are organized into two columns: Address, City, Country, and Telephone on the left; and Phone, Mobile, Fax, and Email on the right. The Address field is populated with 'Sip Industrial Park Plot No 2'. The City field is 'Cikarang', and the Country is 'Indonesia'. The Telephone field is '021-721231', the Mobile field is '08111231234', the Fax field is '021-721231', and the Email field is 'pthaerappliances@gmail.com'. The page also shows a 'Print' button and a 'More' dropdown menu.

7. Sub Menu *Product*

The screenshot shows the NetSuite 'Products' page for '[PJ-01] SCREW'. The interface includes a top navigation bar with 'Sales' and 'Manufacturing' tabs. A sidebar on the left contains a 'PRODUCT COMPANY PERSON' dropdown and a list of navigation links: Sales, Customers, Orders, Sales Orders, Approvals, Related to Customers, Products (highlighted), Configuration, and Units of Measure. The main header area shows 'Products' and a search bar. Below the header, there are 'Edit' and 'Create' buttons. The product details section features an image of a screw, the product name '[PJ-01] SCREW', and three checkboxes: 'Can be Sold', 'Can contribute a landed cost', and 'Can be Purchased'. To the right of these checkboxes is a grid of icons for various business processes: Bill of Materials, Manufacturing, Purchases, Sales, Assembly Rules, Bill of Materials, Items, Receiving Notes, and Receipts. Below the product details, there are tabs for 'Information', 'Purchasing', 'Inventory', 'Sales', and 'Accounting'. The 'Information' tab is active, displaying a table with the following data:

Product Type	Storable Product	Active
Unit of Measure	(100 x)	
Sale Price		

8. Form *Product*

9. Sub Menu *Quotation*

Manufacturing	Sales	Maintenance
PLM COMPANY PERSONA	Quotations	Products
Sales	Create or Import	1-1 of 1
Customers	☐ Quotation Number	Date
Quotations	☐ SOFC	Customer
Sales Orders		PT PAKER ELECTRICAL APPLIANCES INDONESIA
Accounts		Salesperson
Received to Customers		Total
Products		Marketing
Products		\$21.00
Configurations		Quotation Set:
Units of Measure		\$21.00

10. Mengirim Quotation kepada Customer

Quotation SO024

Customer: PT. B3 BODONG
Jl. ...
Kecamatan ...
Kabupaten ...
Provinsi ...

Date: 09/07/2019 17:42:09
Workorder: ...

Product	Description	Quantity	Unit of Measure	Unit Price	Taxes	Subtotal
PU-410 SCREW	PU-410 SCREW	100	Unit(s)	33	PPH 10%	3.300.00
PU-402 BOLT	PU-402 BOLT	100	Unit(s)	23	PPH 10%	2.300.00
PU-402 NUT	PU-402 NUT	100	Unit(s)	25	PPH 10%	2.500.00
PU-404 PH	PU-404 PH	100	Unit(s)	50	PPH 10%	5.000.00

Summary:
 Unbreak Amount: Rp 12.600.00
 Taxes: Rp 1.700.00
Total: Rp 14.300.00

11. Mengirim Quotation kepada Customer (2)

Subject: PT DRA Component Pemasukan Quotation (Ref SO024)

Recipients: Followers of the document and
PT. B3 BODONG <...@...>

Hi PT. B3 BODONG,

Here is your quotation for PT DRA Component Pemasukan.

REFERENCES
 Order number: SO024
 Order total: 14.300.00
 Order date: 09/07/2019 17:42:09 (Asia/Jakarta)
 Your contact: ...

You can access this document and pay online via our Customer Portal.

[View Quotation](#)

Attach a file

Send or **Cancel**

Use template: Sales Order - Send to ...
 Save as new template

12. Konfirmasi Penjualan

Managing Sales Monitoring

PLN COMPANY PERSON /

Quotations / SO024

Send to Email **Create Sales** Cancel Quotation

Draft Quotation **Quotation Sale** Sales Order Done

Quotation SO024

Customer: PT. BB INDONESIA
J. Jachoko T. Bin-ET No.28
Kusadipin Rd. 001000, Hargomel
Cikarang Utara 555
Indonesia

Date: 08/07/2019 17:42:00
Warehouse: Pabrik Gudang

Order Lines Other Information

Product	Description	Quantity	Unit of Measure	Unit Price	Taxes	Subtotal
PJ-10 SCREW	PJ-10 SCREW	100	Unit(s)	83	PPH Kalkulasi 10%	8.300.00
PJ-40 BOLT	PJ-40 BOLT	100	Unit(s)	23	PPH Kalkulasi 10%	2.300.00
PJ-40 NUT	PJ-40 NUT	100	Unit(s)	25	PPH Kalkulasi 10%	2.500.00
PJ-40 PIN	PJ-40 PIN	100	Unit(s)	95	PPH Kalkulasi 10%	9.500.00
Unbreak Amount:						Rp 17.600.00
Taxes:						Rp 1.760.00
Total:						Rp 19.360.00

13. Sub Menu Sales Order

Managing Sales Monitoring

PLN COMPANY PERSON /

Sales Orders

Create Import

1 of 3

Order Number	Date	Customer	Salesperson	Total	Status
☐ SO024	08/07/2019 17:42:00	PT. BB INDONESIA	Marketing	19.360.00	Done
☐ SO021	08/02/2019 03:30:27	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	7.623.00	Sales Order
☐ SO020	08/02/2019 02:52:28	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	2.772.00	Sales Order
☐ SO019	08/01/2019 21:26:28	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	3.465.00	Done
☐ SO018	08/06/2019 15:20:14	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	8.938.00	Done
☐ SO017	08/02/2019 08:12:38	PT. BB INDONESIA	Marketing	1.386.00	Done
☐ SO015	08/02/2019 08:57:23	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	1.038.90	Sales Order
☐ SO014	08/02/2019 08:52:15	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	300.90	Sales Order
☐ SO013	08/02/2019 08:37:38	PT HIER ELECTRICAL APPLIANCES INDONESIA	Marketing	1.592.90	Sales Order
				45.078.30	

14. Form Sales Order

PT. DRA COMPONENT PERSADA

Quotations : SQ024

Customer: PT. BS INDOnesia
Jl. Jababeka X Blok RT No 26
Kawasan Industri Jababeka, Harjamekar
Cikarang Utara 555
Indonesia

Date: 09/09/2018 17:42:08
Warehouse: Finish Good

Product	Description	Quantity	Unit of Measure	Unit Price	Taxes	Subtotal
[PJ-01] SCREW	[PJ-01] SCREW	100	Unit(s)	62	PPn Kawasan 10%	6.300.00
[PJ-02] BOLT	[PJ-02] BOLT	100	Unit(s)	23	PPn Kawasan 10%	2.300.00
[PJ-03] NUT	[PJ-03] NUT	100	Unit(s)	25	PPn Kawasan 10%	2.500.00
[PJ-04] PIN	[PJ-04] PIN	100	Unit(s)	85	PPn Kawasan 10%	8.580.00

Unfired Amount: Rp 17.630.00
Taxes: Rp 1.760.00
Total: Rp 19.390.00

15. Delivery Order Report

PT. DRA COMPONENT PERSADA

Dikun Research & Analyst

PT DRA Component Persada
Jl. Raya Cileungsi Setu RT
001.RW 04 Pasirangin
Bogor, JB
Indonesia

Customer Address:
PT. BS INDONESIA
Jl. Jababeka X Blok RT No 26
Kawasan Industri Jababeka, Harjamekar
Cikarang Utara 555
Indonesia
021-4444254
021-4444254

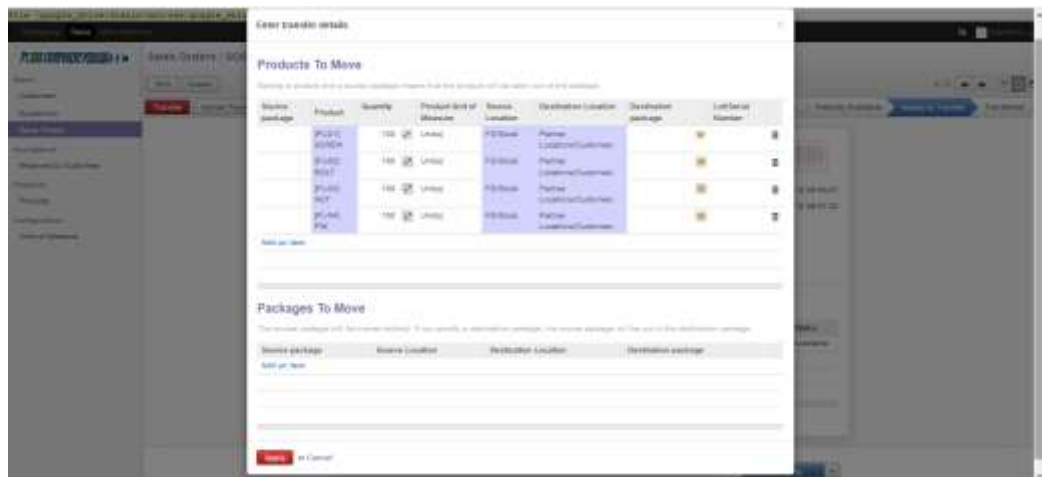
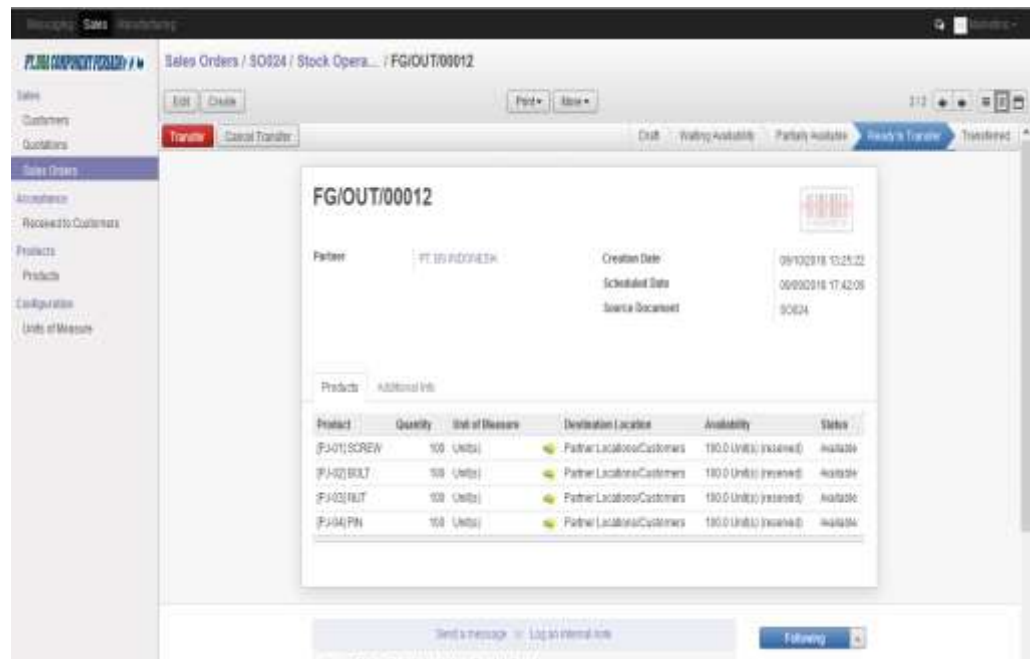
Finish Good: Delivery Orders: FG/OUT/00012

Order (Origin)	State	Commitment Date	Scheduled Date
SQ004	Transferred	09/10/2018 13:25:22	09/09/2018 17:42:08

Product	Quantity	Source	Barcode
[PJ-01] SCREW	100 Unit(s)	FG/Stock	
[PJ-02] BOLT	100 Unit(s)	FG/Stock	
[PJ-03] NUT	100 Unit(s)	FG/Stock	
[PJ-04] PIN	100 Unit(s)	FG/Stock	

Received By: Security Deliver By: Approved By: Marketing

16. Transfer barang pada sistem



17. Detail *transfer* barang pada sistem

18. Konfirmasi Penerimaan Bara

