No. Dok : 7032

D 004. 678

SUMBANGAN ALUMNI

Ls
R.

RANCANG BANGUN SISTEM INFORMASI PERMINTAAN *TOOLING ENGINE* PADA PT MITSUBISHI KRAMAYUDHA MOTORS AND MANUFACTURING BERBASIS WEB MENGGUNAKAN CODEIGNITER DAN MYSQL 5.5.32

TUGAS AKHIR

Untuk Memenuhi Sebagian Syarat Penyelesaian Jenjang Sarjana Terapan Program Studi Sistem Informsi Industri Otomotif Pada Politeknik STMI Jakarta

OLEH

TIYAS PUJI LESTARI

1314092

DATA BUKU PERPUSTAKAAN

Tgl Terima	24/10/21
No Induk Buku	997/sid/SB/TA

POLITEKNIK STMI JAKARTA

KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

JAKARTA

2019

Dipindai dengan CamScanner

POLITEKNIK STMI JAKARTA

KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PENGESAHAN

Judul Tugas Akhir	: **RANCANG BANGUN SISTEM INFORMASI PERMINTAAN TOOLING ENGINE PADA PT MITSUBISHI KRAMAYUDHA MOTORS AND MANUFACTURING BERBASIS WEB DENGAN MENGGUNAKAN CODEIGNITER DAN MYSQL 5.5.32**
Disusun Oleh	:
Nama	: Tiyas Puji Lestari
Nim	: 1314092
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 Rabu tanggal 18 September 2019.

Jakarta, 23 September 2019

Dosen Pembimbing	Ketua Penguji
Drs. Jacob Saragih, MM	**Lucky Heriyanto, ST, MTI**
NIP. 195404281986031002	**NIP. 197908202009011009**
Dosen Penguji	Dosen Penguji
Dedy Trisanto, S.Kom, MMSI.	**Ulil Hamida, ST, MT**
NIP. 197805052005021002	**NIP. 198103272005022001**

Dipindai dengan CamScanner

POLITEKNIK STMI JAKARTA

KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir	: **RANCANG BANGUN SISTEM INFORMASI PERMINTAAN TOOLING ENGINE PADA PT MITSUBISHI KRAMAYUDHA MOTORS AND MANUFACTURING BERBASIS WEB DENGAN MENGGUNAKAN CODEIGNITER DAN MYSQL 5.5.32**
Disusun Oleh	:
Nama	: Tiyas Puji Lestari
Nim	: 1314092
Program Studi	: Sistem Informasi Industri Otomotif
Tanggal Seminar I	: 14 September 2018
Tanggal Seminar II	: 4 September 2019
Tanggal Sidang	: 21 September 2019
Tanggal Lulus	: 21 September 2019

Jakarta, 21 September 2019

Menyetujui

Dosen Pembimbing

Drs. Jacob Saragih, MM

NIP. 195404281986031002

Dipindai dengan CamScanner

Kementerian Perindustrian
REPUBLIK INDONESIA

POLITEKNIK STMI JAKARTA
d.h. SEKOLAH TINGGI MANAJEMEN INDUSTRI
Jl. Letjen Suprapto No. 26 Cempaka Putih, Jakarta 10510
Telp: (021) 42886064 Fax: (021) 42888206
www.stmi.ac.id

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Tiyas Puji Lestari
NIM : 1314092
Judul TA : RANCANG BANGUN SISTEM INFORMASI PERRMINTAAN *TOOLING ENGINE* PADA PT MITSUBISHI KRAMA YUDHA MOTORS AND MANUFACTURING DENGAN BERBASIS WEB MENGGUNAKAN CODEIGNITER DAN MY SQL 5.5.32
Pembimbing : Drs. Jacob Saragih, M.M.

Tanggal	Keterangan	Paraf
25-07-2018	Proposal Tugas Akhir	
27-07-2018	Bab I dan Bab II	
30-07-2018	Bab III	
2-08-2018	Bab IV	
6-08-2018	Acc Bab I, II, III, IV	
7-08-2018	Bab V	
10-08-2018	Revisi Bab V	
13-08-2018	Acc Bab V dan Bab VI	
15-08-2018	Finalisasi Penulisan	
15-08-2018	Acc Seminar	
07-08-2019	Revisi Bab I dan Bab II	
08-08-2019	Revisi Bab III dan Bab IV	
24-08-2019	Acc Bab I, II, III, dan IV	
26-08-2019	Finalisasi Penulisan	
26-08-2019	Acc Seminar Ulang	

Mengetahui,

Ketua Program Studi
Sistem Informasi Industri Otomotif

a.n

Noveriza Yuliasari, MT.
NIP: 197811212009012003

Dosen Pembimbing

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

SAI GLOBAL CERTIFICATION SERVICES Pty.Ltd Registration ISO 9001:2008 No.Reg QEC 264727

Dipindai dengan CamScanner

POLITEKNIK STMI JAKARTA

KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama :Tiyas Puji Lestari

Nim : 1314092

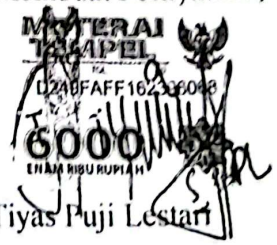
Program Studi : Sistem Informasi Industri Otomotif

Dengan ini menyatakan bahwa karya Tugas Akhir yang saya buat dengan judul: **"RANCANG BANGUN SISTEM INFORMASI PERMINTAAN *TOOLING ENGINE* BERBASIS WEB MENGGUNAKAN CODEIGNITER DAN MY SQL 5.5.32 PADA PT MITSUBISHI KRAMA YUDHA MOTORS AND MANUFACTURING.** Merupakan dibuat dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, survei lapangan, dosen pembimbing dan asisten dosen pembimbing, melalui tanya jawab maupun asistensi serta buku-buku acuan yang tertera dalam referensi pada karya Tugas Akhir ini.

Jika terbukti saya tidak memenuhi apa yang telah dinyatakan di atas, maka karya Tugas Akhir saya dibatalkan.

Jakarta, 16 Agustus 2018

Yang Membuat Pernyataan,



Tiyas Puji Lestari

Dipindai dengan CamScanner

ABSTRAK

PT Mitsubishi Krama Yudha Motors and Manufacturing II merupakan perusahaan yang bergerak di bidang manufaktur yang memproduksi *engine* kendaraan roda empat atau lebih, Pada produksi tersebut terdapat beberapa model kendaraan berupa *truck* Fuso. Dalam menjalankan kegiatannya terutama di kegiatan permintaan *tooling engine* pada *Production Engineering Engine* khususnya bagian *tooling shop* memiliki tahapan yang cukup panjang dan memakan waktu yang lama. Proses permintaan *tooling engine* yang berjalan dilakukan oleh bagian gudang *tooling shop* dan *line poduksi*. Untuk mempermudah pengelolaan data permintaan *tooling engine* dalam basis data sehingga tahapan dalam proses permintaan menjadi lebih singkat dan menghasilkan informasi yang bermanfaat untuk kegiatan permintaan *tooling engine* pada Bagian *Tooling shop*. Metode pengembangan sistem informasi yang digunakan untuk pengumpulan kebutuhan sampai dengan menggunakan sistem adalah *Waterfall*. Analisis dan perancangan berorientasi objek menggunakan pemodelan sistem yaitu *Unified Modelling Language* (UML). Sehingga yang dihasilkan Aplikasi berbasis *web* dengan menggunakan Codeigniter dan MySQL 5.5.32. Dengan adanya aplikasi sistem permintaan *tooling engine* mampu melakukan validasi terhadap *form request material sheet* pada bagian *Line* Produksi ke bagian Gudang *Tooling shop* dan sehingga meminimalisir tingkat kesalahan dalam penginputan dan penyimpanan *document-document* , mempermudah dalam hal pengelolaan data permintaan *tooling engine* karena data pengolalaan disimpan di basis data. Aplikasi ini mampu membuat *view* laporan yang dibutuhkan sehingga informasi dihasilkan menjadi lebih bermanfaat untuk kegiatan permintaan *tooling engine* pada *tooling shop*.

Kata Kunci: Sistem Informasi, Permintaan *Tooling Engine*, Analisis dan Perancangan Berorientasi Objek.

Dipindai dengan CamScanner

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.,

Puji syukur kepada Allah SWT, atas rahmat dan karunia-Nya maka Penulis dapat menyusun dan menyelesaikan Laporan Tugas Akhir dengan judul **"Sistem Infromasi Permintaan Tooling Engine di PT Mitsubishi Kramayudha Motors And Manufacturing Berbasis Web".** Penyusunan laporan tugas akhir ini merupakan salah satu syarat yang harus dipenuhi dalam menempuh jenjang sarjana terapan program studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Penulis menyadari bahwa Laporan Tugas Akhir ini jauh dari kesempurnaan baik dilihat dari segi isinya maupun penyajiannya, walaupun penulis telah berupaya melakukannya sebaik mungkin. Untuk itu penulis mengharapkan kritik dan saran yang bersifat membangun (konstruktif) kepada semua pihak guna perbaikan dimasa yang akan datang.

Dalam penyusunan laporan ini, penulis banyak memperoleh bimbingan dan bantuan dari beberapa pihak. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada :

1. Allah SWT, yang telah memberikan rahmat dan karunia-Nya.
2. Kedua Alm. orang tua dan keluarga saya tercinta terutama alm Ayah saya yang telah membesarkan saya sendiri dari saya kecil hingga sekarang serta Kakak saya juga yang selalu mensuport saya.
3. Sahabat-sahabat saya kuliah dari semester 1 sampai sekarang WaOde dan Hiqlimah.
4. Sahabat-sahabat saya GAA Squad yang telah mensupport saya, menemani saya.
5. Rekan-rekan STMI Program Studi Sistem Informasi kelas SA02 khususnya Della, Ramdhan, yang telah membantu dan mensupport saya selama ini.

Dipindai dengan CamScanner

6. Ibu Noveriza Yuliasari, MT. selaku Ketua Program Studi Sistem Informasi Industri Otomotif.
7. Bapak Jacob Saragih, M.M. selaku Dosen Pembimbing terima kasih saya ucapkan kepada beliau yang selalu memberikan saya bimbingan, saran, motivasi serta teguran membangun agar saya selalu bersemangat menyelesaikan tugas akhir saya dengan baik.
8. Bapak Lucky Heriyanto, MTi dan Ibu Triana Fatmawati, ST,MT. selaku Dosen Penguji Seminar saya yang sudah membantu membimbing memperbaiki tugas akhir saya.
9. Bapak Agus Utomo selaku pembimbing di tempat saya magang PT Mitsubishi Kramayudha Motors and Manufacturing.
10. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.

Semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan laporan ini dengan melimpahkan rahmat dan karunia-Nya. Besar harapan Penulis bahwa laporan ini dapat memberikan manfaat serta pengetahuan bagi pembacanya. Terima kasih.

Jakarta, 25 Agustus 2019

Penulis

Dipindai dengan CamScanner

DAFTAR ISI

Halaman

HALAMAN JUDUL i
LEMBAR PERSETUJUAN DOSEN PEMBIMBING ii
LEMBAR BIMBINGAN TUGAS AKHIR iii
LEMBAR PERNYATAAN KEASLIAN iv
ABSTRAK v
KATA PENGANTAR vi
DAFTAR ISI vii
DAFTAR GAMBAR xi
DAFTAR TABEL xiii
DAFTAR LAMPIRAN xiv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Pokok Permasalahan 2
1.3 Tujuan Penelitian 3
1.4 Batasan Masalah 3
1.5 Manfaat Tugas Akhir 4
1.6 Sistematika Penulisan 4
BAB II LANDASAN TEORI 6
2.1 Konsep Dasar Sistem 6
2.1.1 Karakteristik Sistem 6
2.1.2 Klasifikasi Sistem 8
2.2 Konsep Dasar Informasi 9
2.2.1 Siklus Informasi 9
2.2.2 Nilai Informasi 10
2.2.3 Kualitas Informasi 11
2.3 Konsep Dasar Sistem Informasi 12
2.3.1 Komponen Sistem Informasi 12

Dipindai dengan CamScanner

2.4 Pengertian Sistem Informasi Permintaan 14
2.5 Pengertian *Tooling Engine* 14
2.6 *System Devolopment Life Cycle* 15
2.7 *Waterfall* 16
2.8 *System Requirements* 17
2.8.1 Functional Requiremet 17
2.8.2 Non Functinal Requrement 18
2.9 *Flowmap* 18
2.10 *Unified Modeling Language* 19
2.10.1 *Use Case Diagram* 21
2.10.2 *Use Case Description* 22
2.10.3 *Activity Diagram* 23
2.10.4 *Sequence Diagram* 26
2.10.5 *Class Diagram* 27
2.11 *Deployment Diagram* 28
2.12 *Windows Navigation Diagram* 29
2.13 *Database* 30
2.14 Kamus data 31
2.15 *Entyty Relationship Diagram (ERD)* 31
2.16 *PHP (PHP Hypertext Preprocessor)* 33
2.17 *Web Framework* 33
2.17.1 *Framework Codeigniter* 34
2.18 XAMPP 34
2.19 MySQL 34
2.19.1Keunggulan MySQL 35
2.19.2Tipe Data MySQL 36

BAB III METODOLOGI PENELITIAN 38
3.1 Metedologi Penelitian 38
3.2 Jenis dan Sumber Data 39
3.3 Metode Pengumpulan Data 39

Dipindai dengan CamScanner

3.4 Metode Pengembangan Sistem 40
3.5 Kerangka Penelitian 41
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA 47
4.1 Sejarah Perusahaan 47
4.2 Visi Misi Industri 48
4.3 Kegiatan Usaha Industri 47
4.4 Struktur Organisasi Perusahaan 50
4.4.1 Tugas dan Wewenang setiap Jabatan 51
4.4.2 Tugas Dan Wewenang Divisi PE 56
4.5 Analisis Sistem Berjalan 56
4.6 Analisis Dokumen 57
4.6.1 Dokumen Masukan 57
4.6.2 Dokumen Keluaran 58
4.7 Flowmap yang sedang berjalan 60
BAB V ANALISIS DAN PEMBAHASAN 63
5.1 Analisis Kebutuhan Sistem 63
5.1.1 *Functional Requirement* 63
5.1.2 *Non Functional Requirement* 64
5.1 Prosedur Usulan Sistem Informasi Tooling Engine 64
5.3 Analisis Sistem Usulan 66
5.3.1 *Use Case Diagram* 66
5.3.1.1 *Use Case Description* 67
5.3.2 *Activity Diagram* 70
5.3.3 *Sequence Diagram* 75
5.3.4 *Class Diagram* 81
5.3.5 *Deployment Diagram* 81
5.4 Pemodelan Data Sistem Usulan 82
5.4.1 *Entity Relationship* (ERD) 82
5.4.2 Kamus Data 83
5.5 Perancangan Sistem 86
5.5.1 *Windows Navigation Diagram* (WND) 86

Dipindai dengan CamScanner

5.5.2 Perancangan Antarmuka Pengguna 87
5.6 Implementasi Sistem *Software* dan *Hardware* 91
BAB VI KESIMPULAN DAN SARAN 93
6.1 Kesimpulan 93
6.2 Saran 93
DAFTAR PUSTAKA 94
LAMPIRAN

Dipindai dengan CamScanner

DAFTAR GAMBAR

Halaman

Gambar II.1 Siklus Informasi 9
Gambar II.2 Metode *Waterfall* Menurut Dennis 17
Gambar II.3 Contoh WND 30
Gambar III.1 Metode *Waterfall* 40
Gambar III.2 Kerangka Penelitian 46
Gambar IV.1 Struktur Organisasi PT MKM 49
Gambar IV.2 Struktur Organisasi Divisi PE 51
Gambar IV.4 *Form* RMS 58
Gambar IV.5 *Form* Kartu Stock 59
Gambar IV.6 Flowmap Sistem Berjalan 61
Gambar V.1 *Flowmap* Usulan 65
Gambar V.2 *Use Case* Usulan Sistem Informasi Permintaan Tooling 66
Gambar V.3 *Activity Diagram Login* 70
Gambar V.4 *Activity Diagram* Mengelolah Data *User* 71
Gambar V.5 *Activity Diagram* Mengelolah Data *Tooling* 72
Gambar V.6 *Activity Diagram* Request *Tooling* 73
Gambar V.7 *Activity Diagram* Melihat Request History 73
Gambar V.8 *Activity Diagram* Menyetujui Permintaan *Tooling* 74
Gambar V.9 *Activity Diagram* Mencetak laporan Permintaan *Tooling* 75
Gambar V.10 *Sequence Diagram Login* 76
Gambar V.11 *Sequence Diagram* Mengelola data *User* 77
Gambar V.12 *Sequence Diagram* Mengelolah Data *Tooling* 78
Gambar V.13 *Sequence Diagram Request Tooling* 79
Gambar V.14 *Sequence Diagram* Menyetujui Permintaan *Tooling* 79
Gambar V.15 *Sequence Diagram* Melihat History Permintaan 80
Gambar V.16 *Sequence Diagram* Mencetak Laporam Permintan 80
Gambar V.17 *Class Diagram* Sistem Informasi Usulan 81

Dipindai dengan CamScanner

Gambar V.18 *Deployment Diagram* Sistem Informasi Usulan 82
Gambar V.19 *Entity Relationship Diagram* 83
Gambar V.20 *Windows Navigation Diagram* 87
Gambar V.21 Rancangan *Form* Halaman Utama 87
Gambar V.22 Rancangan *Form* Halaman Data *Tooling* 88
Gambar V.23 Rancangan *Form Request* 88
Gambar V.24 Rancangan *Form* Data Permintaan 89
Gambar V.25 Rancangan Form Update Stock 89
Gambar V.26 Rancangan *Request Story* 90
Gambar V.27 Rancangan *Form* Tambah *User* 90
Gambar V.28 Rancangan *Form* Gudang *User* 91
Gambar V.28 Rancangan *Form Reports* 91

Dipindai dengan CamScanner

DAFTAR TABEL

Halaman

Tabel II.1 Simbol-simbol *Flowmap* 18
Tabel II.2 *Structure Diagram* 20
Tabel II.3 *Behavior Diagram* 21
Tabel II.4 Simbol-simbol *Use Case Diagram* 22
Tabel II.5 Simbol-simbol *Activity Diagram* 24
Tabel II.6 Simbol-simbol *Sequence Diagram* 26
Tabel II.7 Simbol-simbol *Class Diagram* 28
Tabel II.8 Simbol-simbol *Deployment Diagram* 29
Tabel II.9 Simbol-simbol *Entity Relationship Diagram* 32
Tabel II.10 Tipe Data MySQL 37
Tabel V.1 Analisis Kebutuhan Sistem 65
Tabel V.2 *Use Case Description Login* 67
Tabel V.3 *Use Case Description Users* 67
Tabel V.4 *Use Case Description* Mengelola Data *Tooling* 68
Tabel V.5 *Use Case Description Request Tooling* 68
Tabel V.6 *Use Case Description* Melihat *Request Tooling* 68
Tabel V.7 *Use Case Description* Menyetujui Permintaan *Tools* 69
Tabel V.8 *Use Case Description* Mencetak Laporan *Tooling* 69
Tabel V.9 Spesifikasi Tabel *User* 84
Tabel V.10 Spesifikasi Tabel Barang 84
Tabel V.11 Spesifikasi Tabel Permintaan 85
Tabel V.12 Spesifikasi Tabel Permintaan Header 85

Dipindai dengan CamScanner

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

BAB I
PENDAHULUAN

1.1 Latar Belakang

Persaingan bidang industri di era industri ini semakin ketat seiring meningkatnya *Smart Factory* untuk memenuhi kebutuhan masyarakat terhadap sejumlah produk. Hal ini mendorong suatu perusahaan atau industri untuk lebih inovatif, kreatif dan efisien dalam menghasilkan produk yang berkualitas, cepat dan tepat. Untuk menciptakan suatu produk yang berkualitas tinggi, perusahaan atau industri harus memperhatikan proses produksi yang sedang berjalan mulai dari pengadaan bahan baku hingga menjadi produk yang mempunyai kualitas baik serta nilai jual sebelum sampai ke tangan konsumen.

Produksi merupakan kegiatan yang penting dalam suatu perusahaan atau industri, karena memiliki cara, metode dan teknik untuk menghasilkan nilai tambah dari suatu barang dengan mendayagunakan komponen produksi yang ada dengan cara pembuatan sistem informasi sehingga menghasilkan serta meningkatkan kualitas dari suatu produk.

PT Mitsubshi Krama Yudha Motor and Manufacturing II adalah perusahan yang memproduksi *engine* kendaraan roda empat atau lebih. Pada produksi tersebut terdapat beberapa model kendaraan berupa Truk fuso. Dalam memenuhi permintaan konsumen, produksi harus benar-benar ditargetkan, salah satu faktornya adalah sistem informasi permintaan *tools*, khususnya dalam pergerakan produksi harus sesuai jadwal dan kebutuhan ini sangat penting dalam proses produksi. Semua itu di lakukan agar produksi dapat terus berjalan tanpa ada kendala apapun yang dapat menghambat proses produksi.

Proses Permintaan tooling dari line produksi ke bagian *tooling shop* pada PT Mitsubishi Krama Yudha Motors and Manufacturing dalam hal ini masih menggunakan sistem *offline* yaitu dengan mengisikan RMS *(Request Material Sheet)* bagi pemohon *tool* yang diminta. Lalu RMS *(Request Material Sheet)*

Dipindai dengan CamScanner

diberikan kepada bagian gudang, lalu bagian gudang memberikan *tooling* yang tertera pada RMS, selanjutnya bagian gudang menuliskan sisa stok *tooling* yang tersedia pada kartu *stock*. Hal ini merupakan masalah dalam mencatatkan stock yang ada pada gudang karna memakan waktu cukup lama untuk meneliti secara ulang sehingga dalam proses pendistribusian *tooling* ke *line* produksi tersebut kurang optimal.

Sistem Informasi Produksi yang telah diterapkan yaitu sistem informasi yang sebagian besar sudah terintegrasi, Penerapan sistem ini memegang peranan penting dalam proses bisnis pada PT Mitsubishi Krama Yudha Motor and Manufacturing II, akan tetapi sistem terintegrasi ini tidak dapat menghubungkan bagian lini dengan bagian penyedia *tooling* pada bagian produksi.

Untuk mengatasi masalah tersebut diperlukan suatu aplikasi permintaan *tooling*. Untuk itu akan dirancang suatu aplikasi permintan *tooling* dari *line* produksi ke bagian gudang yang berfungsi untuk mempermudah meminta *tools* yang diminta dan mengetahui apakah stok *tools* yang diminta tersedia atau tidak. Selain itu aplikasi yang dirancang tersebut bisa mendukung pembuatan laporan permintaan *tooling* dengan mudah dan cepat, agar meningkatkan proses produksi. Adapun judul Tugas Akhir ini adalah " RANCANG BANGUN SISTEM INFORMASI PERMINTAAN TOOLING ENGINE BERBASIS WEB MENGGUNAKAN CODEIGNITER DAN MY SQL 5.5.32".

1.2 Pokok Permasalahan

Permasalahan yang terjadi pada PT Mitsubishi Kramayudha Motor And Manufacturing adalah sebagai berikut:

1. Proses permintaan *tooling engine* pada bagian permintaan *tooling engine* belum terkomputerisasi pada divisi produksi ke bagian gudang tooling shop sehingga memerlukan waktu yang tidak bisa ditentukan untuk pembuatan dokumen-dokumen pada bagian gudang *tooling shop*.

Dipindai dengan CamScanner

2. Proses pencarian data permintaan *tooling engine* belum ada *database* sebagai penyimpan data, sehingga sulit bagi staff dalam pencarian dokumen dokemen permintaan *tools* karena masih menggunakan cara manual yaitu dengan menggunakan penulisan tangan dan penyimpanan dokumen-dokumen tersebut masih dengan menggunakan map-map yang tersimpan pada rak-rak.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian Tugas Akhir ini adalah sebagai berikut:

1. Untuk merancang dan membangun aplikasi sistem permintaan *tooling engine* untuk membantu dalam pengolahan data permintaan *tooling engine.*
2. Membuat sistem basis data permintaan *tooling engine* sehingga memudahkan dalam penyimpanan data dan pelaporan serta mencegah terjadinya kerusakan dan kehilangan dokumen.

1.4 Batasan Masalah

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

1. Penelitian dilakukan di Divisi PE pada PT Mitsubishi Kramayudha Motors And Manufacturing
2. Penelitian ini dilakukan dalam kurun waktu 1 (satu) bulan, sejak bulan 25 November 2017 s.d. 24 Desember 2017.
3. Penelitian dilakukan hanya sebatas menangani masalah pengolahan data *tooling*, pendataan permintaan bahan *tooling*, pengelolaan data pengeluaran tooling dan laporan ketersediaan *tooling*.
4. Menggunakan Codeigniter dan MySQL 5.5.32 dalam membangun sistem informasi permintaan *tooling.*
5. Menggunakan Aplikasi berbasis *website.*

Dipindai dengan CamScanner

1.5 Manfaat Tugas Akhir

Adapun manfaat yang dapat diambil dari penelitian pada Tugas Akhir ini adalah sebagai berikut:

Hasil penelitian ini agar dapat mengatasi masalah pada bagian *Production Engeneering Engine* seperti permintaan *tooling engine* dari produksi yang langsung ke gudang dan mengatasi masalah pengecekan stock yang dilakukan secara manual dengan masih melihat *stock* di setiap rak.

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

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

BAB II LANDASAN TEORI

Bab ini membahas tentang pengertian sistem, informasi, sistem informasi, sistem informasi persediaan bahan baku, analisis sistem, metode pengembangan sistem, *flowchart, flowmap, Unified Modelling Language* (UML), XAMPP, Codeigniter, dan MySQL 5.5.32 sebagai alat bantu untuk membuat rancangan konseptual, dan teori-teori lain yang berhubungan dengan perancangan sistem.

BAB III METODOLOGI PENELITIAN

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

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini akan menguraikan tentang hasil pengamatan dalam penelitian yang telah dilakukan, seperti fungsi sistem permintaan *tooling,* alur

Dipindai dengan CamScanner

proses sistem permintaan *tooling*, dokumen-dokumen yang terlibat dalam permintaan *toling* dan transaksi-transaksi yang terjadi di sistem permintaan *tooling.*

BAB V ANALISIS DAN PEMBAHASAN

Bab ini berisi analisis rinci dari pengolahan data, yakni mulai dari identifikasi kebutuhan sistem, analisis sistem yang meliputi diagram alir sistem yang berjalan, perancangan sistem dengan menggunakan diagram UML, *Windows Navigation Diagram*, dan Kamus Data, pemodelan data dengan ERD, perancangan basis data, perancangan tampilan layar, perancangan hierarki menu, pembuatan *flowchart* program, dan spesifikasi kebutuhan *software* dan *hardware* untuk mengimplementasi sistem yang dibangun.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan secara keseluruhan yang diperoleh dari hasil penelitian yang dilakukan, serta saran-saran dalam penerapan sistem permintaan *tooling* untuk perusahaan dan pengembangan selanjutnya.

Dipindai dengan CamScanner

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

BAB II
LANDASAN TEORI

2.1 Konsep Dasar Sistem

Sistem berasal dari bahasa Latin "*systēma*" atau bahasa Yunani "*sustēma*" yang berarti suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi, atau energi. Pengertian sistem menurut para ahli dapat didefinisikan sebagai berikut:

1. Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling bergantung satu sama lain, dan terpadu (Sutabri, 2012).
2. Sistem adalah sebuah kelompok dari dua lebih komponen yang saling berhubungan atau subsistem untuk mencapai tujuan bersama (Hall, 2011).

Berdasarkan beberapa pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa sistem adalah kumpulan elemen-elemen yang terhubung, berinteraksi dan saling mempengaruhi untuk mencapai suatu tujuan.

2.1.1 Karakteristik Sistem

Menurut Sutabri (2012) sistem memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah yang mempunyai komponen sistem, batasan sistem, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah, dan sasaran sistem.

1. Komponen Sistem *(Components)*

   Komponen-komponen system tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari system yang menjalankan suatu fungsi tertentu dan mempengaruhi proses system secara keseluruhan.

Dipindai dengan CamScanner

2. Batasan Sistem *(Boundary)*

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem *(Environtment)*

   Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem *(Interface)*

   Media yang menghubungkan sistem dengan subsistem yang lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut.

5. Masukan Sistem *(Input)*

   Energi yang dimasukan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan *(maintenance input)* dan sinyal *(signal input)*.

6. Keluaran Sistem *(Output)*

   Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini dapat menjadi masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi.

7. Pengolah Sistem *(Process)*

   Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

Dipindai dengan CamScanner

8. Sasaran Sistem *(Objective)* dan Tujuan (*Goal*)

   Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.1.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, di antaranya sebagai berikut (Sutabri, 2012):

1. Sistem Abstrak dan Sistem Fisik

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu suatu sistem yang berupa pemikiran tentang hubungan antara manusia dengan Tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem Alami dan Sistem Buatan Manusia

   Sistem alami adalah sistem yang keberadaannya terjadi secara alami atau natural tanpa campuran tangan manusia. Sedangkan sistem buatan manusia adalah sebagai hasil kerja manusia. Contoh sistem alamiah adalah sistem tata surya yang terdiri dari atas sekumpulan planet, gugus bintang dan lainnya.

3. Sistem Deterministik dan Sistem Probabilistik

   Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

4. Sistem Terbuka dan Sistem Tertutup

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis

Dipindai dengan CamScanner

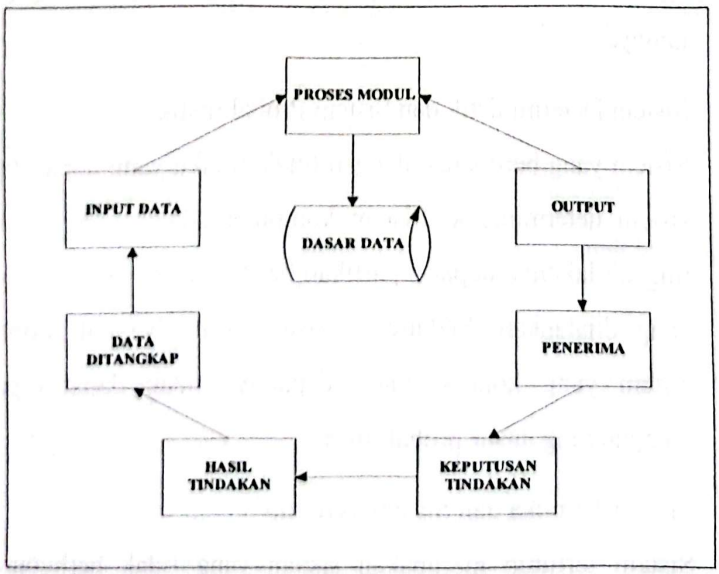
tanpa ada campur tangan dari pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

2.2 Konsep Dasar Informasi

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya. Nilai informasi berhubungan dengan keputusan. Bila tidak ada pilihan atau keputusan maka informasi tidak diperlukan (Sutabri, 2012).

2.2.1 Siklus Informasi

Data diolah melalui suatu model informasi. Si penerima akan menerima informasi tersebut untuk membuat suatu keputusan dan melakukan tindakan yang akan mengakibatkan munculnya sejumlah data lagi. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model, dan seterusnya sehingga membentuk suatu siklus. Siklus inilah yang disebut siklus informasi *(information cycle)* yang dapat dijelaskan pada Gambar II.1 (Sutabri, 2012).



Gambar II.1 Siklus Informasi
Sumber: Sutabri (2012)

Dipindai dengan CamScanner

2.2.2 Nilai informasi

Menurut Sutabri (2012), nilai dari informasi ditentukan dari dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai apabila manfaat yang diperoleh lebih berharga dibandingkan dengan biaya untuk mendapatkannya. Berikut ini adalah nilai informasi berdasarkan atas 10 (sepuluh) sifat, yaitu:

1. Mudah diperoleh

   Sifat ini menunjukan informasi dapat diperoleh dengan mudah dan cepat. Kecepatan memperoleh dapat diukur, misalnya 1 menit versus 24 jam. Akan tetapi beberapa nilainya bagi pemakai informasi sulit mengukurnya.

2. Luas dan lengkap

   Sifat ini menunjukan lengkapnya isi informasi. Hal ini tidak berarti hanya mengenai voluenya, tetapi juga mengenai keluaran informasinya. Sifat ini sangat kabur, karena itu sulit mengukurnya.

3. Ketelitian

   Sifat ini menunjukan minimnya kesalahan dalam informasi. Dalam hubungannya dengan volume data yang besar biasanya terjadi dua jens kesalahan, yakni kesalahan pencatatan dan kesalahan perhitungan.

4. Kecocokan

   Sifat ini menunjukan seberapa baik keluaran informasi dalam hubungan dengan permintaan para pemakai. Isi informasi harus ada hubungannya dengan masalah yang sedang dihadapi. Semua keluaran yang lainnya tidak berguna tetapi mahal mempersiapkannya. Sifat ini sulit mengukurnya.

5. Ketepatan waktu

   Sifat ini menunjukan tak ada keterlambatan jika ada yang sedang ingin mendapatkan informasi. Masukan, pengolahan dan pelaporan keluaran kepada para pemakai biasanya tepat waktu. Dalam beberapa hal, ketepatan waktu dapat diukur. Misalnya berapa banyak penjualan dapat ditamba dengan memberikan tanggapan segera kepada permintaan langgana mengenai tersedianya barang-barang inventaris.

Dipindai dengan CamScanner

6. Kejelasan

   Sifat ini menunjukan keluaran informasi yang bebas dari istilah-istilah yang tidak jelas. Membetulkan laporan dapat memakan biaya yang besar.

7. Keluwesan

   Sifat ini berhubungan dengan dapat disesuaikannya keluaran informasi tidak hanya beberapa keputusan, tetapi juga dengan beberapa pengambil keputusan. Sifat ini sulit diukur, tetapi dalam banyak hal dapat diberikan nilai yang dapat diukur.

8. Dapat dibuktikan

   Sifat ini menunjukan kemampuan beberapa pemakai informasi untuk menguji keluaran informasi dan sampai pada kesimpulan yang sama.

9. Tidak ada prasangka

   Sifat ini berhubungan dengan tidak adanya keinginan untuk mengubah informasi guna mendapatkan kesimpulan yang telah dipertimbangkan sebelumnya.

10. Dapat diukur

    Sifat ini menunjukan hakikat informasi yang dihasilkan dari sistem informasi formal. Meskipun kabar angin, desas-desus, dugaan-dugaan, klenik, dan sebagainya sering dianggap informasi, hal-hal tersebut berada di luar lingkup pembicaraan kita.

Informasi bernilai sempurna apabila pengambil keputusan dapat mengambil keputusan secara optimal dalam setiap hal, dan bukan keputusan yang "rata-rata" akan menjadi optimal dan untuk menghindari kejadian-kejadian yang akan mendatangkan kerugian.

2.2.3 Kualitas informasi

Menurut Sutabri (2012) pada buku Analisis Sistem Informasi, kualitas dari suatu informasi tergantung dari 3 hal, yaitu informasi harus akurat (*accurate*), tepat waktu (*timeliness*) dan relevan (*relevance*).

1. Akurat (*Accurate*)

Dipindai dengan CamScanner

Informasi harus bebas dari kesalahan–kesalahan dan tidak menyesatkan. Akurat juga berarti bahwa informasi harus jelas mencerminkan maksudnya.

2. Tepat Waktu (*Timelines*)

   Informasi yang datang kepada penerima tidak boleh terlambat. Informasi yang sudah usang tidak mempunyai nilai lagi, karena informasi merupakan suatu landasan dalam mengambil sebuah keputusan dimana bila pengambilan keputusan terlambat maka akan berakibat fatal untuk organisasi.

3. Relevan (*Relevance*)

   Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk setiap orang berbeda. Menyampaikan informasi tentang penyebab kerusakan mesin produksi kepada akuntan perusahaan tentunya kurang relevan. Akan lebih relevan bila ditujukan kepada ahli teknik perusahaan. Sebaliknya informasi mengenai harga pokok produksi disampaikan untuk ahli teknik merupakan informasi yang kurang relevan, tetapi akan sangat relevan untuk seorang akuntan perusahaan.

2.3 Konsep Dasar Sistem Informasi

Dalam buku yang ditulis oleh Sutabri (2012), sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sedangkan menurut Laudon dan Laudon (2010), sistem informasi merupakan komponen yang saling bekerja sama untuk mengumpulkan, mengolah, menyimpan dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, pengendalian, analisis masalah dan visualisasi dalam sebuah organisasi.

2.3.1 Komponen Sistem Informasi

Dipindai dengan CamScanner

Sistem informasi terdiri dari komponen-komponen yang disebut blok angunan (*building block*), yang terdiri dari blok *input*, blok *model*, blok *output*, lok teknologi, blok basis data, dan blok kontrol. Semua blok tersebut saling erinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran. Berikut ini adalah penjelasan dari masing-masing blok yang terlibat dalamnya, yaitu (Sutabri, 2012):

1. Blok *input*.

   Input yang dimaksud adalah metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model.

   Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok *output*.

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi.

   Teknologi merupakan *tool box* dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok basis data.

   Basis data merupakan kumpulan data yang saling berkaitan dan berhubungan satu sama lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih

Dipindai dengan CamScanner

lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas

6. Blok kontrol.

   Blok yang bertugas untuk mengendaliakan sistem terhadap ancaman dari dalam maupun luar. Blok kontrol ini didefinisikan bagaimana melindungi data yang ada di *database* dan bila terlanjut terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.4 Pengertian Sistem Informasi Permintaan

Permintaan adalah suatu aktifitas yang dilakukan oleh konsumen sebelum meminta atau mendapatkan barang yang dibutuhkan. Untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem permintaan yang baik. Menurur Kamus Besar Bahasa Indonesia yang dimaksud permintaan adalah "proses, perbuatan, cara meminta atau memesan (tempat,barang,dsb) kepada orang lain.

Berdasarkan pendapat diatas, penulis manyimpulkan bahwa sistem informasi permintaan adalah suatu rancangan sistem informasi yang digunakan untuk membantu kegiatan permintaan atau pemesanan suatu barang disuatu kegiatan produksi dan pergudangan

2.5 Pengertian *Tooling Engine*

Sebuah alat mesin adalah mesin untuk membentuk atau mesin logam atau bahan keras lainnya, biasanyan dengan memotong, mematahkan, menggiling. Tooling engine merupakan suatu alat yang melakukan pemotongan atau shaping. Semua alat mesin memilki beberapa cara untuk menghambat benda kerja dan memberikan gerakan dibimbing dari bagian mesin. Dengan demikian gerakan relative antara benda kerja dan alat potong dikendalikan atau dihambat oleh mesin untuksetidaknya batas tertentu , bukannya seluruhnya "begitu saja" atau "freehand".

Definisi yang tepat dari alat mesin istilah bervariasi antara pengguna, seperti yang dijelaskan di bagian "penamaan dan konsep-konsep kunci". Adalah

Dipindai dengan CamScanner

aman untuk mengatakan bahwa semua peralatan mesin yang "mesin yang membantu orang untuk membuat hal-hal", meskipun tidak semua mesin perlu adalah perlatan mesin.

2.6 *System Development Life Cycle* (SDLC)

Menurut Dennis (2015) *System Development Life Cycle* (SDLC) memiliki empat perangkat fase dasar yaitu *planning*, *analysis*, *design*, dan *implementation*. Beberapa proyek dalam membangun sebuah sistem informasi mungkin dapat berbeda-beda sesuai dengan caranya masing-masing, tetapi hampir semua proyek memiliki elemen dari empat fase tersebut. Masing-masing fase tersebut tersusun dari beberapa langkah yang menghasilkan *deliverable* atau hasil kegiatan seperti beberapa dokumen spesifik dan *file* yang menjelaskan pemahaman tentang proyek.

Dokumen yang dihasilkan dalam tahap analisis, memberikan ide umum dari suatu bagian dari sistem baru. Dokumen dari *deliverable* ini digunakan sebagai *input* pada tahap *design*, yang kemudian disempurnakan untuk menghasilkan dokumen yang menjelaskan secara detil dari sistem yang dibuat. Dokumen ini akan digunakan dalam tahap implementasi untuk menghasilkan sistem yang sebenarnya. Berikut merupakan penjabaran dari setiap fase dalam SDLC:

1. *Planning* (Perencanaan)

   Tahap perencanaan adalah proses dasar yang menjelaskan mengapa sebuah sistem informasi harus dibangun dan menentukan bagaimana sebuah tim proyek akan membangunnya.

2. *Analysis* (Analisis)

   Tahap analisis dalam SDLC menjelaskan siapa yang akan memakai sistem, apa yang sistem kerjakan, serta kapan dan dimana sistem akan digunakan. Selama pada tahap ini, tim proyek menyelidiki sistem yang ada saat ini, mengidentifikasi peluang untuk melakukan perbaikan, dan mengembangkan sistem baru.

3. *Design* (Perancangan)

CS Dipindai dengan CamScanner

Tahap perancangan memutuskan bagaimana sistem akan beroperasi, mulai dari hal yang berkaitan dengan perangkat keras (*hardware*), perangkat lunak (*software*), dan infrastruktur jaringan seperti antarmuka (*interface*), formulir, laporan, serta program spesifik (*database*, dan *file-file* lainnya yang akan dibutuhkan). Meskipun sebagian besar keputusan strategis mengenai sistem dibuat dalam pengembangan konsep sistem selama tahap analisis, langkah-langkah dalam tahap desain menentukan secara tepat bagaimana sistem akan beroperasi.

Kumpulan dari hasil kegiatan ini seperti desain arsitektur, desain antarmuka, spesifikasi *database* dan file, dan perancangan program merupakan spesifikasi sistem yang diserahkan ke tim pemrograman untuk diimplementasi. Pada akhir tahap desain, analisis kelayakan dan rencana proyek diperiksa ulang dan diperbaiki, dan keputusan lain dibuat oleh sponsor proyek dan komite persetujuan mengenai apakah proyek dihentikan atau dilanjutkan.

4. *Implementation* (Implementasi)

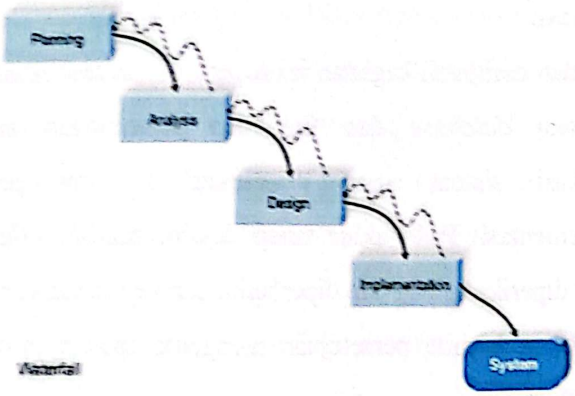
   Tahap akhir dalam SDLC adalah tahap implementasi, dimana sistem benar-benar dibangun. Fase ini sangat penting, karena kebanyakan sistem merupakan bagian terlama dan paling mahal dari proses pengembangan.

2.7 *Waterfall*

Menurut Menurut Dennis et.al (2015), pada penggunaan metode pengembangan *waterfall*, seorang analis dan *user* memproses pengembangan secara bertahap dari satu fase ke fase berikutnya. Setiap fase biasanya berlangsung cukup lama dan setiap fase yang dilewati akan dipresentasikan kepada sponsor untuk mendapatkan persetujuan. Jika sponsor belum menyetujui suatu fase maka pengembangan sistem tidak dapat dilanjutkan ke fase berikutnya. Metodologi ini menyerupai air terjun atau *waterfall* karena bergerak maju dari satu fase ke fase berikutnya secara bertahap seperti cara kerja air terjun, walaupun dalam SDLC memungkinkan untuk kembali ke fase sebelumnya, namun hal ini akan sulit dilakukan dalam Metode *Waterfall*.

Dipindai dengan CamScanner

Keuntungan yang didapat dari pengembangan dengan Metode *Waterfall* yaitu, dapat mengidentifikasi kebutuhan sistem jauh sebelum proses pemrograman berlangsung sehingga meminimalisir perubahan yang dapat terjadi pada kebutuhan sistem saat proyek berjalan. Kelemahan dari Metode *Waterfall* adalah desain harus benar-benar ditentukan sebelum pemrograman dimulai dan lamanya proses yang dibutuhkan untuk menyelesaikan sistem. Berikut penggambaran Metode *Waterfall* menurut Dennis (2010):



Gambar II.2 Metode *Waterfall*
(Sumber: Dennis et.al, 2015)

2.8 *System Requirements*

System requirements atau kebutuhan sistem adalah semua aktivitas yang harus dilakukan atau didukung oleh sistem baru dan batasan-batasan yang harus dicapai sistem baru (Satzinger, 2012).

2.8.1 *Functional Requirement*

Merupakan penjelasan tentang layanan yang perlu disediakan oleh sistem, bagaimana sistem menerima dan mengolah masukan, dan bagaimana sistem mengatasi situasi-situasi tertentu. *Functional Requirement* menggambarkan *system requirement* secara detil seperti *input*, *output*, dan pengecualian yang berlaku.

Contoh *Functional Requirement* pada sistem informasi perpustakaan:

1. Sistem dapat melakukan *input* pendataan buku.
2. Sistem dapat melakukan transaksi peminjaman.

Dipindai dengan CamScanner

3. Sistem dapat melakukan transaksi pengembalian.
4. Sistem dapat menghitung denda apabila peminjam terlambat mengembalikan buku.
5. Sistem dapat menampilkan laporan bulanan peminjaman dan pengembalian buku.

2.8.2 *Non Functional Requirement*

Secara umum berisi batasan-batasan pada pelayanan atau fungsi yang disediakan oleh sistem. Termasuk di dalamnya adalah batasan waktu, batasan proses pembangunan, standar-standar tertentu. Karena berkaitan dengan kebutuhan sistem secara keseluruhan, maka kegagalan memenuhi kebutuhan jenis ini berakibat pada sistem secara keseluruhan.

Contoh *Non Functional Requirement* pada sistem informasi perpustakaan:

1. Sistem dapat dijalankan oleh beberapa *software web browser* di antaranya Internet Explore, Google Chrome, dan Mozilla Firefox.
2. Proses dari pengguna membuka sebuah artikel / dokumen untuk dibaca sampai sistem mengeluarkan / menampilkan artikel tersebut, berlangsung tidak lebih dari 10 detik.
3. Sistem harus dapat memastikan bahwa data yang digunakan dalam sistem harus terlindung dari akses yang tidak berwenang.

2.9 *Flowmap*

Flowmap adalah campuran peta dan *flowchart* yang menunjukan 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). Simbol-simbol yang digunakan dalam *flowmap* dapat dilihat pada tabel II.1 berikut:

Dipindai dengan CamScanner

Tabel II.1 Simbol-simbol *Flowmap*

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

(Sumber Jogiyanto, 2001)

2.10 *Unified Modeling Language* (UML)

Unified Modelling Language (UML) adalah suatu bahasa yang digunakan untuk menemukan, 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.

Dipindai dengan CamScanner

Menurut Dennis et.al (2015), diagram-diagram yang ada dalam UML terbagi menjadi dua bagian utama yaitu, *Structure Diagram* dan *Behavior Diagram*. *Structure Diagram* biasanya digunakan untuk mepresentasikan 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
Class Diagram	Menggambarkan hubungan antara pemodelan-pemodelan *class* di dalam sistem.
Object Diagram	Menggambarkan hubungan antara pemodelan-pemodelan objek di dalam sistem.
Package Diagram	Mengelompokkan elemen UML lainya menjadi satu untuk membentuk tingkat konstruksi yang lebih tinggi.
Composite Diagram	Menggambarkan struktur internal dari suatu *class* dan hubungan di antara bagian-bagian dari suatu *class*.
Deployment Diagram	Menampilkan arsitektur fisik dari suatu sistem. Bisa juga digunakan untuk menunjukkan komponen *software* yang sedang dijadikan arsitektur fisik suatu sistem.

(Sumber: Dennis et.al, 2015)

2. *Behavior Diagram*

   Behavior Diagram dapat dilihat pada Tabel II.3 berikut:

Tabel II.3 *Behavior Diagram*

Nama Diagram	Fungsi

Dipindai dengan CamScanner

Activity Diagram	Menggambarkan proses bisnis masing-masing *class*.
Sequence Diagram	Memodelkan kebiasaan dari objek-objek dalam suatu *use case*.

Tabel II.3 *Behavior Diagram* (Lanjutan)

Nama Diagram	Fungsi
Communication Diagram	Memodelkan kebiasaan dari objek-objek dalam suatu *use case*. Fokus dalam komunikasi antara satu set dari kolaborasi objek dari suatu objek.
Timing Diagram	Menggambarkan interaksi yang terjadi di antara suatu set dari objek-objek dan perubahan keadaan selama perjalanan waktu.
Behavioral State Machine Diagram	Memeriksa kebiasaan dari suatu *class*.
Use Case Diagram	Menangkap kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi di antara sistem dan lingkungannya.
Protocol State Machine Diagram	Menggambarkan ketergantungan di antara perbedaan-perbedaan *interface* dari suatu *class*.

(Sumber: Dennis et.al, 2015)

2.10.1 *Use Case Diagram*

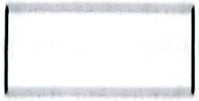
Use Case merupakan alat utama pemodelan yang menjelaskan perilaku (*behavior*) pada suatu sistem (Dennis, 2009). Sebuah *use-case* menggambarkan bagaimana pengguna *(user)* berinteraksi dengan sistem untuk melakukan beberapa aktifitas, seperti membuat pesanan, mencari informasi dan lain-lain. *Use Case* digunakan untuk mengidentifikasi dan menyampaikan kebutuhan (*requirement*) untuk sistem kepada programmer yang akan menulis program di sistem tersebut.

Dipindai dengan CamScanner

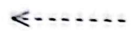
Use Case memiliki dua bentuk, *use case description* dan *use case diagram*. Sebuah *use case description* berisi tentang semua informasi yang dibutuhkan untuk membuat sebuah *use case diagram* yang nantinya akan dibuat dan sebuah use case diagram merupakan gambaran dari pendeskripsian singkat yang telah dijelaskan di *use case description*. *Use Case Diagram* mengilustrasikan fungsi utama pada sistem dan berbagai jenis pengguna (*user*) yang berinteraksi dengan sistem dengan cara sangat sederhana.

Ada beberapa hal yang harus ada dalam sebuah *use case diagram* seperti pada tabel II.4

Tabel II.4 *Use Case* Diagram dan Fungsi-fungsinya

No.	Simbol	Deskripsi
1.	Use Case	▪ Merepresentasikan bagian utama dari sistem secara fungsional. ▪ Diletakan didalam *system boundary*. ▪ Dilabelkan dengan frasa kata kerja deskriptif
2.	Aktor	▪ Seseorang atau sistem yang mendapatkan keuntungan dari sistem. ▪ Dilabelkan dengan peran. ▪ Bisa diasosiasikan dengan aktor lainya berdasarkan spesialisasi. ▪ Diletakan diluar batas sistem.
3.	Boundary 	▪ Nama dari sistem terdapat didalam atau diatas ▪ Merepresentasikan ruang lingkup dari sistem.
4.	Association	▪ Menghubungkan suatu aktor dengan *use case* dengan apa interaksinya.

Dipindai dengan CamScanner

5.	Extend 	– Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu.
6.	Generalization	– Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

(Sumber: Dennis et.al, 2015)

2.10.2 Use Case Description

Use case description merupakan deskripsi yang mencatat mengenai detil pemrosesan dari suatu *use case* (Satzingeretal., 2010). *Use case* memiliki urutan yang lengkap dari tahapan-tahapan untuk menyelesaikan bisnis proses. *Scenario* atau *use case instance* merupakan suatu kumpulan unik dari aktivitas internal di dalam *use case* yang menggambarkan langkah unik sepanjang *use case.*

Menurut Dennis (2010), elemen yang dimiliki use case description adalah sebagai berikut;

1. *Basic Information*
   - Nama : Nama digunakan untuk memjelaskan *use case* dan membedakan *use satu* dengan yang lainnya. Nama yang digunakan harus sederhana namun mampu menjelaskan kondisi dan memungkinkan.
   - *Description* : *Description* berarti uraian singkat untuk menyampaikan tujuan dari *use case.*
   - 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 dilakuan untuk mengeksekusi respon terhadap masukan dan keluaran.
3. *Alternative Courses*

Dipindai dengan CamScanner

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.

2.10.3 Activity Diagram

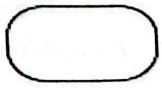
Digunakan untuk model perilaku dalam independen proses bisnis benda. Dalam banyak hal, diagram aktivitas dapat dipandang sebagai diagram aliran data yang canggih yang digunakan dalam hubunganya dengan analisis terstruktur. Namun, tidak seperti aliran data diagram, diagram aktivitas termasuk notasi yang 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:

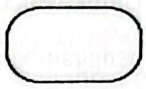
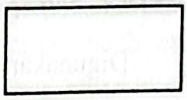
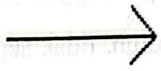
- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
- Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
- Rancangan pengujian dimana setiap 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* seperti pada tabel II.5 dibawah ini:

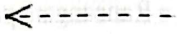
Tabel II.5 Simbol-simbol *Activity Diagram*

No.	Nama elemen	Fungsi	Notasi
1.	*Action*	Untuk menggambarkan perilaku yang sederhana dan bersifat *non- decomposable*.	

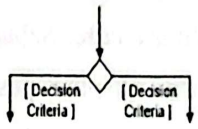
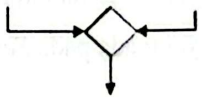
Dipindai dengan CamScanner

2.	*Activity*	Untuk mewakili kumpulan aksi (*action*).	
3.	*Object Node*	Untuk mewakili objek yang terhubung dengan kumpulan *object flow*.	Class Name 
4.	*Control Flow*	Menunjukkan rangkaian pelaksanaan.	

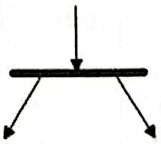
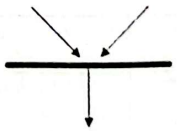
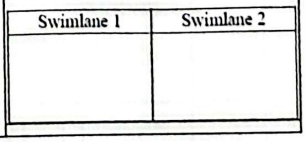
Tabel II.5 Simbol-simbol *Activity Diagram* (Lanjutan)

No.	Nama elemen	Fungsi	Notasi
5.	*Object Flow*	Menunjukkan aliran sebuah objek dari sebuah aktivitas atau aksi ke aktivitas atau aksi lainnya.	
6.	*Initial Node*	Menandakan awal dari kumpulan aksi atau aktivitas.	
7.	*Final-Activity Node*	Untuk menghentikan seluruh *control flows atau object flows* pada sebuah aktivitas (atau aksi).	
8.	*Final-Flow Node*	Untuk menghentikan *control flow* atau *object flow* tertentu.	

Dipindai dengan CamScanner

9.	*Decision Node*	Untuk mewakili suatu kondisi pengujian yang bertujuan untuk memastikan bahwa *control flow* atau *object flow* hanya menuju ke satu arah.	
10	*Merge Node*	Untuk menyatukan kembali *decision path* yang dibuat dengan menggunakan *decision node*.	

Tabel II.5 Simbol-simbol *Activity Diagram* (Lanjutan)

No.	Nama elemen	Fungsi	Notasi
11.	*Fork Node*	Untuk memisahkan perilaku menjadi kumpulan aktivitas yang berjalan secara paralel atau bersamaan.	
12.	*Join Node*	Untuk menyatukan kembali kumpulan aktivitas yang berjalan secara paralel atau bersamaan.	
13.	*Swimlane*	Untuk membagi sebuah *activity diagram* menjadi kolom guna menempatkan aktivitas atau aksi tertentu pada individu atau	

(Sumber: Dennis et.al, 2015)

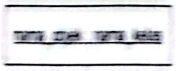
2.10.4 *Sequence Diagram*

Sequence diagram adalah salah satu dari dua jenis diagram interaksi. Mereka menggambarkan benda-benda yang berpartisipasi dalam kasus

Dipindai dengan CamScanner

penggunaan dan pesan yang melewati antara mereka dari waktu ke waktu untuk satu use case. Sebuah diagram *sequence* adalah model dinamis yang menunjukan urutan eksplisit pesan yang lewat di antara objek dalam interaksi 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 (Alan Dennis, 2011). Berikut simbol yang ada pada *Sequence Diagram* pada tabel II.6 berikut:

Tabel II.6 Simbol-simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1.		Aktor	Orang atau sistem yang berasal dari manfaat dan eksternal ke sistem yang berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan
2.		Garis Hidup/*lifeline*	Menyatakan kehidupan suatu objek.
3.		Objek	Berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan yang diempatkan diatas diagram.
4.		*Execution Occurence*	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
5.		*Message*	Pesan yang menggambarkan komunikasi yang terjadi antar objek.

Dipindai dengan CamScanner

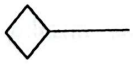
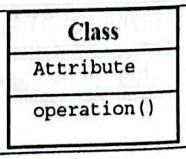
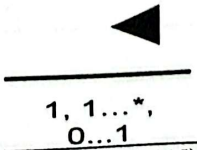
6.		*Message (return)*	Pesan yang dikirim untuk diri sendiri secara langsung.
7.		*Message (return)*	Pesan yang dikirim untuk diri sendiri.

(Sumber: Dennis et.al, 2015)

2.10.5 *Class Diagram*

Sebuah diagram kelas adalah model statis yang menunjukan kelas dan hubungan antar kelas yang tetap konstan dalam sistem dari waktu ke waktu. 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 (metoda fungsi) (Alan Dennis, 2011). Berikut simbol-simbol yang ada pada diagram kelas pada tabel II.7:

Tabel II.7 Simbol-simbol *Class Diagram*

No	Simbol	Nama	Keterangan
1		*Generalization*	Merupakan sebuah *taxonomic relationship* antara *class* yang lebih umum dengan yang lebih khusus.
2		*Aggregation*	Menggambarkan suatu *class* terdiri dari *class* lain atau sustu *class* adalah bagian dari class lain.
3		*Class*	Kelas pada struktur sistem.
4		*Association*	Asosiasi yang menghubungkan *class* dengan *class Multiplycity*.

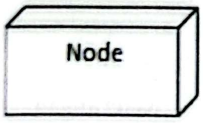
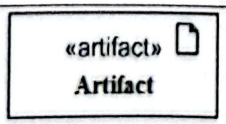
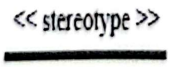
(Sumber: Dennis et.al, 2015)

Dipindai dengan CamScanner

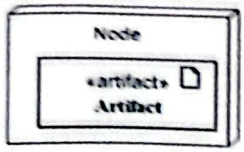
2.11 *Deployment Diagram*

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 menunjukan 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). Simbol-simbol *Deployment Diagram* dapat dilihat pada Tabel II.8 berikut:

Tabel II.8 Simbol-simbol *Deployment Diagram*

No	Simbol	Nama	Deskripsi
1		*Node*	Untuk menggambarkan sumber daya komputasi dalam sebuah sistem (misalnya, computer klien, *server,* jaringan yang terpisah, atau individu perangkat jaringan).
2		*Artifact*	Untuk menggambarkan spesifikasi dari *software* atau *database,* misalnya *file* sumber, tabel *database, executable file.*
3		*Communica-tion Path*	Untuk menggambarkan hubungan antara dua node untuk bertukar pesan.

Dipindai dengan CamScanner

4		*Node with a Deployed Artifact*	Untuk menggambarkan *artifact* yang ditempatkan pada *node* fisik. Mendukung pemodelan distribusi perangkat lunak melalui jaringan.

(Sumber: Dennis et.al, 2015)

2.12 *Window Navigation Diagram* (WND)

Menurut Dennis et.al (2015), desain struktur navigasi mendefinisikan komponen dasar antarmuka dan bagaimana mereka bekerja sama untuk menyediakan fungsionalitas kepada pengguna. *Windows Navigation Diagram* (WND) digunakan untuk menunjukkan bagaimana semua layar, bentuk, dan laporan yang digunakan oleh sistem terkait dan bagaimana pengguna berpindah dari satu ke lainnya. Kebanyakan sistem memiliki beberapa WND's, satu untuk setiap bagian utama dari sistem.

WND sangat mirip dengan *state machine*, karena keduanya memodelkan perubahan status mesin. Keadaannya biasanya membuat model perubahan dari suatu objek, sedangkan keadaan model perubahan lainnya yaitu di antarmuka pengguna. Di WND setiap status antarmuka pengguna direpresentasikan sebagai kotak. Kotak biasanya sesuai dengan komponen antarmuka pengguna, seperti jendela, *form*, tombol, atau laporan.

Struktur navigasi dasar antarmuka mengikuti struktur dasar dari proses bisnis itu sendiri, sebagaimana didefinisikan dalam kasus penggunaan dan model perilaku. Analis memulai dengan kasus-kasus penggunaan esensial dan mengembangkan aliran mendasar dari kontrol sistem ketika bergerak dari objek ke objek. Analis kemudian memeriksa skenario penggunaan untuk melihat seberapa baik dari WND seperti pada gambar II.3

Dipindai dengan CamScanner

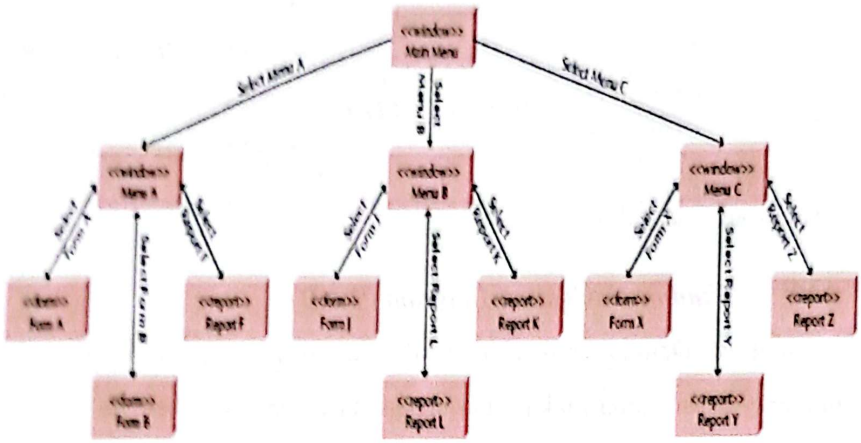


FIGURE 12-7 An Example Window Navigation Diagram

Gambar II.3 Contoh WND
(Sumber: Dennis et.al, 2015)

2.13 Database

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data dapat diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan *Database Management System* (DBMS) (Puspitawati dan Anggadini, 2014).

2.14 Kamus Data

Menurut Jogiyanto (2010) dalam buku Analisis dan Desain, menjelaskan bahwa: "Kamus Data merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. dalam kamus data harus memuat hal-hal berikut:

1. Nama Arus Data. Nama arus data dicatat pada kamus data, sehingga mereka yang membaca DAD memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu.

Dipindai dengan CamScanner

2. Alias. Untuk menyatakan nama lain dari element atau data store yang sebenarnya sama dengan data element atau data *store* yang telah ada.
3. Bentuk Data. Dipergunakan untuk mengelompokan kamus data ke dalam kegunaanya sewaktu perancangan sistem.
4. Arus Data. Menunjukan dari mana data mengalir dan kemana data akan menuju,keterangan arus data ini perluh dicatat di kamus data supaya memudakan mencari arus data ini di DAD.
5. Struktur data. Struktur data menunjukan harus data yang dicatat di kamus data terdiri dari item - item data apa saja.

Berdasarkan definisi di atas maka dapat disimpulkan bahwa kamus data merupakan suatu bantuan yang berguna untuk kebutuhan informasi dari suatu sistem informasi.

2.15 *Entity Relationship Diagram* (ERD)

Terdapat sebuah model perancangan hubungan antar entitas (tabel) dari sebuah basis data (*database*). Istilah untuk frase ini biasa dikenal dengan nama *Entity Relationship Model*. Model hubungan ini seterusnya akan berlanjut menjadi sebuah Diagram Hubungan Antar Entitas yang biasa dikenal dengan nama *Entity Relationship Diagram* (ERD) (Jeffrey A. Hoffer, 2010). Sedangkan menurut Brady dan Loonam (2010), *Entity Relationship Diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh

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:

Dipindai dengan CamScanner

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.

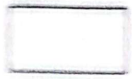
2. *One-to-Many*

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

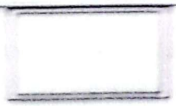
3. *Many-to-Many*

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, seperti pada tabel II.9 berikut.

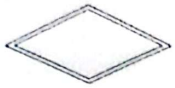
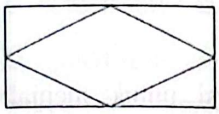
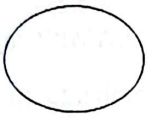
Tabel II.9 Simbol-Simbol *Entity Relationship Diagram*

Simbol	Nama	Deskripsi
	*Strong Entity*	Karakteristik atau tipe entitas yang tidak bergantung pada entitas lain.
	Attribute	Properti atau karakteristik tipe entitas.
	Multivalued Attribute	Karakteristik tipe entitas yang dapat memiliki nilai lebih dari satu.

Tabel II.9 Simbol-Simbol *Entity Relationship Diagram* (Lanjutan)

	Relationship	Hubungan antar entitas.
	*Weak Entity*	Entitas yang bergantung pada entitas lain.

Dipindai dengan CamScanner

	*Identifying Entity*	Entitas yang merupakan induk dari *weak entity.*
	*Associative Entity*	Hanya bisa terdapat antara 2 entitas dan menggambarkan hubungan diantara 2 entitas tersebut.
	*Derived Attribute*	Suatu atribut yang dihasilkan dari atribut yang lain.

(Sumber: Jeffrey A. Hoffer, 2010)

2.16 PHP (PHP Hypertext Preprocessor)

PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web.* Pada saat dipanggil *web browser,* program yang ditulis dengan PHP akan di-*parsing* di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*. Oleh sebab itu, kode PHP tidak akan terlihat pada saat *user* memilih perintah "*View Source"* pada *web browser* yang mereka gunakan. Meskipun PHP 5 dapat digunakan untuk membuat aplikasi CLI (*Command Line Interface)* dan juga aplikasi desktop, namun pada umumnya menggunakan php untuk tujuan pembuatan aplikasi *web.*

2.17 *Web Framework*

Web Framework adalah suatu kumpula kode berupa pustaka dan alat yang dipadukan sedemikian rupa menjadi satu kerangka kerja guna memudahkan dan mempercepat proses pengembangan aplikasi *web.*

2.17.1 Framework Codeigniter

Codeigniter merupakan *framework web* untuk bahasa pemrograman PHP, yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab. CodeIgniter membiliki banyak fitur yan membantu para engembang PHP untuk dapat membuat aplikasi *web* secara mudah dan cepat. Adapun untuk kelebihan dari *codeigniter* yaitu:

Dipindai dengan CamScanner

1. Support pada PHP4 dan PHP5.
2. Lebih kecil dan cepat.
3. Menggunakan konsep MVC (Model, View dan Controller).

2.18 XAMPP

XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal. XAMPP berperan sebagai *web server* pada komputer. XAMPP juga dapat disebut sebagai sebuah *panel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses internet (Kadir, 2014).

Fungsi lainnya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP *Server*, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X, Apache, MySQL, PHP, dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.

2.19 MySQL

MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam basis data sejak lama, yaitu SQL (*Structure Query Language*). SQL adalah sebuah konsep pengoperasian basis data terutama untuk proses seleksi, pemasukan, pengubahan, dan penghapusan data yang dimungkinkan dapat dikerjakan dengan mudah dan otomatis (Sutaji, 2012).

MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Terdapat tiga kategori tipe data yang didukung oleh MySQL, yaitu tipe data numerik, string, serta penanggalan dan waktu. Sebuah data yang akan disimpan harus sesuai dengan tipe data yang bersangkutan (Wahana Komputer, 2015).

2.1 Kelunggulan MySQL

Dipindai dengan CamScanner

Berikut keunggulan dari MySQL, di antaranya adalah (Sutaji, 2012):

1. *Portability*

   Dapat berjalan stabil pada berbagai sistem operasi, di antaranya Windows, Linux, FreeBSD, MacOS X Server, Solaris, Asigma.

2. *Open source*

   Didistribusikan secara gratis di bawah lisensi dari *General Public License* (GPL), dimana setiap orang bebas untuk menggunakannya tetapi tidak boleh untuk dijadikan program induk turunan bersifat *close source* (komersial).

3. *Multi user*

   Dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan.

4. *Performance tuning*

   Mempunyai kecepatan yang tinggi dalam menangani *query*.

5. *Column types*

   Memiliki tipe data yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year,* dan *enum*.

6. *Command* dan *function*

   Memiliki operator dan fungsi penuh yang mendukung *select* dan *where* dalam *query*.

7. *Security*

   Memiliki beberapa lapisan sekuritas seperti tingkat *subnet mask, hostname, privilege user* dengan sistem perizinan yang mendetail serta *password* yang terenkripsi.

8. *Scalability* dan *limits*

   Mampu menangani basis data dalam jumlah besar, dengan jumlah *field* lebih dari 50 juta, 60 ribu tabel dan 5 miliar *record*. Batas indeks mencapai 32 buah per tabel.

9. *Localization*

   Dapat mendeteksi pesan kesalahan (*error code*) pada klien dengan menggunakan lebih dari 20 bahasa.

10. *Connectivity*

Dipindai dengan CamScanner

Dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix Socket, Named Pipes*.

11. *Interface*

    Memiliki antarmuka terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan API.

12. *Client* dan *tools*

    Dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi basis data sekaligus dokumen petunjuk *online*.

13. Struktur tabel

    Memiliki struktur tabel yang lebih fleksibel dalam menangani *alter* tabel dibandingkan dengan PostgreSQL dan Oracle.

2.19.2 Tipe Data MySQL

MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Penjelasan singkat kegunaan masing-masing jenis data akan sangat bermanfaat dalam memilih suatu jenis data yang dipakai dalam merancang *table* (Sutaji, 2012).

Beberapa jenis data yang tersedia pada MySQL dapat dilihat pada Tabel II.10 berikut

Tabel II.10 Tipe Data MySQL

Tipe Data	Keterangan
CHAR	Sebuah *string* dengan panjang tetap. Sisa jumlah karakter yang belum terisi akan diisi dengan spasi, akan tetapi spasi ini dibuang jika data dipanggil. Jangkauan nilai M adalah 1-255 karakter.
VARCHAR	*String* dengan panjang berupa variabel. M bisa mencapai 65535.
DATE	Data berupa tanggal. Format tanggal dalam bentuk 'YYYY-MM-DD'.
TIME	Data berupa waktu. Format waktu dalam bentuk 'HH:MM:SS'.
TINYINT	Bilangan antara -128 sampai dengan +127.

Dipindai dengan CamScanner

3.3 Metode Pengumpulan Data

Menurut Sugiyono (2013) teknik pengumpulan data merupakan langkah yang paling strategis dalam penelitian, karena tujuan utama dari penelitian adalah mendapatkan data. Penelitian ini menggunakan beberapa teknik pengumpulan data, yaitu:

1. Studi lapangan

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

   a. Pengamatan, tahap ini dilakukan secara langsung di divisi PE PT Mitsubishi Kramayudha Motors and Manufacturing dengan mengamati sistem informasi permintaan *tooling engine*, mulai dari *request* barang sampai pengecekan stok untuk pemberian barang yang di minta. Melalui teknik ini, data yang dibutuhkan diamati, dikumpulkan dan diolah sebagai bahan dalam penelitian.

   b. Wawancara, yaitu mencari data yang dibutuhkan secara langsung dengan memberikan pertanyaan spontan tentang proses bisnis permintaan *tooling engine* yang diperlukan pada penyusunan tugas akhir ini. Wawancara ini dilakukan kepada pegawai di bagian PE yaitu Bapak Agus Utomo dan pegawai di bagian gudang *tooling* yaitu Bapak Sarjuli.

2. Studi Literatur

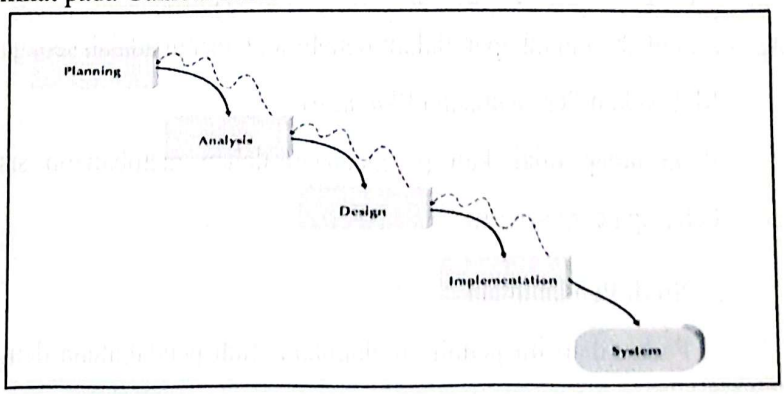
   Tahap ini merupakan tahap awal pengumpulan data dari sumber-sumber seperti buku yang berhubungan analisis dan desain sistem informasi, mengelola kebutuhan dan perintaan *tooling engine*, pemrograman web dengan Codeigniter, MySQL, pengembangan perangkat lunak, dan juga mencari data tambahan yang diperlukan seperti konsep pemrograman Codeigniter, beberapa contoh *coding* program melalui media *internet*.

Dipindai dengan CamScanner

3.4 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah metode pengembangan sistem *waterfall*. Metode *waterfall* ini terdiri dari tahap *planning, analysis, design,* dan *implementation*. Masing-masing fase tersebut tersusun dari beberapa langkah-langkah yang menghasilkan *deliverable* atau hasil kegiatan seperti beberapa dokumen spesifik dan file yang menjelaskan pemahaman tentang proyek. Pada penelitian ini untuk mengatasi masalah yang ada pada sistem, diputuskan untuk merancang dan mengembangkan sistem. Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode *waterfall*.

Metode *waterfall* sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Metode *waterfall* menyediakan pendekatan alur hidup perangkat lunak dengan terurut dari fase awal (*planning*) hingga fase akhir (*implementation*), dan pada setiap fase pihak pengembang dan sponsor akan selau ada kontak untuk mendapatkan persetujuan sebelum masuk pada tahap selanjunya (Dennis, 2010). Berikut ini akan dijelaskan secara singkat mengenai tahapan-tahapan yang dilakukan dalam pengembangan sistem dengan metode *waterfall*. Tahapan-tahapan metode *waterfall* menurut Dennis *et al* (2015) dapat dilihat pada Gambar III.1.



Gambar III.1 Metode *Waterfall*
(Sumber: Dennis *et al*, 2015)

1. Tahap Perencanaan (*Planning*)

   Pada tahap ini akan dilakukan perencanaan untuk membangun suatu sistem dengan membuat sebuah *system request*.

2. Tahap Analisis (*Analysis*)

Dipindai dengan CamScanner

Pada tahap ini, merupakan proses analisis kebutuhan sistem. Analis mengumpulkan data-data sebagai bahan pengembangan sistem. Pengumpulan data dapat dilakukan dengan teknik wawancara maupun teknik observasi.

3. Tahap Desain (*Design*)

   Proses desain pengembang melakukan desain seperti desain struktur data, dan representasi antarmuka. Data-data yang didapat dari tahap analisis diterapkan dalam tahap desain.

4. Tahap Implementasi (*Implementation*)

   Pada tahap ini, pengembang akan membangun sebuah sistem dengan rancangan yang sudah ada diterjemahkan ke dalam sebuah *code* atau aktivitas *coding*. Bahasa pemrograman yang digunakan adalah PHP dengan menggunakan *framework* CodeIgniter dan MySQL sebagai *database* yang digunakan.

3.5 Kerangka Penelitian

Dalam penelitian yang dibahas dalam Tugas Akhir ini, dilakukan langkah-langkah atau tahapan dalam penelitian yang ada pada Gambar III.1 sesuai dengan metodologi penelitian yang digunakan dalam penelitian tersebut. Penjelasan langkah-langkah atau tahapan dalam penelitian tersebut adalah sebagai berikut:

1. Melakukan Perencanaan (*Planning*)

   Pada tahap dilakukan perencanaan dalam membangun sistem dengan beberapa cara seperti:

   a. Studi Pendahuluan

      Pada tahap ini penulis melakukan studi pendahuluan dengan beberapa cara seperti:

      1) Melakukan studi pustaka dengan mempelajari buku referensi, jurnal-jurnal yang serupa dengan topik yang diambil, dan menelaah dari berbagai laporan terkait seperti, struktur organisasi dalam divisi PE, proses permintaan yang dilakukan di perusahaan, dan proses bisnis yang digunakan di PT Mitsubishi Kramayudha Motors and Manufacturing.

Dipindai dengan CamScanner

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Sejarah Industri

Pada tanggal 18 Januari 1973 terjadi persetujuan usaha patungan (*Joint*Venture) antara PT. Krama Yudha, Mitsubitshi Corporation, dan Mitsubishi Motor Corporation yang kemudian pada tanggal 19 Mei 1973 didirikan PT.Mitsubishi Krama Yudha Motors and Manufacturing yang memiliki modal dasar $ 42.866.250.

PT. Mitsubishi Krama Yudha Motors and Manufacturing ini merupakan industri dengan penanaman modal asing (PMA) yang bekerja sama dengan jepang dengan pemegang sahamnnya terdiri dari :

1. PT Krama Yudha, Indonesia : 18,22 %
2. PT Krama Yudha Tiga Berlian Motors, Indonesia : 17,22 %
3. Mitsubishi Corporation, Jepang : 32,28 %
4. Mitsubishi Fuso Truck and Bus Corporation, Jepang : 32,28 %

Tahap – tahap pengembangan diawali dengan pembangunan mesin pabrik dimulai pada tanggal 14 Januari dan selesai dalam waktu empat bulan yang dilanjutkan dengan pemasukan mesin – mesin dan peralatan pada bulan Mei.Produksi percobaan dimulai pada pertengahan bulan Oktober 1974 yang berlangsung untuk beberapa bulan sedangkan produksi komersial dimulai pada tanggal 6 Januari 1975.

Pada bulan Desember 1982 didirikan PT. Colt Engine and Manufacturing (CEM) yang merupakan usaha patungan anatara Indonesia dan Jepang yaitu PT. Mitsubishi Corporation dan Mitsubishi Motors Corporation PT. Colt Engine and Manufacturing mulai berproduksi secara komersial pada tahun 1985. Pada tanggal 1 Januari 1982 PT. Mitsubishi Krama Yudha Motors and Manufacturing melakukan mergerdengan PT. Colt Engine and Manufacturing yang kemudian disebut PT. Mitsubishi Krama Yudha Motors and Manufacturing.

Dipindai dengan CamScanner

2. Melakukan Analisis (*Analysis*)

Pada tahap ini penulis melakukan identifikasi kebutuhan sistem dengan beberapa cara seperti:

a. Analasis sistem yang berjalan yaitu, melakukan analisis dokumen *input* dan *output* yang berjalan dan menganalisis proses bisnis yang berjalan di PT Mitsubishi Kramayudha Motors and Manufacturing.

b. Analisis sitem usulan yaitu, mengidntifikasi kebutuhan pengembangan sistem informasi yang ada di PT Mitsubishi Kramayudha Motors and Manufacturing terutama divisi PE bagian gudang *tooling*.

3. Melakukan Desain (*Design*)

Pada tahap ini penulis melakukan perancangan sistem dimana dalam pembuatan sistem menggunakan dua proses yaitu:

a. Membuat *flowmap* usulan untuk mengetahui alur proses bisnis dari sistem yang digunakan.

b. Menggunakan *usecase diagram* yang bertujuan untuk menggambarkan proses sistem yaitu kebutuhan sistem dari sudut pandang *user* , menffunakan *activity* diagram yang bertuuan menggambarkan proses bisnis modeling untuk memperlihatkan urutan aktivitas proses bisnis, menggunakan *sequence diagram* yang bertujuan untuk menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu kejadian untuk menghasilkan output, menggunakan *class diagram* bertujuan menggambarkan struktur hubungan dan deskripsi antar class.

c. Menggunakan *Entity Relationship Diagram* (ERD) yang bertujuan untuk memodelkan suatu data dan dapat dilakukan juga untuk perencanaan *database* serta hubungan data dalam pembuatan kamus data dengan tujuan memvalidasi diagram alir data untuk hal kelengkapan dan keakuratan data.

d. Menggunakan *Windows Navigation* (WND) untuk desain *interface*.

Dipindai dengan CamScanner

4. Melakukan Implementasi (*Implementation*)

Pada tahap implementasi perangkat lunak yang dilakukan yaitu:

a. *Coding* pada aplikasi yang akan dibuat sesuai dengan kebutuhan yang ada dengan Menggunakan CodeIgniter dan *MySql* 5.5.32 dalam menuliskan kode program untuk pembuatan aplikasi sistem yang dibutuhkan. Selain itu menggunakan juga XAMPP, XAMPP merupakan sebuah server yang berdiri sendiri *(localhost)* yang bertujuan untuk mempermudah dalam mengoperasikaan sistem operasi *MySq*l 5.5.32 dan CodeIgniter. *MySql* merupakan sistem operasi yang digunakan untuk pembuatan *database*. Menggunakan *MySql* ini dikarenakan *MySql* merupakan *database* yang baik untuk digunakan dalam pembuatan aplikasi menggunakan CodeIgniter.

b. Membuat dokumentasi dalam bentuk laporan pengembangan Sistem Informasi Permintaan *tooling engine* Di PT Mitsubishi Kramayudha Motors and Manufacturing Berbasis Web.

Berikut adalah Gambar III.2 *flowchart* kerangka penelitian untuk menyelesaikan masalah dalam Tugas Akhir ini:

Dipindai dengan CamScanner

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Sejarah Industri

Pada tanggal 18 Januari 1973 terjadi persetujuan usaha patungan (*Joint*Venture) antara PT. Krama Yudha, Mitsubitshi Corporation, dan Mitsubishi Motor Corporation yang kemudian pada tanggal 19 Mei 1973 didirikan PT.Mitsubishi Krama Yudha Motors and Manufacturing yang memiliki modal dasar $ 42.866.250.

PT. Mitsubishi Krama Yudha Motors and Manufacturing ini merupakan industri dengan penanaman modal asing (PMA) yang bekerja sama dengan jepang dengan pemegang sahamnnya terdiri dari :

1. PT Krama Yudha, Indonesia : 18,22 %
2. PT Krama Yudha Tiga Berlian Motors, Indonesia : 17,22 %
3. Mitsubishi Corporation, Jepang : 32,28 %
4. Mitsubishi Fuso Truck and Bus Corporation, Jepang : 32,28 %

Tahap – tahap pengembangan diawali dengan pembangunan mesin pabrik dimulai pada tanggal 14 Januari dan selesai dalam waktu empat bulan yang dilanjutkan dengan pemasukan mesin – mesin dan peralatan pada bulan Mei.Produksi percobaan dimulai pada pertengahan bulan Oktober 1974 yang berlangsung untuk beberapa bulan sedangkan produksi komersial dimulai pada tanggal 6 Januari 1975.

Pada bulan Desember 1982 didirikan PT. Colt Engine and Manufacturing (CEM) yang merupakan usaha patungan anatara Indonesia dan Jepang yaitu PT. Mitsubishi Corporation dan Mitsubishi Motors Corporation PT. Colt Engine and Manufacturing mulai berproduksi secara komersial pada tahun 1985. Pada tanggal 1 Januari 1982 PT. Mitsubishi Krama Yudha Motors and Manufacturing melakukan mergerdengan PT. Colt Engine and Manufacturing yang kemudian disebut PT. Mitsubishi Krama Yudha Motors and Manufacturing.

Dipindai dengan CamScanner

4.2 Visi dan Misi Industri

PT Mitsubishi Krama Yudha Motors and Manufacturing mempunyai visi dan misi sebagai berikut:

Visi Industri :

1. Menjadikan industri yang mampu bersaing dengan industry sejenis di Indonesia dengan hasil prosuksi yang baik dan berkualitas tinggi.
2. Mengelola pabrik yang aman dan maju dengan melaksanakan *control* QCD (*Quality, Cost, Delivery*) dengan mempunyai tanggung jawab terhadap lingkungan dan menempatkan priorotas utama untuk mendapatkan kepercayaan konsumen.
3. Meningkatkan kepuasan kepada pemilik saham, pemerintah, direktur, dan seluruh karyawan.

Misi Industri :

1. Penurunan Biaya.
2. Peningkatan kualitas.
3. Pengawasan terhadap jadwal pengiriman
4. Mengadakan persiapan yang lancer dan baik untuk produk model baru.
5. Peningkatan dalam bidang menejemen, keselamatan dan lingkungan.

4.3 Kegiatan Usaha Industri

PT. Mitsubishi Krama Yudha Motors and Manufacturing (PT MKM) merupakan industry yang memproduksi komponen untuk kendaraan, baik itu komponen untuk badan kendaraan maupun komponen untuk mesin kendaraan, oleh karena itu dalam kegiatan terbagi menjadi dua bagian yang masing – masing menghasilkan produksi yang berlainan, adapun bagian – bagian tersebut adalah :

1. *Stamping Factory*

Pada bagian ini memproduksi komponen untuk badan rangka kendaraan, dimana kegiatannya adalah mengelola lempengan logam dari pemotongan, pengeprsan

Dipindai dengan CamScanner

sampai perakitan untuk menghasilkan kendaraan yang berkualitas tinggi. Adapun komponenyang dihasilkan adalah :

a. *Body* kendaraan
b. *Chasing*
c. *Steering*
d. *Fuel Tank*
e. *Exhaust Pipe*
f. *Muffler*

2. *Engine Factory*

Pabrik ini memproduksi berbagai macam komponen untuk mesin kendaraan pabrik ini juga merakit komponen – komponenmesin, baik itu local maupun komponen yang berasal dari luar menjadi satu mesin yang siap pakai dengan kualitas tinggi. Adapun komponen yang dihasilkan antara lain ;

a. *Cylinder Head*
b. *Crank Shaft*
c. *Cylinder Block*
d. *Connection Rad*
e. *Cam Shaft*
f. *Transmission Case*
g. *Extension Housing*

Dalam menjalan kegiatan produksinya PT. Mitsubishi Krama Yudha Motors and Manufacturing memproduksi komponen tersebut berdasarkan jumlah pesanan yang diterima dari distributor tunggal kendaraan Mitsubishi di Indonesia, yang dipegang oleh PT. Krama Yudha Tiga Berlian (KTB) yang didirikan pada tanggal 23 april 1973.

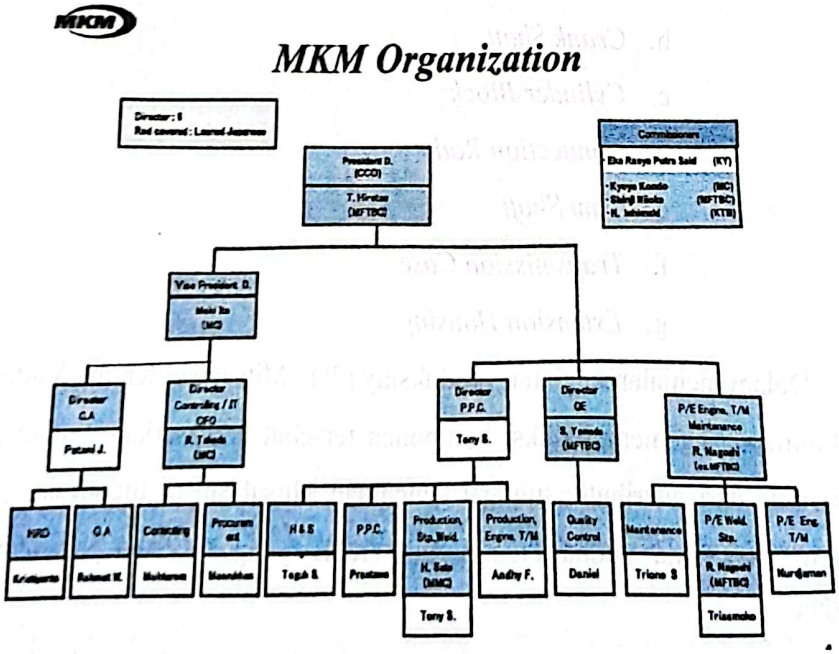
Produksi yang dihasilkan oleh PT. Mitsubishi Krama Yudha Motors and Manufacturing di distribusikan ke PT. Krama Yudha Ratu Motors (KRM) untuk jenis T120SS, L30 dan Truk.PT.Krama Yudha Ratu Motors (KRM) sendiri didirikan pada tannggal 2 Juni 1973. Ketiga industry tersebut (PT. KTB, PT.MKM, dan PT.KRM)

Dipindai dengan CamScanner

tergabung ke dalam Krama Yudha Group, yang dalam menjalankan usahannya saling berkaitan dan saling membantu antar industry yang satu dengan industry yang lain.

4.4 Struktur Organisasi Perusahaan

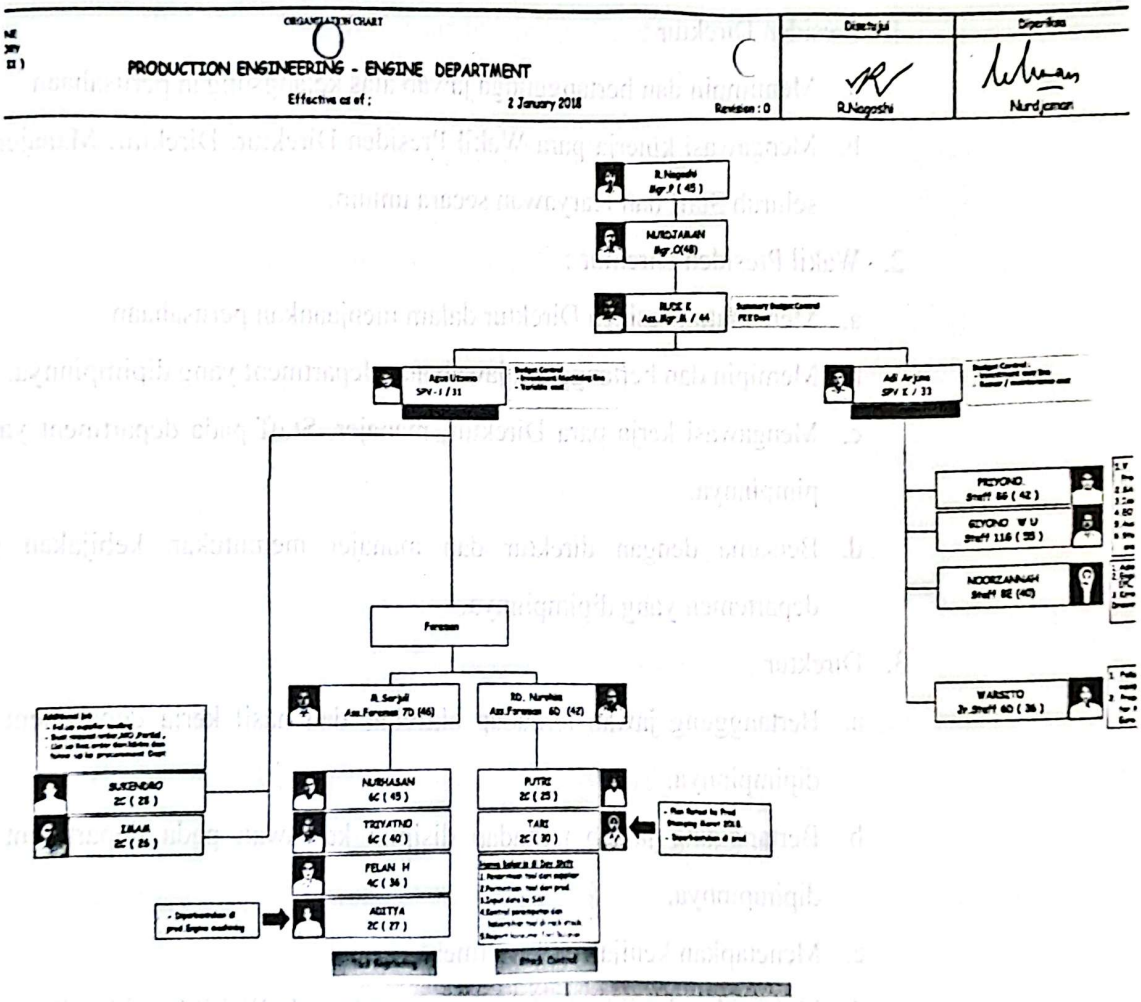
Untuk menjalankan usahannya, setiap industry memerlukan suatu struktur organisasi yang baik.Struktur organisasi dapat diartikan sebagai susunan dan hubungan antara bagian dan posisi dalam suatu industri.Struktur organisasi menggambarkan pembagian kerja, pelimpahan wewenang, kesatuan perintah dan tanggung jawab yang jelas. Struktur Organisasi yang tersusun dengan baik akan memudahkan koordinasi, integrasi, serta meningkatkan efektivitas dan efisiensi suatu industry didalam mencapai tujuannya. Berikut merupakan struktur organisasi PT. Mitsubishi Krama Yudha Motors and Manufacturing (PT MKM)



Gambar IV.1 Struktur Organisasi PT Mitsubishi Krama Yudha Motors and Manufacturing
(Sumber: PT.MKM II, 2017)

Dipindai dengan CamScanner

Berikut merupakan struktur organisasi pada Divisi PE – E :



Gambar IV.2 Struktur Organisasi Divisi PE-E
(Sumber: PT MKM II, 2017)

4.4.1 Tugas dan Wewenang setiap jabatan di PT MKM

Dengan adanya suatu organisasi yang baik, maka memungkinkan terselenggarannya suatu kelancaran dalam proses usaha yang dilakukan oleh industry dikarenakan dalam organisasi tersebut sudah ditetapkan dengan jelas mengenai wewnang, tugas, tanggung jawab seseorang. PT MKM mempunyai pimpinan tertinggi yang dipegang oleh Presiden Direktur yang dibantu oleh 2 orang Wakil Presiden Direktur dimana setiap direktur membawahi beberapa department sebagai berikut :

Dipindai dengan CamScanner

1. Presiden Direktur :
   a. Memimpin dan bertanggunga jawab atas kelangsungan perusahaan
   b. Mengawasi kinerja para Wakil Presiden Direktur, Direktur, Manajer, dan seluruh Staff dan Karyawan secara umum.
2. Wakil Presiden Direktur :
   a. Membantu Presiden Direktur dalam menjaankan perusahaan
   b. Memipin dan bertanggung jawab atas department yang dipimpinnya.
   c. Mengawasi kerja para Direktur, manajer, Staff pada department yang di pimpinnya.
   d. Bersama dengan direktur dan manajer memntukan kebijakan untuk departemen yang dipimpinnya.
3. Direktur
   a. Bertanggung jawab terhadap aktivitas dan hasil kerja department yang dipimpinnya.
   b. Bertanggung jawab terhadap disiplin karyawan pada department yang dipimpinnya.
   c. Menetapkan kenijakan department.
   d. Melaporkan hasil kerja departemennya kepada Wakil Presiden Direktur.
4. Manager
   a. Bertanggung jawab terhadap aktivitas dan hasil kerja department yang dipimpinnya.
   b. Bertanggung jawab terhadap pelaksanaan displin karyawan pada department yang dipimpinnya.
   c. Menetapkan dan memgawasi jalannya kebijakan department
   d. Melaporkan hasil kerja departemennya pada Wakil Presiden Direktur.
5. HRD Department
   a. Bertanggung jawab terhadap aktivitas dan hasil kerja department
   b. Bertanggung jawab terhadap hubungan industrial

Dipindai dengan CamScanner

c. Sebagai wakil perusahaan dalam perundingan bipartite dengan PUK (Pimpinan Unit Kerja) dan hubungan dengan instansi pemerintah
d. Bertanggung jawab terhadap pengadaan dan pengembangan karyawan
e. Bertanggung jawab terhadap peningkatan keterampilan dan pengetahuan karyawan.
f. Bertanggung jawab terhadap disiplin karyawan
g. Bertanggung jawab terhadap pelaksanaan BOD Meeting
h. Bertanggung jawab terhadap pelaksanaan magang kerja atau praktik kerja lapangan yang menjadi rogram perusahaan.

6. General Affairs Department
   a. Bertanggung jawabterhadap aktivitas dan hasil kerja di GAD
   b. Bertanggung jawab terhadap pelaksanaan BOD Meeting
   c. Bertanggung jawab terhadap kesejahteraan karyawan
   d. Bertanggung jawab terhadap keselamatan karyawan dan aset perusahaan
   e. Menjalin kerja sama yang baik antar perusahaan
7. Marketing Department
   a. Bertanggung jawab dalam mengatur atau mengkontrol pembelian baik di pasar domestic serta ical vendor
   b. Bertanggung jawab dalam pembelian sub material dan equipment dari agen atau supplier dalam negri
   c. Beranggung jawab dan mengontrol pembelian raw material, equipments tool import.
8. Controling Department
   a. Penagihan ke pelanggan setelah sepakat mengenai harga jual
   b. Tanda tangan persetujuan tagihan pelanggan
   c. Laporan penjualan
   d. Menindak lanjuti dan menyelesaikan masalah pembayaran
   e. Anggaran penjualan
   f. Proposal anggaran penjualan

Dipindai dengan CamScanner

g. Laporan anggaran penjualan
h. Monitor terhadap kas keluar (pembayaran)
i. Monitor terhadap kas masuk
j. Check dan persetujuan PO (Purchase Order / Order Pembelian) dan WO (Work Order / Order Pekerjaan)
k. Pengecekan untang dan piutang
l. Pengecekan dan persetujuan laporan keuangan

9. Product Stamping Department
   a. Membuat system antara lain : control produksi, control kualitas, control afety, control sub material, control mesin dan alat, control man power, pendidikan
   b. Improvement antara lain : lay out, cara kerja, cost down, man hour
10. Production Engineering Engine (PE-E)
   a. Follow up dan koordinasi pekerjaan staff PE
   b. Kontrol ide perbaikan, keselamatan, 2s, over time.
   c. Membantu dalam merencanakan perbaikan dan pemanfaatan layout pabrik
   d. Membantu dalam perencanaan penurunan biaya
   e. Menjabarkan sasaran atau target department.
11. Production Planning and Control (PPC)

   Stamping :

   a. Bertanggung jawab terhadap perencanaan dan pengawasan produksi yang meliputi: CKD, Material sub kontraktor, dan vendor, produksi welding, produksi hand work, produksi steering, produksi frame dan domestic serta export delivery
   b. Bertanggung jawab terhadap prasarana kebutuhan perencanaan dan pengawasan produksi di pabrik *stamping*

   Engine :

   a. Bertanggung jawab terhadap perencanaan dan pengawasan produksi yang meliputi: CKD, Vendor, produksi *assy line*, produksi *transimis assy line*,

Dipindai dengan CamScanner

produksi *streering assy line,* produksi machining line dan *domestic* serta *export delivery.*

b. Bertanggung jawab terhadap prasarana kebutuhan perencanaan dan pengaasanproduksi di pabrik *engine*.

c. Bertanggung jawab terhadap pelaksanaan tugas dan pengawasan produksi di pabrik *engine*.

12. Production Engineering Stamping (PE-S)

a. Manajemen total pekerjaan di department

a.1 Proses realisasi produk baru/ganti model

a.2 pelaksanaan EO/perubahan barang/ECR

a.3 Proses pembuatan master *schedule*

a.4 die/jig manufacturing dan maintenance

b. leader total pekerjaan di department

c. monitoring total pekerjaan department

Dipindai dengan CamScanner

c. Mengatur kebutuhan listrik perusahaan agar proses produksi dapat berjalan sesuai rencana

d. Mengatur kondisi pergudangan serta melaporkan kondisi pergudangan atau stock yang dimiliki perusahaan.

4.4.2 Tugas dan Wewenang Di divisi *Production Engineering Engine* (PE-E) PT MKM II

1. Product Engineering Engine Department (PE-E)
   a. Follow up dan berkoordinasi pekerjaan staff PE (Product Engine)
   b. Kontrol Ide perbaikan, keselamatan, 2s, over time
   c. Membantu dalam merencanakan perbaikan dan pemanfaatan layout pabrik
   d. Membantu dalam perancangan penurunan biaya
   e. Menjabarkan sasaran atau target department
2. *Tool Regrinding*
   a. Menyiapkan atau mengadakan kembali *tooling* yang didapat dari *line* untuk *machine*
   b. Pengadaan penyimpanan untuk *tool* yang telah di regrinding
   c. Mengasah kembali *tool* yang masih layak dipakai
3. *Cutting Tools*
   a. Mengkontrol bagian *cutting* pada mesin
   b. Pengadaan penyimpanan untuk *cutting* tool dan setup *tool*
   c. Mensupply *cutting tool* dan setup *tool* ke *line*

4.5 Analisis Sistem yang Berjalan

Analisis sistem merupakan langkah untuk menentukan seberapa jauh sistem yang berjalan telah mencapai sasarannya dan mengetahui kelemahan-kelemahannya sehingga dapat diusulkan perbaikan. Oleh karena itu, dalam rangka untuk mengidentifikasi masalah dan kelemahan-kelemahan yang ada pada sistem yang

Dipindai dengan CamScanner

sedang berjalan saat ini maka dilakukan analisis sistem permintaan *tooling* di divisi Production Engineering pada PT Mitsubishi Kramayudha Motors And Manufacturing. Teknik pengumpulan fakta dan data yang dilakukan melalui wawancara, diskusi dan observasi. Wawancara, diskusi dan observasi tersebut dilakukan ke beberapa karyawan yang berada di Divisi PE dan bagian *warehouse tooling shop.*

Dalam merencanakan jumlah tooling yang dibutuhkan untuk proses produksi agar sesuai dengan kebutuhan mesin, Line Produksi mengajukan beberapa dokumen kepada divisi terkait untuk diproses lebih lanjut.

4.6 Analisis Dokumen

Analisis dokumen berikut ini akan menjelaskan dokumen apa saja yang digunakan dalam proses perencanaan bahan baku di divisi PE pada PT MKM terdiri dari dokumen masukan dan keluaran.

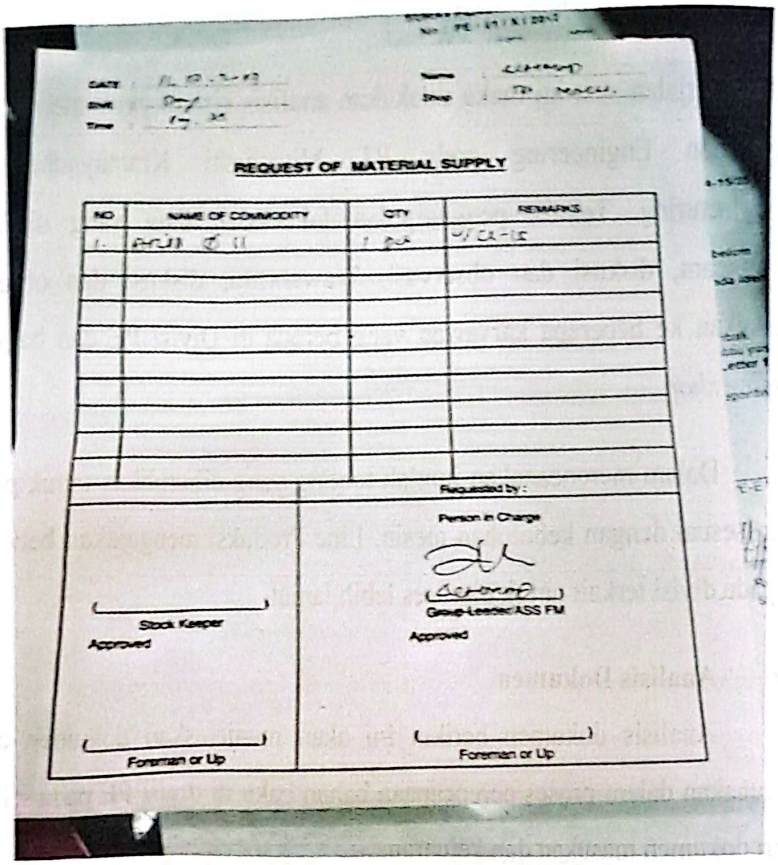
4.6.1 Dokumen Masukan

Dokumen masukan berfungsi untuk menjelaskan mengenai semua dokumen *input* yang digunakan pada proses perencanaan bahan. Berikut adalah bentuk dokumen masukan yang ada di PT Mitsubishi Krama Yudha Motors and Manufacturing.

1. *Request Material of Supply*

   Request Material of Supply adalah form yang masuk ke bagian gudang *tooling shop* dari *line* produksi untuk meminta *tooling engine*, form RMS pada gambar IV.5

Dipindai dengan CamScanner



REQUEST OF MATERIAL SUPPLY

NO	NAME OF COMMODITY	QTY	REMARKS
1.			

Requested by :
Person in Charge
Group Leader/ASS FM
Approved
Foreman or Up
Stock Keeper
Approved
Foreman or Up

Gambar IV.3 *Request Material of Supply*
(Sumber: PT MKM II, 2017)

Keterangan:

1. Asal : Line Produksi
2. Tujuan : PE-E *Tooling* Shop
3. Date : Menunjukan tanggal permintaan
4. Shift : Menunjukan waktu pergantian
5. Time : Menunjukan waktu jam permintaan
6. Nama : Nama peminta *tooling*
7. Shop : Keterangan dari line produksi mana
8. Kolom-kolom :
   - No : Nomor urut *material*
   - Name of commondity : Nama *material*
   - Banyaknya : Jumlah *material* yang dipesan
   - Remarks : Tanda tangan dari peminta

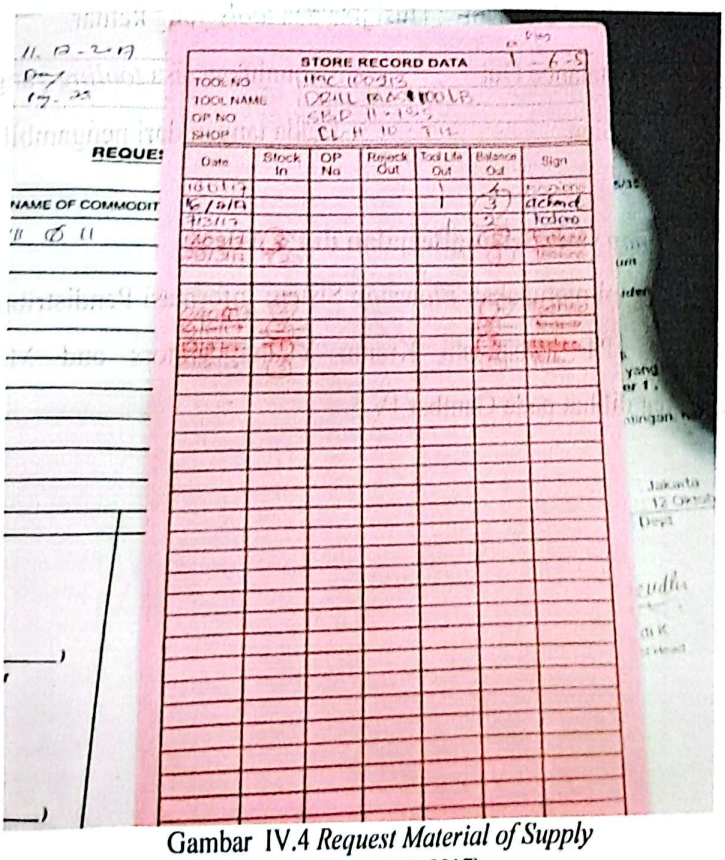
Dipindai dengan CamScanner

4.6.2 Dokumen Keluaran

Dokumen masukan berfungsi untuk menjelaskan mengenai semua dokumen *output* yang digunakan pada proses perencanaan bahan. Berikut adalah bentuk dokumen masukan yang ada di PT Mitsubishi Krama Yudha Motors and Manufacturing:

1. Kartu Stok

   Kartu stok berisi data stok tools engine. Pemeriksaan *tools* yang tersedia. Memonitor *tools* yang ada. Kartu stok pada gambar IV.4 diisi oleh petugas *Tooling Shop* dan kemudian diletakkan pada tiap rak. Kartu stok digunakan saat proses barang jika ada *tooling* masuk atau *tooling* keluar.



Gambar IV.4 *Request Material of Supply*
(Sumber: PT MKM II, 2017)

Keterangan:

1. Asal : PE-E *Tooling Shop*

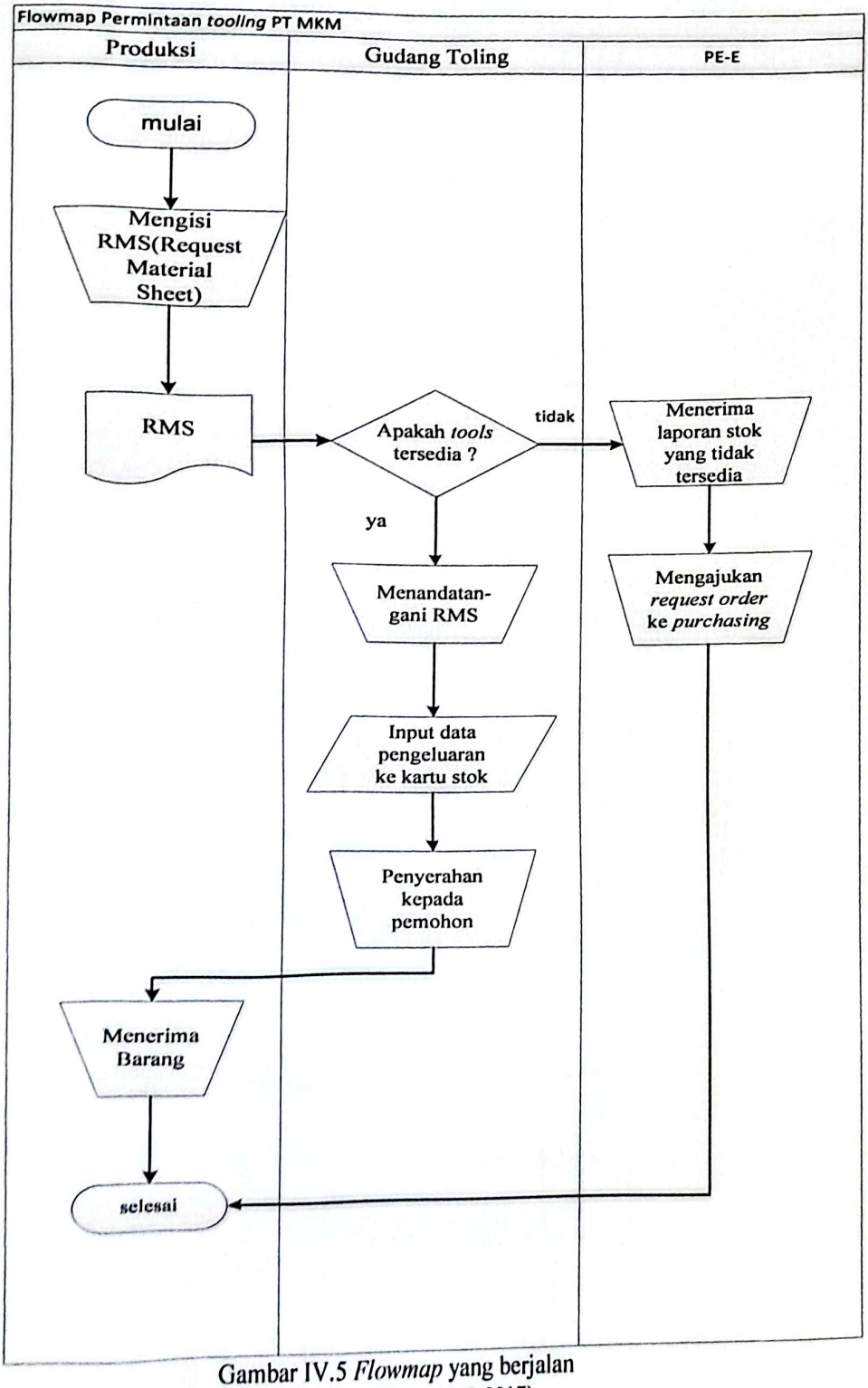
Dipindai dengan CamScanner

2. Tujuan : PE-E *Tooling Shop*
3. Tool No : Menunjukan kode *tooling* yang dimaksud
4. Tool Name : Menunjukan tooling *tooling* yang dimaksud
5. OP No : Menunjukan nomer Operasional
6. Shop : Keterangan dari line produksi mana
7. Kolom-kolom :
    - Date : Menunjukan tanggal tools masuk atau keluar
    - Stock In : Diisi Jika ada *tools* yang baru masuk
    - Op No : Menunjukan nomer operasional
    - Reject Out : Menunjukan jika ada barang yang reject
    - Tool Life Out : Diisi jika ada *tools* yang keluar
    - Balance Out : Menunjukan sisa *tooling* yang ada di rak
    - Sign : Tanda tangan dari pengambil *tools*

4.7 *Flowmap* yang SedangBerjalan di PT MKM

Berikut ini merupakan *Flowmap* Sistem Informasi Pendistribusian yang sedang berjalanpada PT Mitsubishi Krama Yudha Motors and Manufacturing (PT MKM)dapat dilihat pada Gambar IV.5.

Dipindai dengan CamScanner



Gambar IV.5 *Flowmap* yang berjalan
(Sumber: PT MKM II, 2017)

Dipindai dengan CamScanner

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

BAB V

ANALISIS DAN PEMBAHASAN

5.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem untuk sistem informasi permintaan *tooling engine* dapat dilihat pada Tabel V.1 sebagai berikut:

5.1.1 Kebutuhan Fungsional Sistem

Tabel V.1 Analisis Kebutuhan Fungsional Sistem

No.	Permasalahan	Kebutuhan *User*	Solusi	Kebutuhan Fungsional Sistem
1.	Setiap meminta *tools* dari bagian produksi ke gudang *tooling* masih menggunakan kertas belum terinput ke sistem.	Sistem yang dapat membntu menginput data permintaan *tooling*	Merancang dan membangun suatu sistem yang dapat membantu dalam permintaan *tooling*	- Mengelolah data permintaan *tooling*
2.	Mengiput data stok dari kartu stok kes sistem masih menggunakan *Microsoft Exel*	Sistem yang apat membantu melihat stok *tooling*	- Membuat sistem yang dapat menginput data stok *tooling* ke dalam *database* - Membuat sistem yang bisa meng*update* stok *tooling*	- Mengelolah data *tooling*

Dipindai dengan CamScanner

5.1.2 Kebutuhan Non Fungsional Sistem

Kebutuhan Non Fungssional dari sistem informasi permintaan yang diusulkan antara lain

1. Aplikasi harus dijalankan menggunakan *mouse* dan *keyboard*
2. Aplikasi dapat dijalankan oleh Bagian Produksi dan Bagian Gudang *Tooling*
3. *Login* dengan memasukkan *username* dan *password*.
4. Aplikasi akan beroperasi di lintas *platform*.

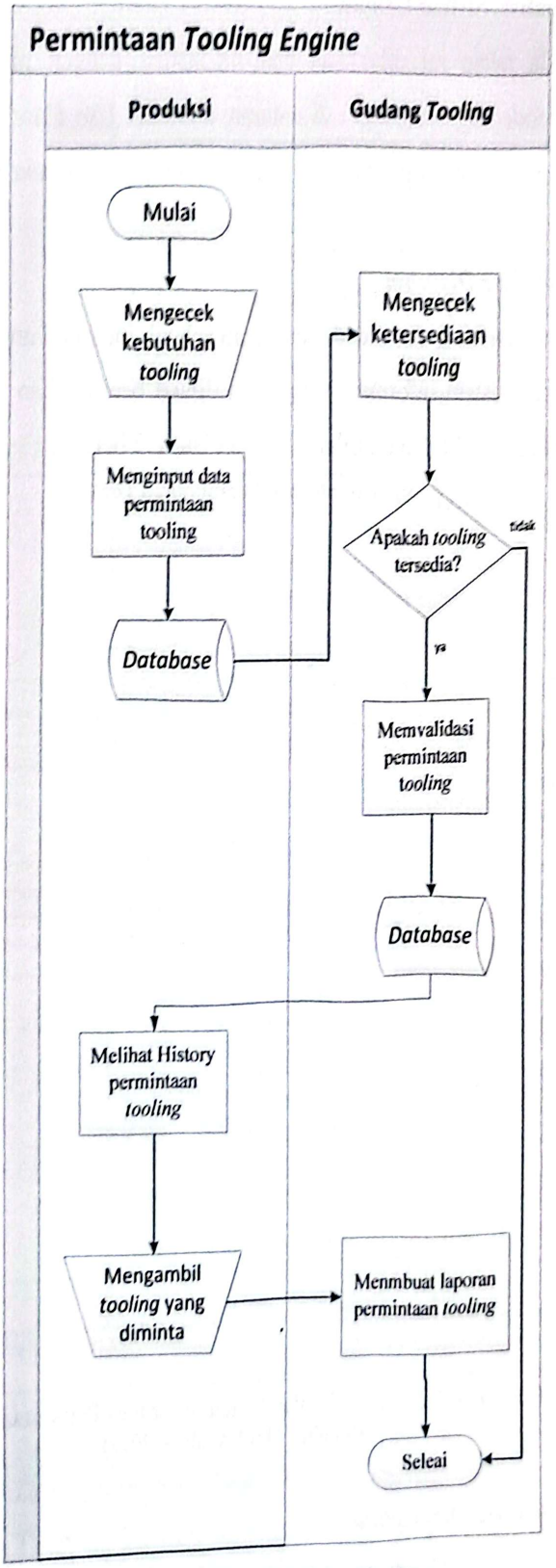
5.2 Prosedur Usulan Sistem Informasi Permintaan *Tooling Engine*

Prosedur sistem informasi *tooling engine* yang diusulkan sebagai bentuk perbaikan sistem informasi permintaan *tooling engine* di PT Mitsubishi Kramayudha Motors and manufacturing melalui tahapan-tahapan sebagai berikut:

1. Produksi mengidentifikasi kebutuhan *tooling*.
2. Setelah itu menginput *database* ke dalam sistem (data *tools* yang dibutuhkan).
3. Selanjutnya jika permintaan di *approve* oleh bagian gudang *tooling shop* bisa langsung diambil ke bagian gudang *tooling shop*.
4. Bagian Produksi bisa melihat *history* permintaan *tooling*
5. Bagian gudang *tooling shop* membuat laporan permintaan *tooling*.

PT Mitsubishi Kramayudha Motors and Manufacturing, seperti pada gambar IV.1 dibawah ini:

Dipindai dengan CamScanner



Gambar V.1 *Flowmap* Usulan
(Sumber: Hasil Analisis, 2018)

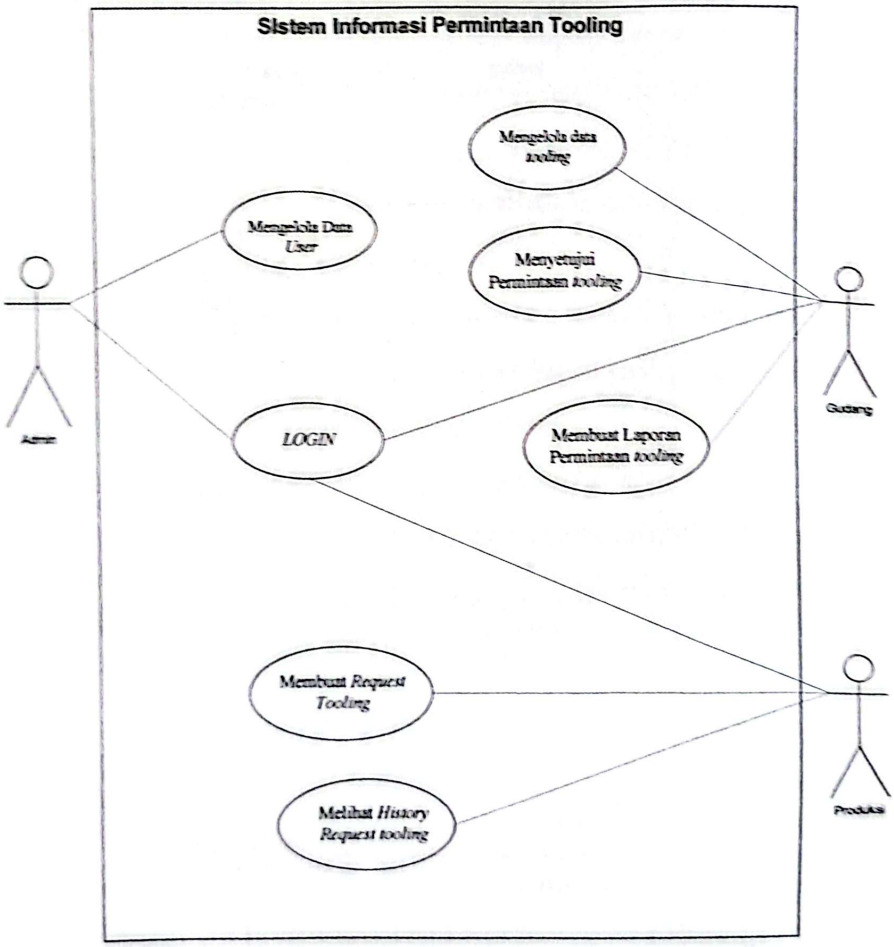
Dipindai dengan CamScanner

5.3 Analisis Sistem Usulan

Pada tahap ini dilakukan analisis sistem dengan menggunakan UML (*Unified Modelling Language*) di antaranya adalah *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Deployment Diagram*.

5.3.1 *Use Case Diagram*

Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat berdasarkan analisis kebutuhan sistem pada tabel V.I dapat dilihat pada gambar V.2 berikut.



Gambar V.2 *Use Case* Usulan Sistem Informasi Permintaan Tooling
(Sumber: Hasil Analisis, 2018)

5.3.1.1 *Use Case Description*

Berikut adalah *use case description* dari *use case* usulan pada gambar V.2:

1. *Use Case Description* Melakukan *Login*

Dipindai dengan CamScanner

Tabel V.2 *Use Case Description Login*

Nama *Use Case*	*Login*
Aktor	*Admin,* Produksi, Gudang *Tooling*
Deskripsi	*Use case* ini menggambarkan aktor dapat masuk ke dalam sistem.
Normal Flow of Events	1. Aktor membuka *login form* 2. Aktor memasukan *username* dan *password* pada *login form.* 3. Sistem akan memvalidasi data, apakah data benar atau tidak 4. Jika data benar, maka aktor akan masuk ke dalam halaman utama sistem 5. Jika data salah, maka aktor tidak dapat masuk ke sistem menandakan bahwa data tidak valid.
Alternate Flow	

Sumber: Hasil Analisis, 2018

2. *Use Case Description* Mengelolah Data *Users*

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

Nama *Use Case*	Mengelola data *users*
Aktor	*Admin*
Deskripsi	*Use case* ini menggambarkan aktor mengelola data *users.*
Normal Flow of Events	1. Aktor memilih menu *users* lalu memilih pilihan add *users.* 2. Sistem akan mengambil semua informasi data *users* yang tersimpan pada *database* dan menampilkan informasi tersebut kedalam tabel data *users.*
Alternate Flow	

Sumber: Hasil Analisis, 2018

3. *Use Case Description* Mengelolah Data *Tooling*

Tabel V.4 *Use Case Description* Mengelola Data *Tooling*

Nama *Use Case*	Mengelola data *Tooling*
Aktor	Gudang *Tooling*

Dipindai dengan CamScanner

Deskripsi	*Use case* ini menggambarkan bagian gudang *tooling* mengelola data *tooling*.
Normal Flow of Events	1. Aktor Memilih menu *tools* lalu memilih pilihan tambah *tools*. 2. Sistem akan mengambil semua informasi data *tooling* yang tersimpan pada *database* dan menampilkan informasi tersebut kedalam tabel data *tooling*. 3. Dapat melihat *detail database* yang tersimpan dan dapat juga menambahkan, mengubah dan menghapus data *tooling*.
Alternate Flow	

Sumber: Hasil Analisis, 2018

4. *Use Case Description* Membuat *Request tooling*

Tabel V.5 *Use Case Description* Membuat *Request tooling*

Nama *Use Case*	Membuat *Request tooling*
Aktor	Gudang *tooling*
Deskripsi	*Use case* ini menggambarkan bagian produksi meminta *tooling* apa saja yang ingin di gunakan
Normal Flow of Events	1. Aktor memilih menu *request* lalu memilih pilihan *request form* 2. Dapat melihat data *tooling* yang tersedia di bagian gudang dan dapat melihat *history* permintan barang apakah di *approve* oleh bagian gudang atau tidak.
Alternate Flow	

Sumber: Hasil Analisis, 2018

5. *Use Case Description* Melihat *History Request tooling*

Tabel V.6 *Use Case Description* Melihat *History Request tooling*

Nama *Use Case*	*Input* melihat *history request tooling*
Aktor	Bagian Produksi

Dipindai dengan CamScanner

Tabel V.6 *Use Case Description* Melihat *History Request tooling* (Lanjutan)

Deskripsi	*Use case* ini menggambarkan bagian produksi dapat melihat *form* request *history* .
Normal Flow of Events	1. Aktor memilih menu *request* lalu pilih sub menu *request history* 2. Dapat melihat status permintaan apakah sudah di *approve* atau belum
Alternate Flow	

Sumber: Hasil Analisis, 2018

6. *Use Case Description* Menyetujui Permintaan *Tooling*

Tabel V.7 *Use Case Description* Menyetujui Permintaan *Tooling*

Nama *Use Case*	Menyetujui Permintaan *Tooling*
Aktor	Bagian Gudang
Deskripsi	*Use case* ini menggambarkan aktor menyetujui permintaan *tooling*
Normal Flow of Events	1. Aktor memilih *approval* 2. Aktor mengklik tombol *approval* atau *reject*
Alternate Flow	

Sumber: Hasil Analisis, 2018

7. *Use Case Description* Membuat Laporan Permintaan *Tooling*

Tabel V.8 *Use Case Description* Membuat Laporan Permintaan *Tooling*

Nama *Use Case*	Membuat Laporan Permintaan *Tooling*
Aktor	Bagian Gudang
Deskripsi	*Use case* ini menggambarkan bagian gudang membuat laporan permintaan *tooling*.
Normal Flow of Events	1. Aktor memilih *reports* 2. Aktor memilih tanggal permintaan 3. Sistem akan menampilkan data permintaan 4. Aktor mengklik tombol cetak untuk mencetak data permintaan.
Alternate Flow	

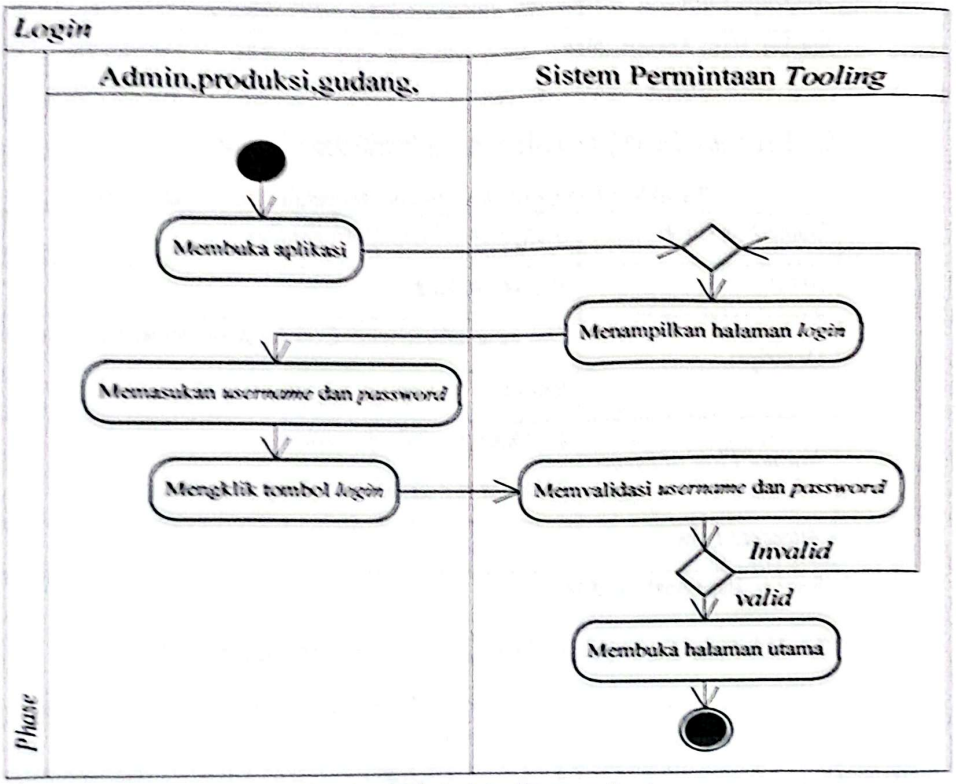
Sumber: Hasil Analisis, 2018

Dipindai dengan CamScanner

5.3.2 *Activity Diagram*

Activity Diagram merupakan sebuah diagram yang menggambarkan urutan aktivitas proses bisnis pada suatu sistem, sebagai bantuan dalam proses memahami proses bisnis yang berjalan pada sistem permintaan *tooling engine*. Berikut merupakan *activity diagram* usulan dari sistem informasi permintaan *tooling engine* di PT Mitsubishi Kramayudha:

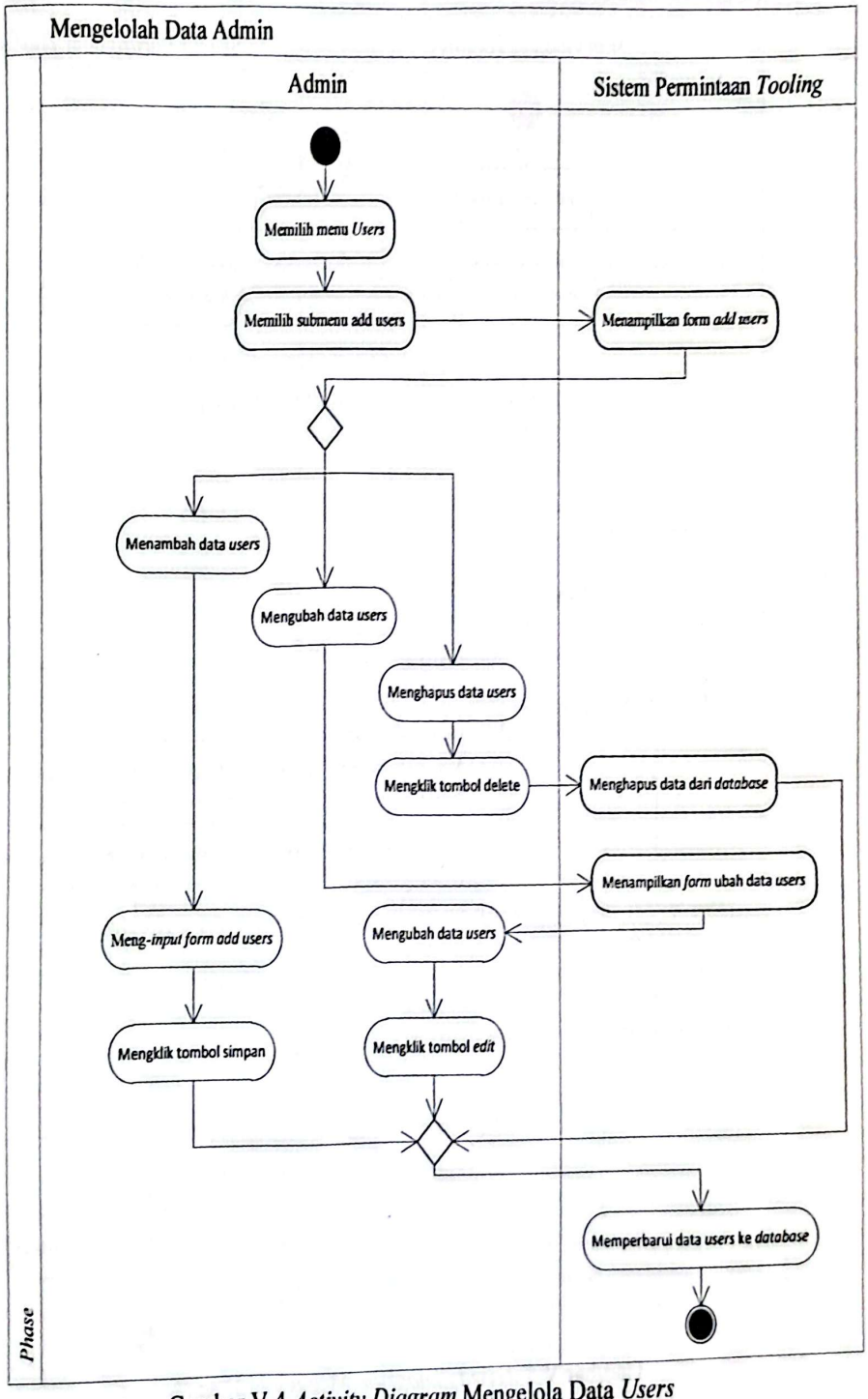
1. *Activity Diagram* Melakukan *Login*



Gambar V.3 *Activity Diagram Login*
(Sumber: Hasil Analisis, 2018)

Dipindai dengan CamScanner

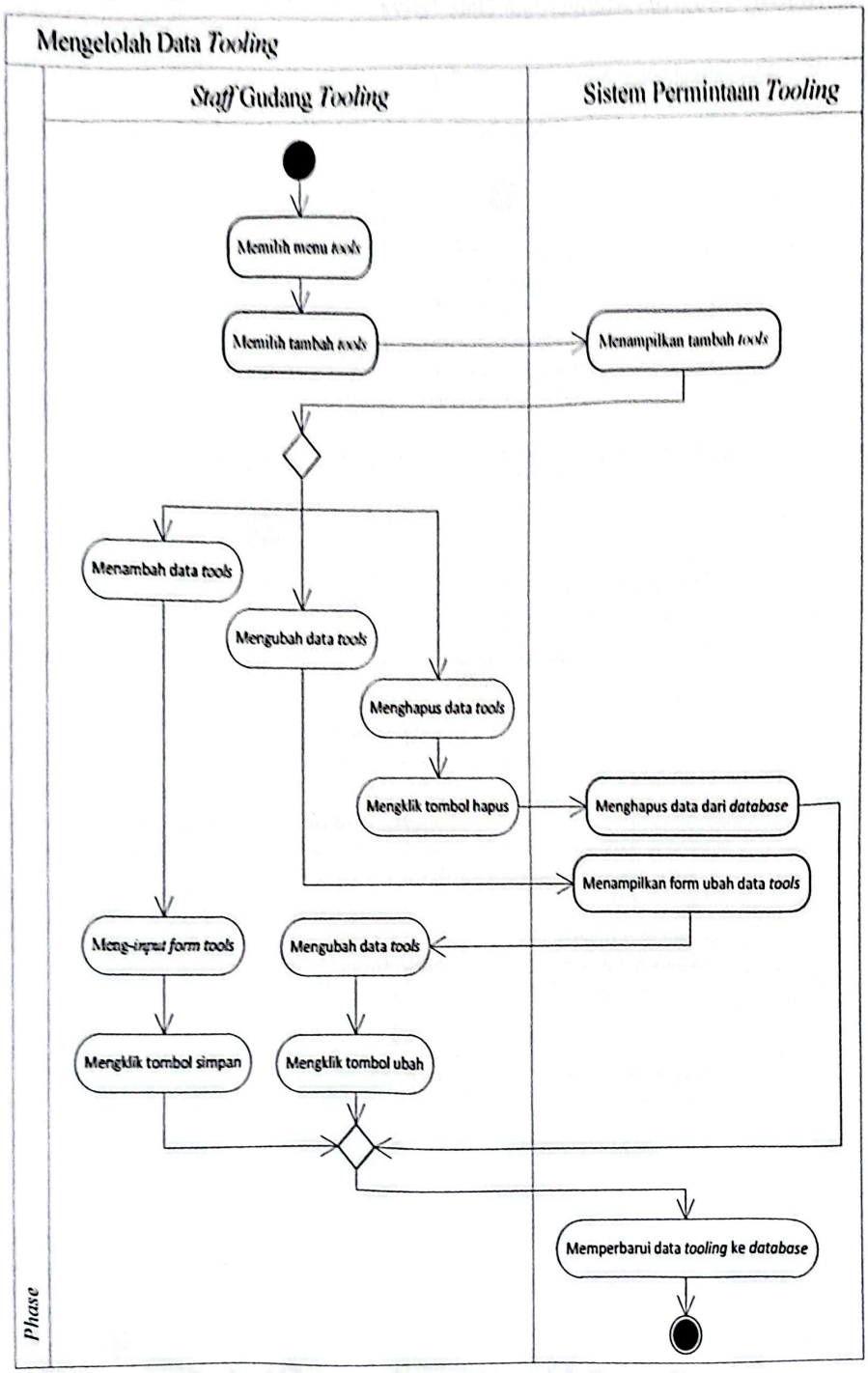
2. *Activity Diagram* Mengelolah Data *Users*



Gambar V.4 *Activity Diagram* Mengelola Data *Users*
(Sumber: Hasil Analisis, 2018)

Dipindai dengan CamScanner

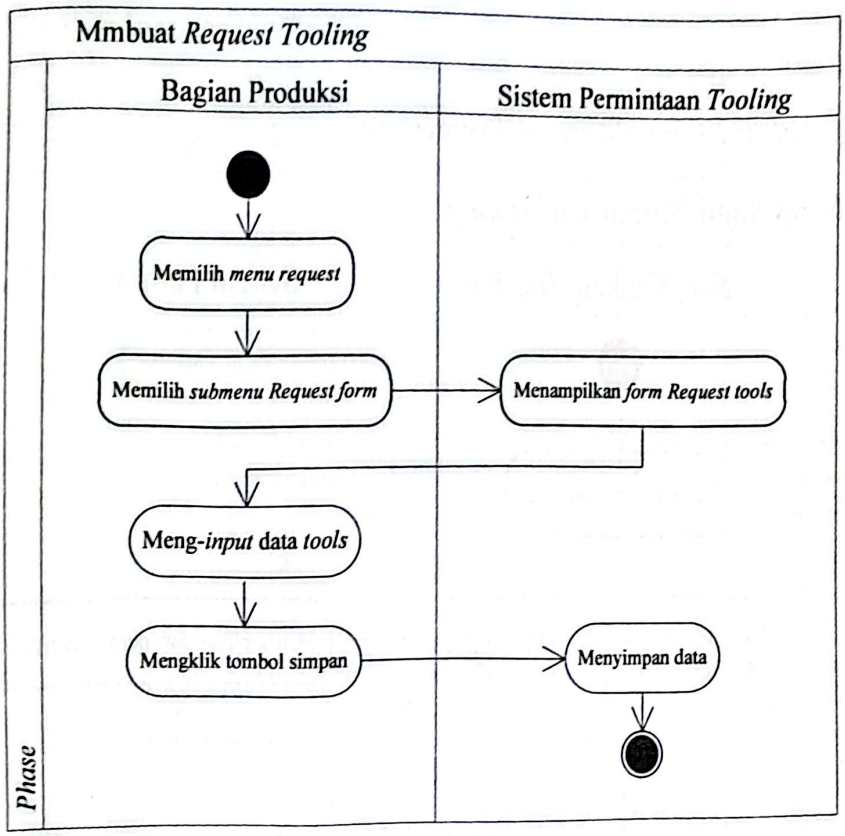
3. *Activity Diagram* Mengelolah Data *Tooling*



Gambar V.5 *Activity Diagram* Mengelola Data *tooling*
(Sumber: Hasil Analisis, 2018)

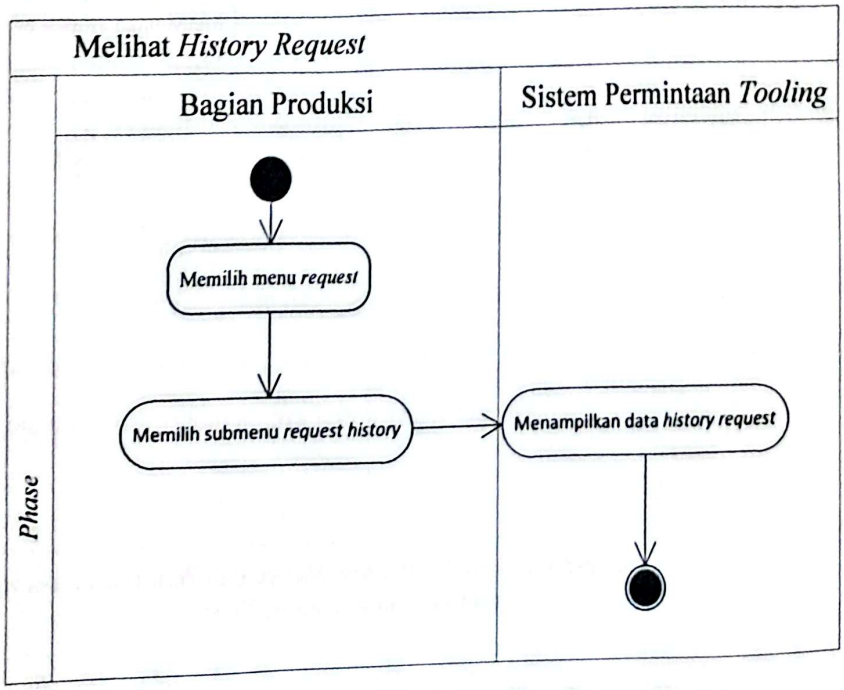
4. *Activity Diagram* Membuat *Request Tooling*

Dipindai dengan CamScanner



Gambar V.6 *Activity Diagram* Membuat *Request Tooling*
(Sumber: Hasil Analisis, 2018)

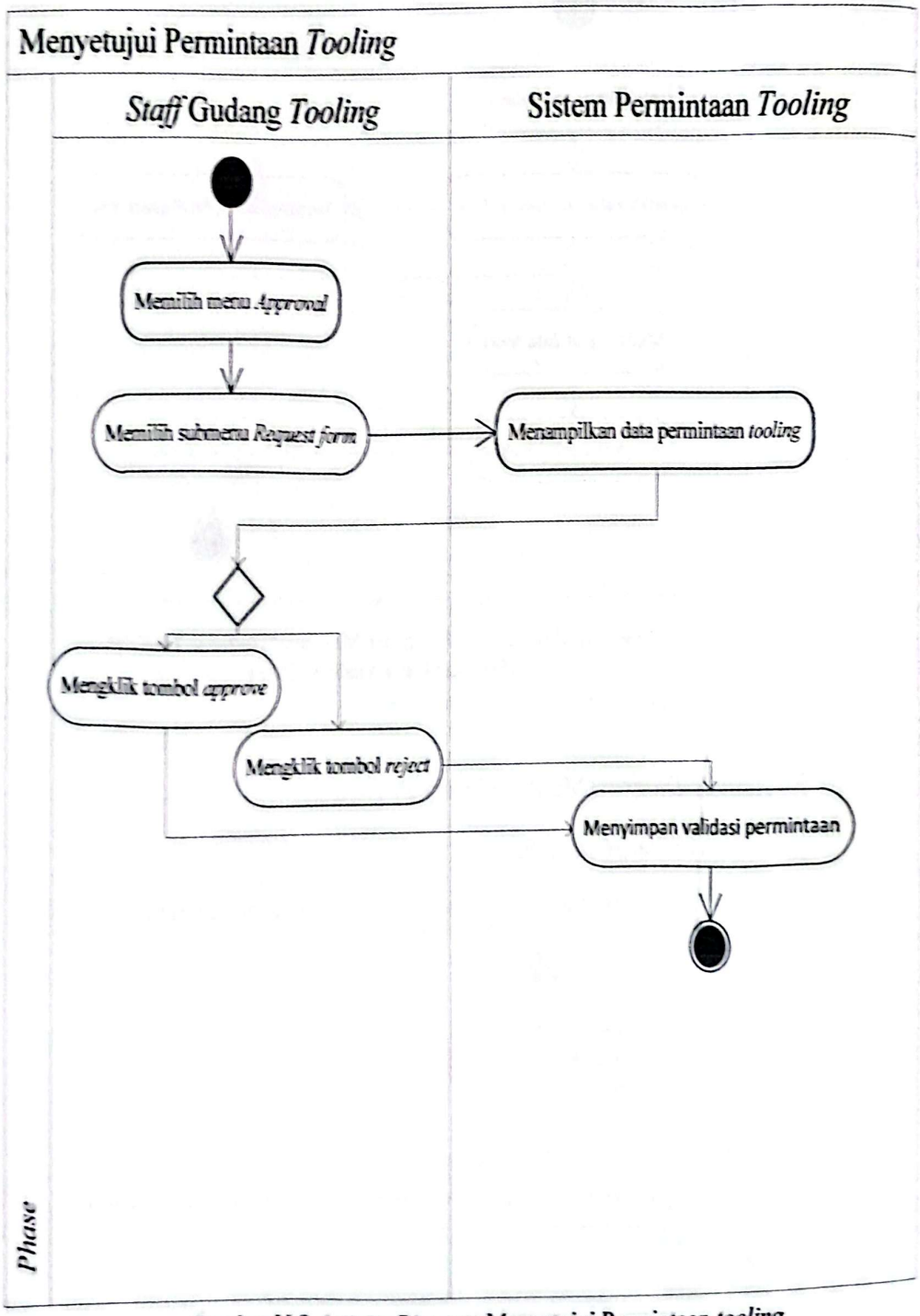
5. *Activity Diagram* Melihat *History Request*



Gambar V.7 *Activity Diagram* Melihat *Request History*
(Sumber: Hasil Analisis, 2018)

Dipindai dengan CamScanner

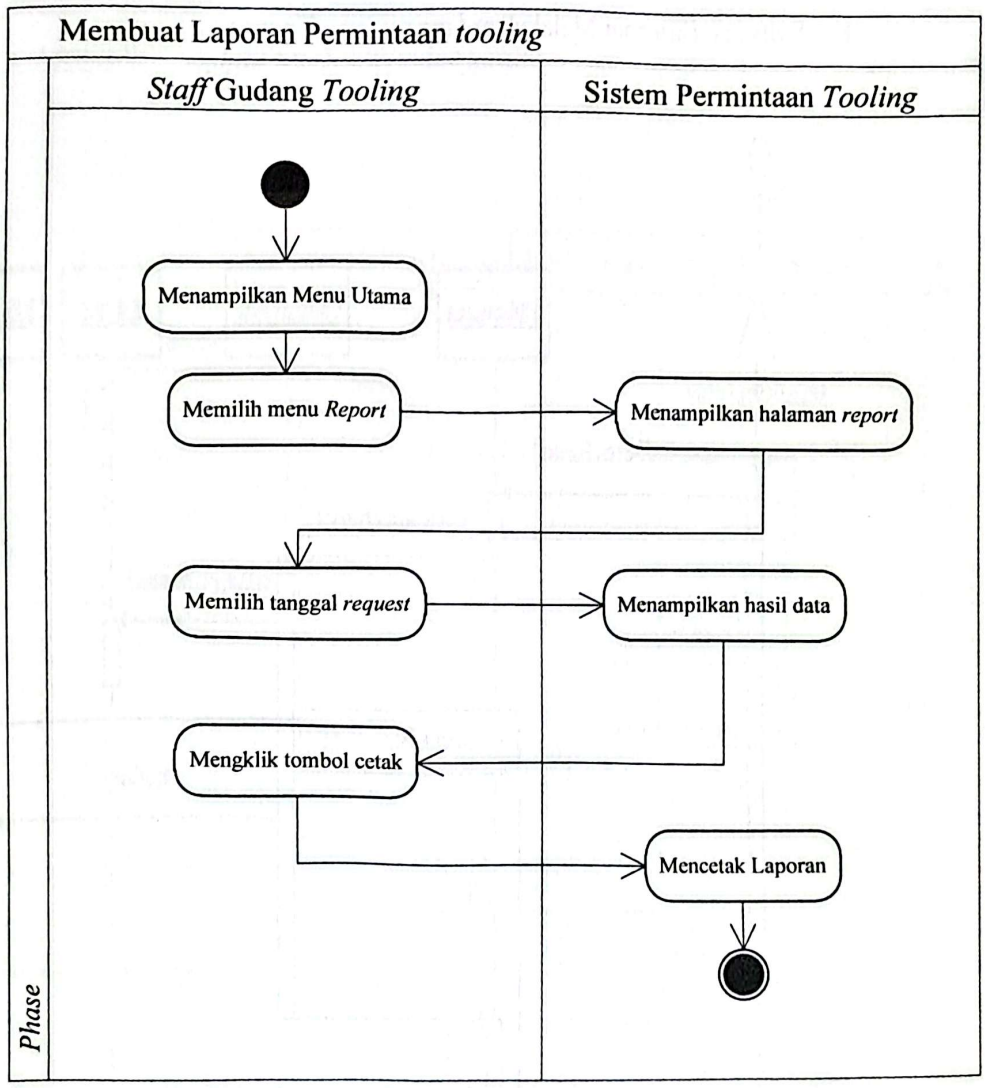
6. *Activity Diagram* Menyetujui permintaan *tooling*



Gambar V.8 *Activity Diagram* Menyetujui Permintaan *tooling*
(Sumber: Hasil Analisis, 2018)

Dipindai dengan CamScanner

7. *Activity Diagram* Membuat laporan permintaan



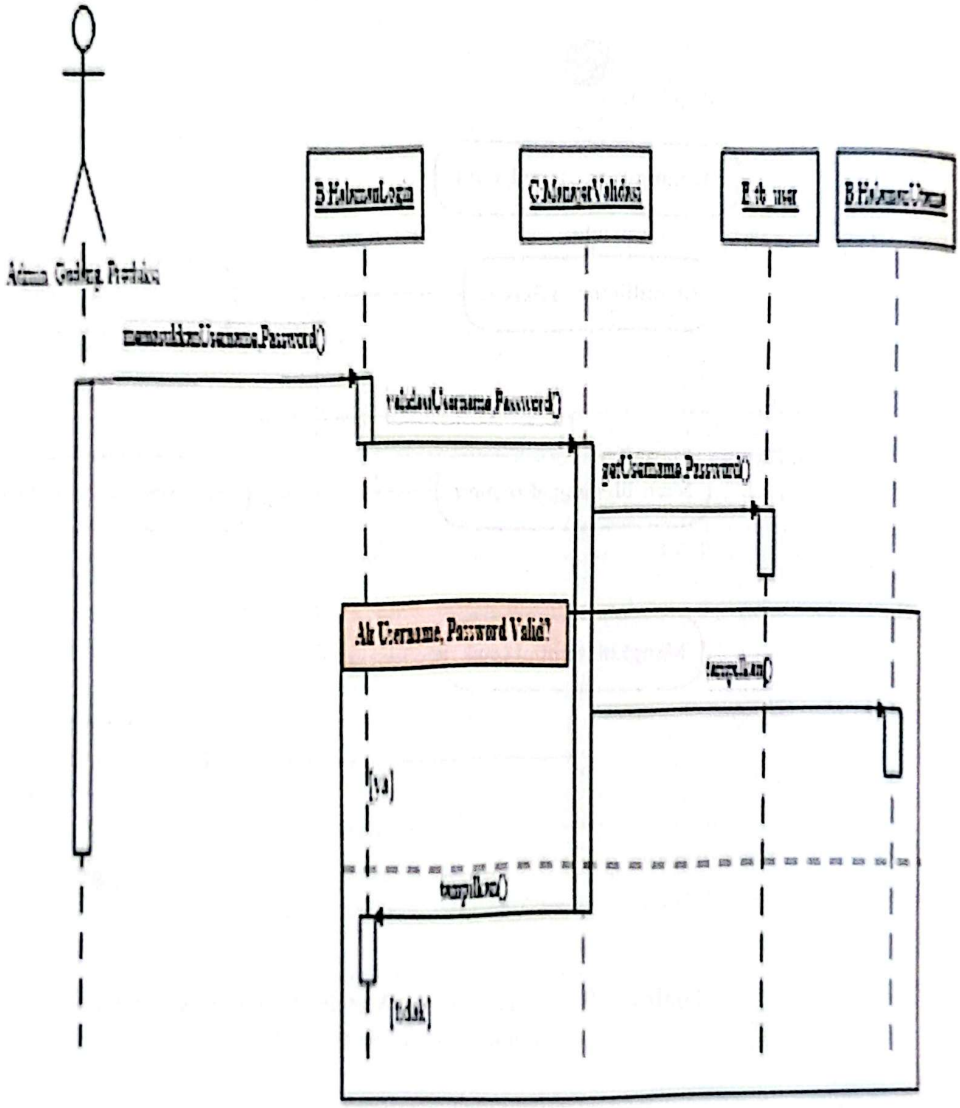
Gambar V.9 *Activity Diagram* Membuat laporan permintaan *tooling*
(Sumber: Hasil Analisis, 2018)

5.3.3 *Sequence Diagram*

Sequence Diagram menggambarkan interaksi yang terjadi antar objek pada suatu *Use Case Diagram*. *Sequence Diagram* sistem informasi permintaan *tooling engine* yang diusulkan antara lain sebagai berikut:

Dipindai dengan CamScanner

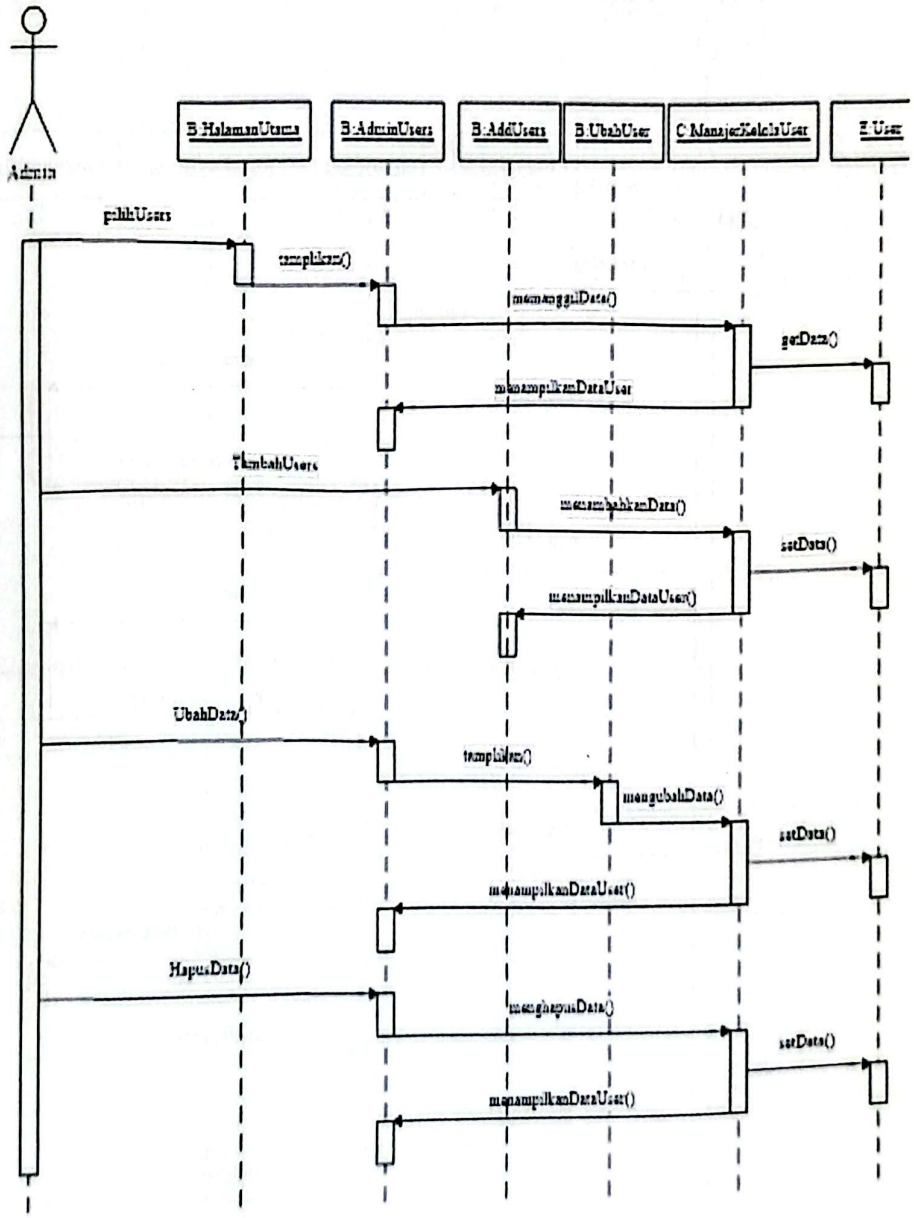
1. *Sequence Diagram* Melakukan *Login*



Gambar V.10 *Sequence Diagram* Melakukan *Login*
(Sumber: Hasil Analisis, 2018)

Dipindai dengan CamScanner

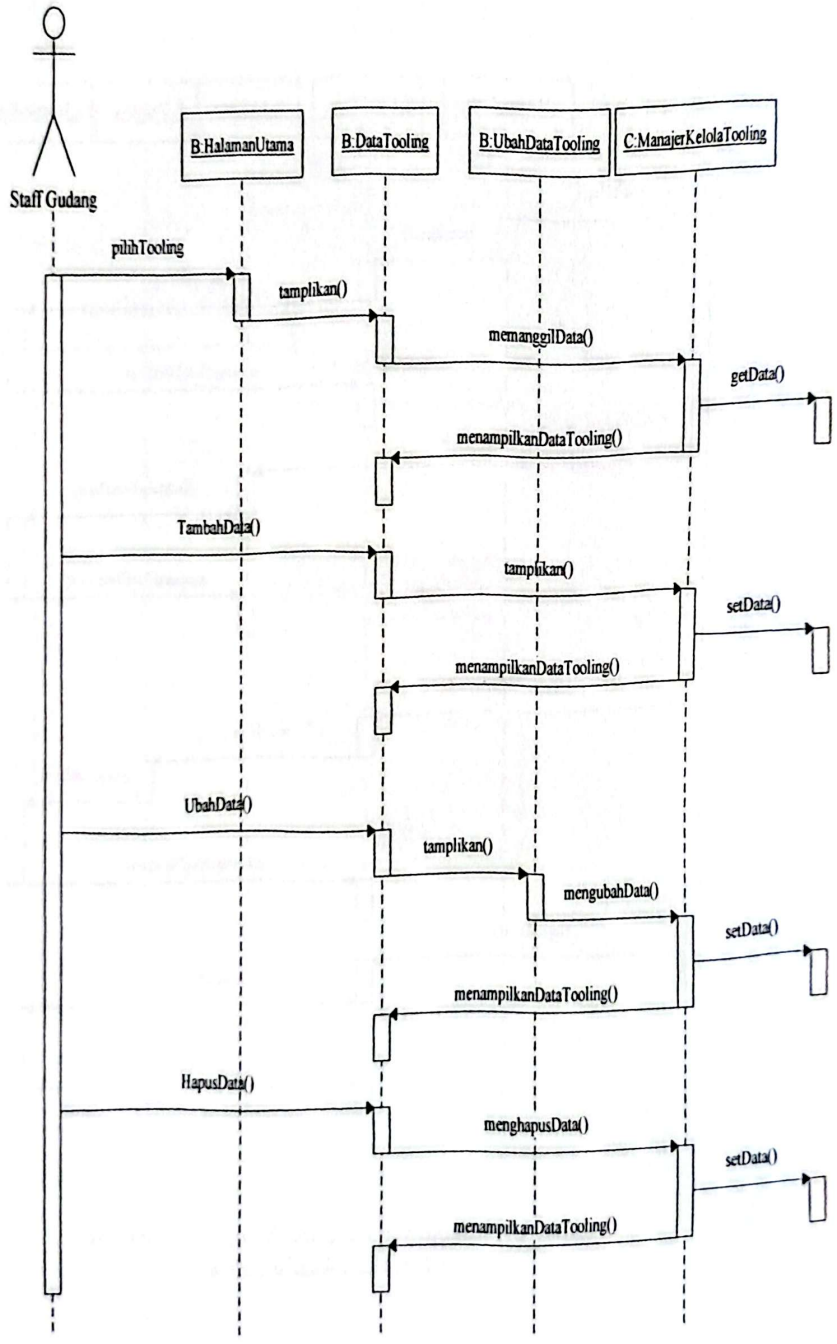
2. *Sequence Diagram* Mengelola data *Users*



Gambar V.11 *Sequence Diagram* Mengelola data *Users*
(Sumber: Hasil Analisis, 2018)

Dipindai dengan CamScanner

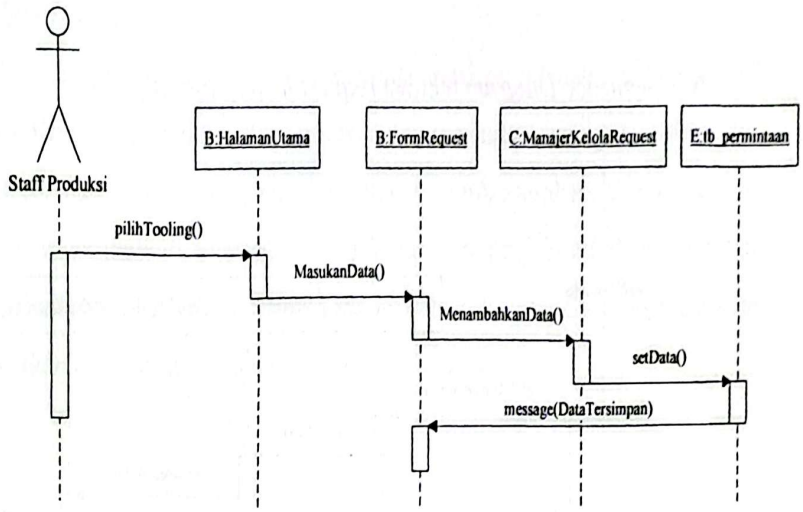
3. *Sequence Diagram* Mengelolah Data *Tooling*



Gambar V.12 *Sequence Diagram* Mengelolah Data *Tooling*
(Sumber: Hasil Analisis, 2018)

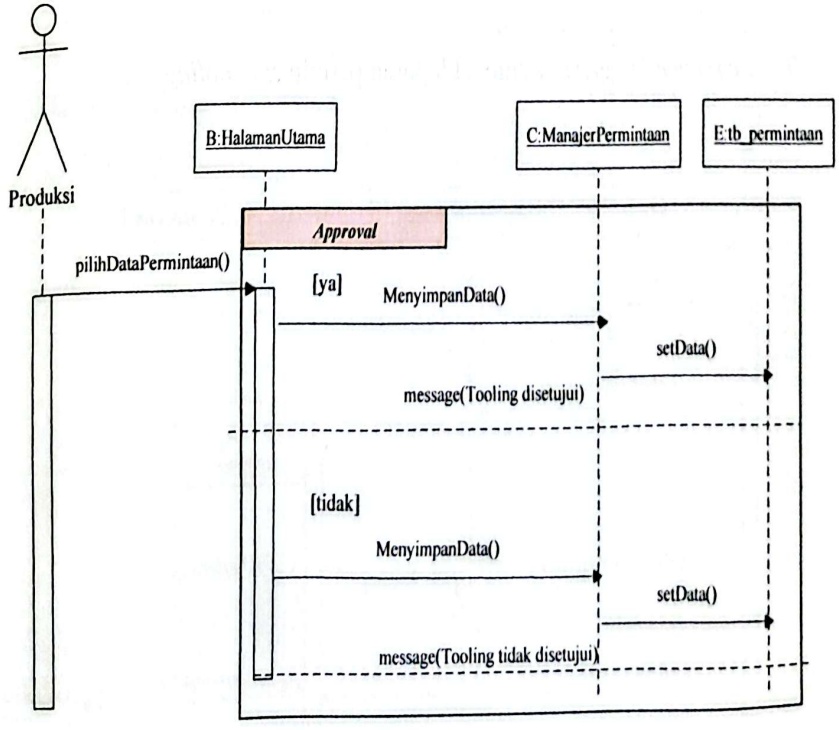
Dipindai dengan CamScanner

4. *Sequence Diagram* Membuat *Request tooling*



Gambar V.13 *Sequence Diagram* Membuat *Request tooling*
(Sumber: Hasil Analisis, 2018)

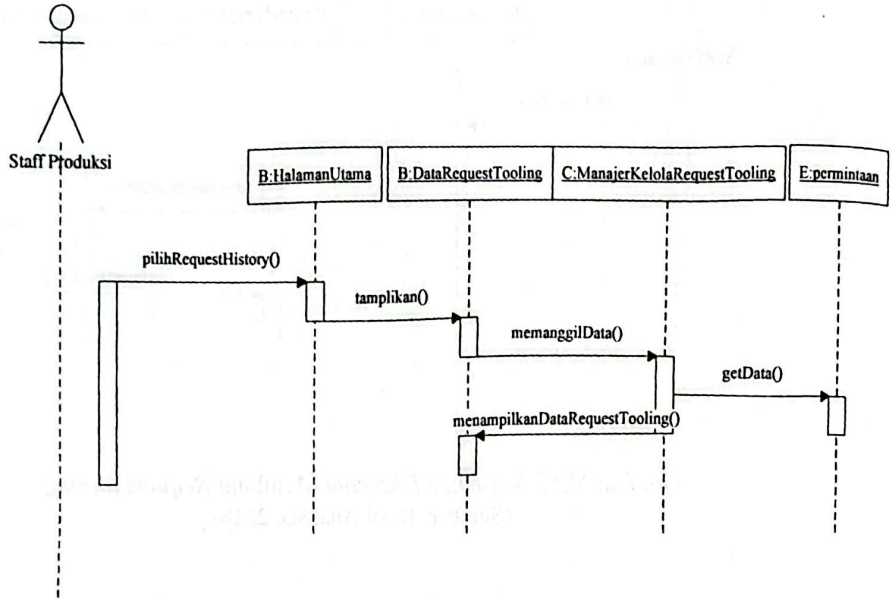
5. *Sequence Diagram* Menyetujui Permintaan *tooling*



Gambar V.14 *Sequence Diagram* Menyetujui Permintaan *tooling*
(Sumber: Hasil Analisis, 2018)

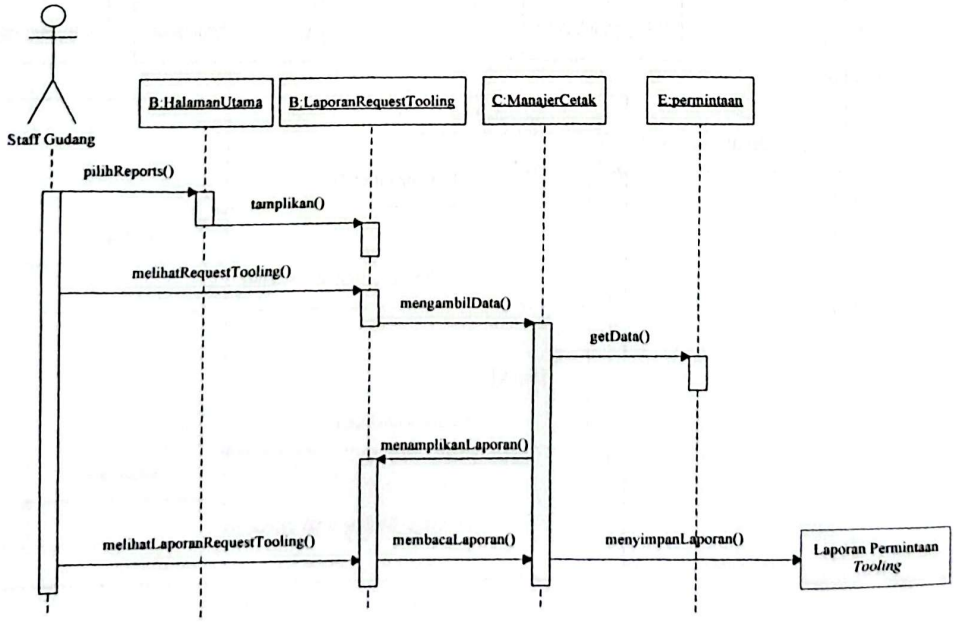
Dipindai dengan CamScanner

6. *Sequence Diagram* melihat *request history tooling*



Gambar V.15 *Sequence Diagram Request History*
(Sumber: Hasil Analisis, 2018)

7. *Sequence Diagram* Membuat laporan permintaan *tooling*



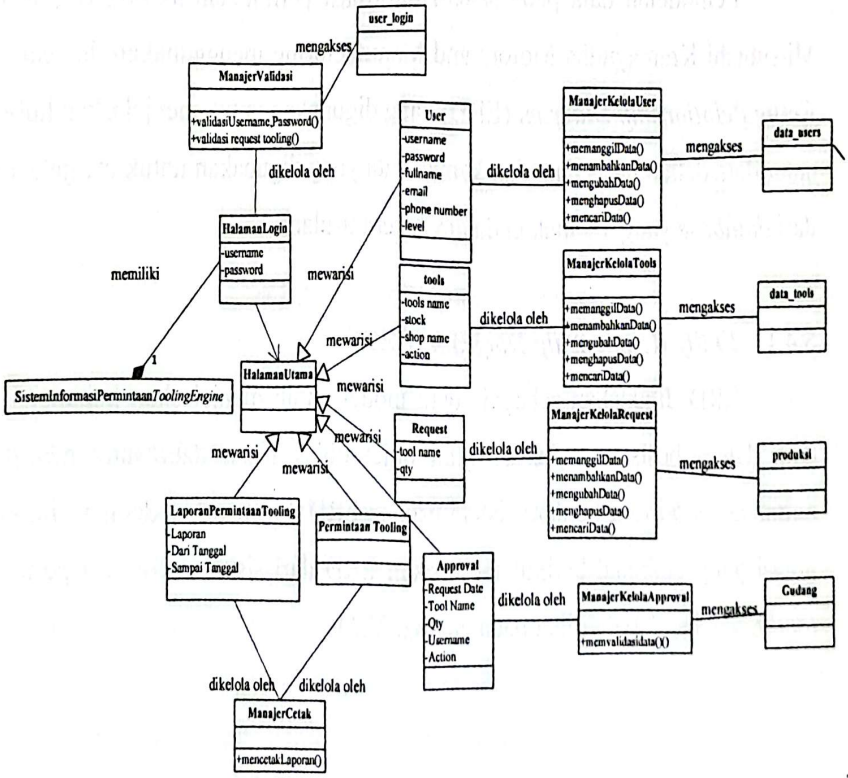
Gambar V.16 *Sequence Diagram* Membuat laporan permintaan *tooling*

(Sumber: Hasil Analisis, 2018)

Dipindai dengan CamScanner

5.3.4 *Class Diagram*

Class Diagram pada usulan sistem informasi permintaan *tooling* di PT Mitsubishi Kramayudha digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat, untuk membangun sistem *class diagram* memperlihatkan hubungan antar kelas dan penjelasan *detail* tiap-tiap kelas di dalam model desain dari suatu sistem, berikut merupakan *class diagram* yang diusulkan dapat dilihat pada Gambar V.21 berikut:

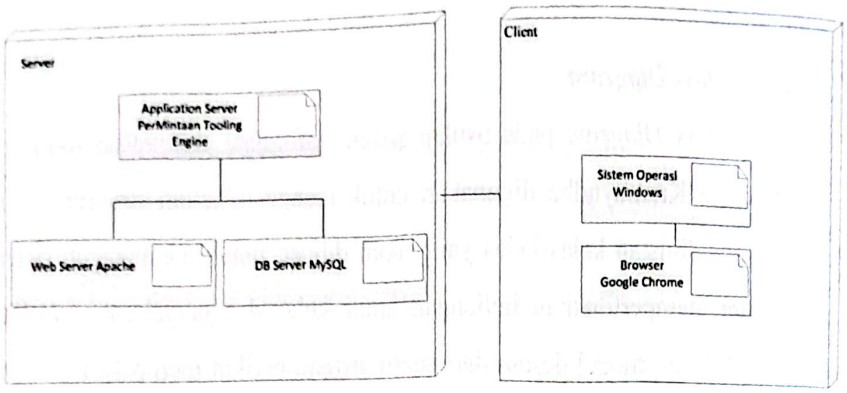


Gambar V.17 *Class Diagram* Sistem Informasi Usulan
(Sumber: Hasil Analisis, 2018)

5.3.5 *Deployment Diagram*

Deployment Diagram sistem informasi permintaan *tooling engine* yang diusulkan dapat dilihat pada Gambar V.22 berikut:

Dipindai dengan CamScanner



Gambar V.18 *Deployment Diagram* Sistem Informasi Usulan
(Sumber: Hasil Analisis, 2018)

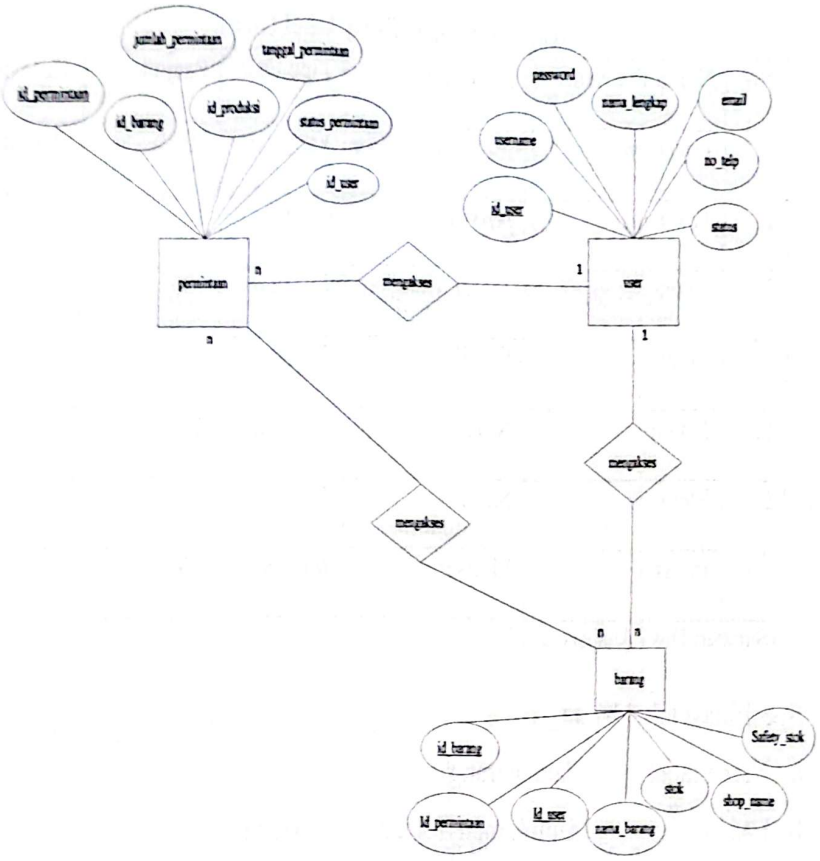
5.4 Pemodelan Data Sistem Usulan

Pemodelan data pada sistem informasi permintaan *tooling engine* di PT Mitsubishi Kramayudha Motors and Manufacturing menggunakan dua cara yaitu *Entity Relationship Diagram* (ERD) yang digunakan untuk menjelaskan hubungan antar data dalam basis data dan kamus data yang digunakan untuk menjelaskan isi dari *database* yang digunakan dalam sistem usulan.

5.4.1 *Entity Relationship Diagram* (ERD)

ERD digunakan sebagai suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi dalam sistem usulan. ERD digambarkan dengan simbol dan notasi yang berbeda, berikut merupakan ERD dari sistem informasi permintaan *tooling engine* dapat dilihat pada gambar V.23:

Dipindai dengan CamScanner



Gambar V.19 *Entity Relationship Diagram*
(Sumber: Hasil Analisis, 2018)

5.4.2 Kamus Data

Kamus menggambarkan sebuah katalog data yang terdapat didalam sistem dengan maksud untuk mendefinisikan data yang mengalir di dalam sistem dengan lengkap sehingga user dan analis sistem mempunyai pengertian yang sama tentang *input, output*, dan komponen *data store*. Pada tahap perancangan sistem, kamus data digunakan untuk merancang *input,* laporan dan basis data. Berikut adalah kamus data sistem informasi permintaan *tooling engine*:

1. Spesifikasi tabel *user*
   a. Nama tabel : user_login
   b. Fungsi : Untuk menyimpan data *user*/pengguna
   c. Tipe : File data *master*

Dipindai dengan CamScanner

Tabel V.9 Spesifikasi Tabel *User*

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	*Username*	*username*	*Varchar*	20	*Primary key*
2	*Password*	*password*	*Char*	40	
3.	Nama_lengkap	Nama_Lengkap	*Varchar*	100	
4.	Email	Email	*Varchar*	100	
5.	Notelp	Notelp	*Varchar*	15	
6	Status	Status	*Varchar*	25	
7	Id_user	Id_user	*Integer*	30	

(Sumber: Hasil Analisis, 2018)

2. Spesifikasi tabel barang

   a. Nama tabel : data_barang

   b. Fungsi : Untuk menyimpan data *tooling*

   c. Tipe : File data *master*

Tabel V.10 Spesifikasi Tabel Barang

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Id_barang	Id_barang	*Integer*	16	*Primary key*
2	Nama_barang	Nama_barang	*Varchar*	40	
3	Stok	Stok	*Integer*	11	
4	Safety_stok	Safety_stok	*Integer*	11	
5	Shop name	Shop_name	*Varchar*	10	
6	Tool_No	Tool_No	*Varchar*	50	
7	Op_NO	Op_No	*Varchar*	50	
8	Id_gudang	Id_gudang	*Integer*	11	

(Sumber: Hasil Analisis, 2018)

3. Spesifikasi tabel Permintaan Header

   a. Nama tabel : Permintaan Header

Dipindai dengan CamScanner

b. Fungsi : Untuk menyimpan data permintaan

c. Tipe : File data transaksi

Tabel V.11 Spesifikasi Tabel Permintaan Header

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	id Permintaan	id_permintaan	*Integer*	7	*Primary Key*
2	Tanggal Permintaan	tanggal_permintaan	*Integer*	7	
3	Status Permintaan	status_permintaan	*Varchar*	10	

(Sumber: Hasil Analisis, 2018)

4. Spesifikasi tabel Permintaan Detail

d. Nama tabel : Permintaan Detail

e. Fungsi : Untuk menyimpan data permintaan

f. Tipe : File data transaksi

Tabel V.12 Spesifikasi Tabel Permintaan Detail

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Id Permintaan detail	id_permintaan_detail	*Integer*	7	*Primary Key*
2	Id Permintaan	id_permintaan	*Integer*	7	*Foreign Key*
3	Id barang	id_barang	*Integer*	10	*Foreign Key*
4	Jumlah Permintaan	jumlah_permintaan	*Integer*	100	

(Sumber: Hasil Analisis, 2018)

5.5 Perancangan Sistem

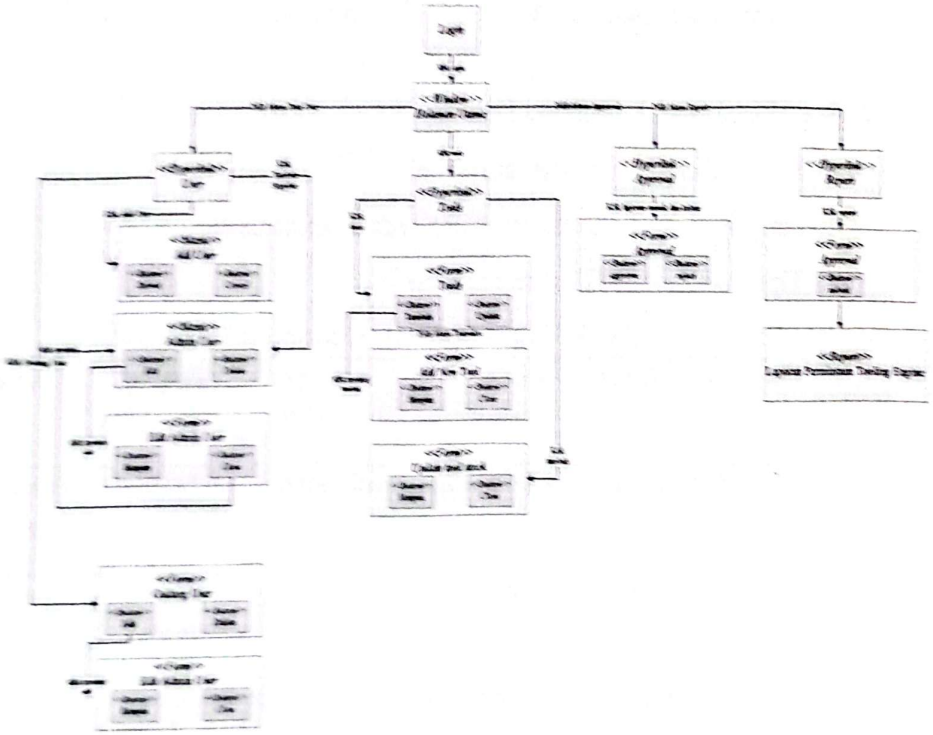
Perancangan sistem pada sistem usulan digunakan sebagai ilustrasi yang menggambarkan bagaimana aplikasi yang dibuat berjalan dan juga

Dipindai dengan CamScanner

menggambarkan navigasi dari aplikasi tersebut dengan *tools Windows Navigation Diagram* (WND) serta menggambarkan tampilan antar muka dari aplikasi.

5.5.1 *Windows Navigation Diagram* (WND)

Windows Navigation Diagram pada sistem usulan digunakan untuk menunjukan bagaimana navigasi dari halaman-halaman yang terdapat di dalam aplikasi. Di WND setiap status antarmuka pengguna direpresentasikan sebagai kotak. Kotak biasanya sesuai dengan komponen antarmuka pengguna, seperti jendela, *form*, tombol, atau laporan, berikut merupakan *Windows Navigation Diagram* sistem permintaan *tooling engine* dapat dilihat pada gambar V.24:



Gambar V.20 Windows Navigation Diagram
(Sumber: Hasil Analisis, 2018)

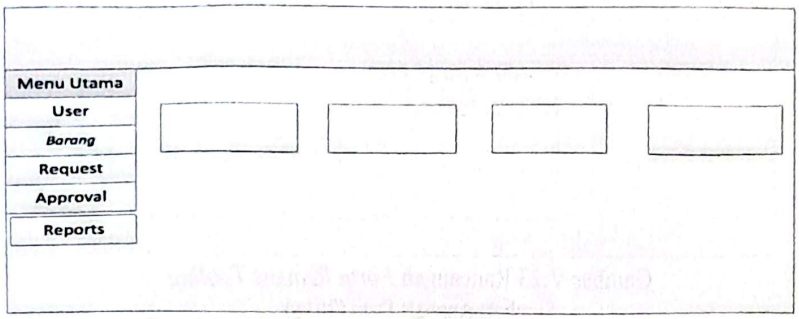
5.2 Perancangan Antarmuka Pengguna

Perancangan antarmuka dalam sistem usulan merupakan penggambaran antarmuka aplikasi yang hendak dibuat, antarmuka merupakan bagian yang

Dipindai dengan CamScanner

berhubungan langsung dengan pengguna, berikut nerupakan rancangan antarmuka sistem informasi permintaan *tooling engine:*

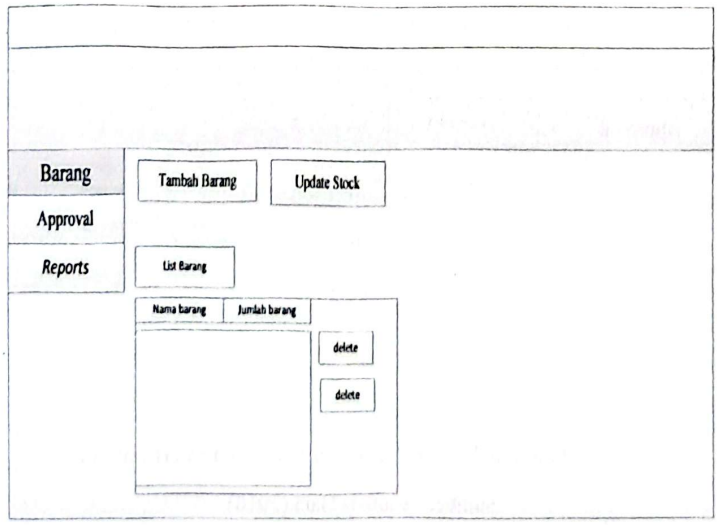
1. *Form* Halaman Utama



Gambar V.21 Rancangan *Form* Halaman Utama
Sumber: Analisis Data (2018)

2. Halaman Data *Tooling*

Form data barang adalah *form* yang digunakan untuk mengelola dan menampilkan *tooling*. *Form* ini dapat dilihat pada gambar V.22.

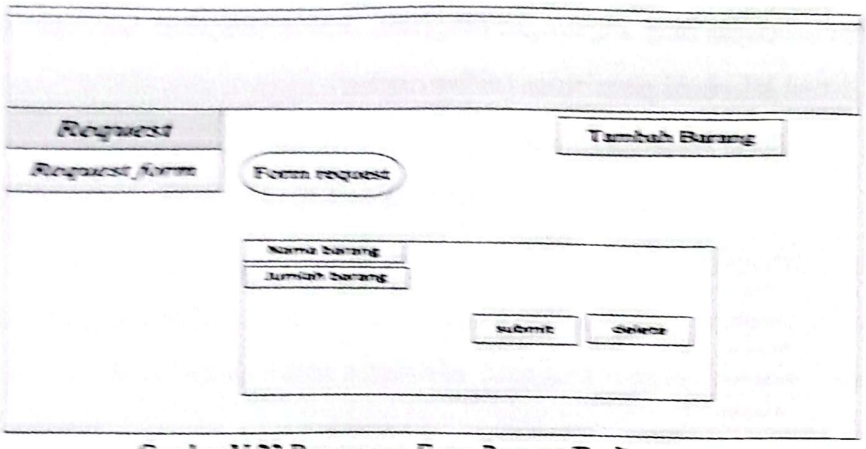


Gambar V.22 Rancangan *Form Data Tooling*
Sumber: Analisis Data (2018)

3. *Form Request*

Form ini digunakan untuk meminta barang dari produksi ke bagian gudang. *Form* ini dapat dilihat pada gambar V.23.

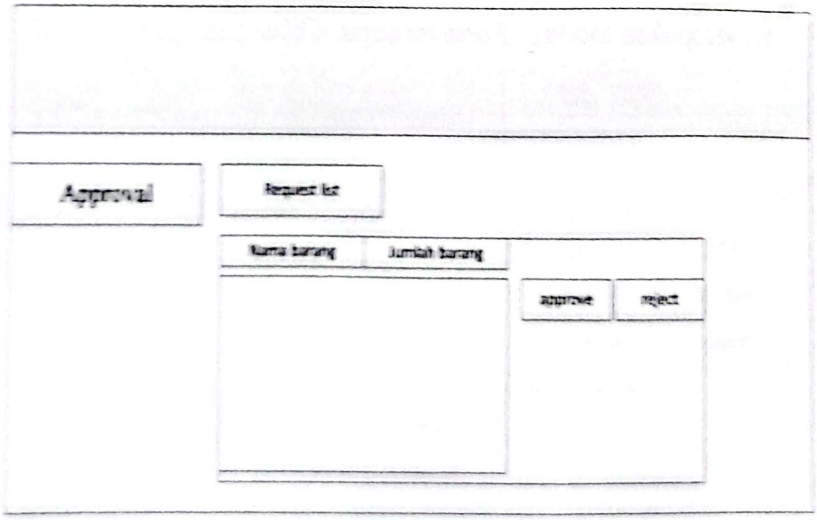
Dipindai dengan CamScanner



Gambar V.23 Rancangan *Form Request Tooling*
Sumber: Analisis Data (2018)

4. *Form* Data Data Permintaan

Form data permintaan untuk melihat data permintaan dan untuk menyetujui permintan. *Form* ini dapat dilihat pada gambar V.24.

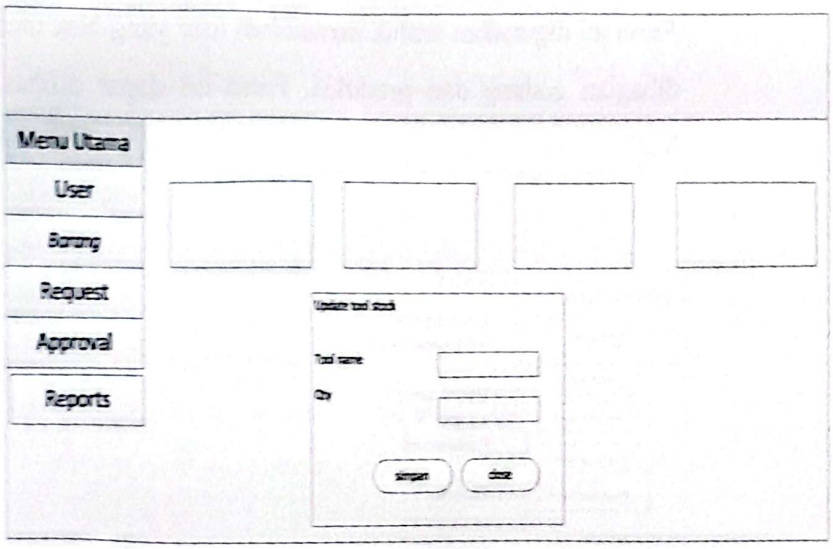


Gambar V.24 Rancangan *Form* Data Permintaan
Sumber: Analisis Data (2018)

5. *From Update Stock*

Form untuk mengupdate stock barang yang tersedia. *Form* ini dapat dilihat pada gambar V.25.

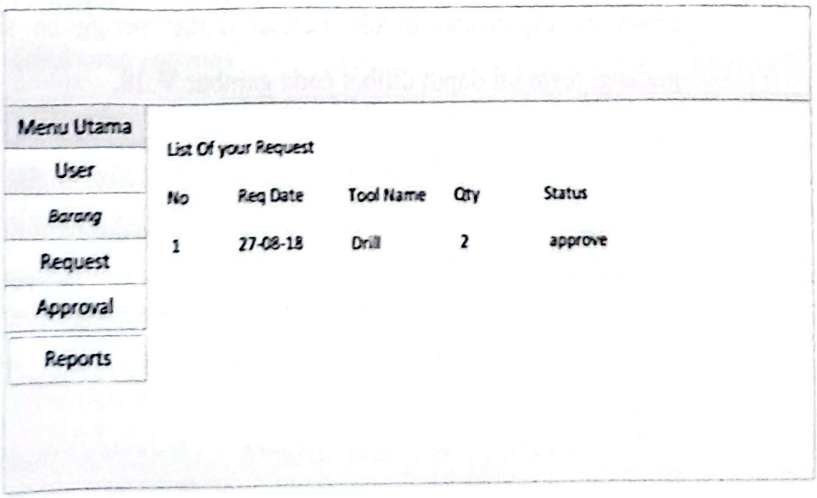
Dipindai dengan CamScanner



Gambar V.24 Rancangan *Form Update Stock*
Sumber: Analisis Data (2018)

6. *Form Request Story*

Form ini digunakan untuk melihat ststus permintaan barang apakah di aaprove oleh bagian gudang apa tidak. *Form* ini dapat dilihat pada gambar V.26.

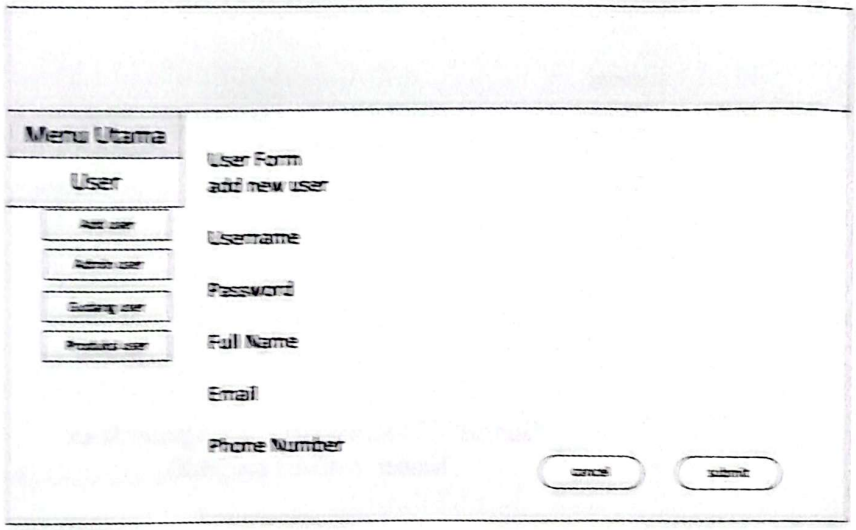


Gambar V.26 Rancangan *Form Request Story*
Sumber: Analisis Data (2018)

7. *Form* tambah *user*

Dipindai dengan CamScanner

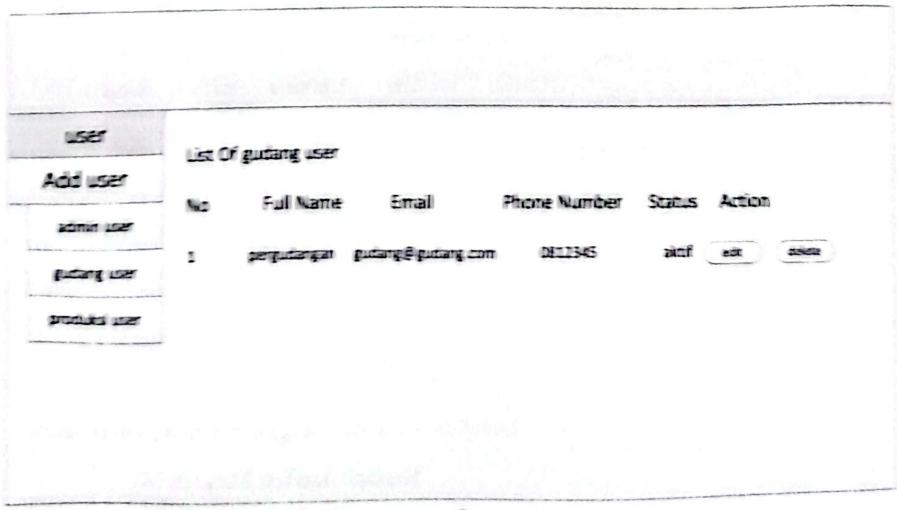
Form ini digunakan untuk menambah *user* yang bisa mengakses sistem dibagian gudang dan produksi. Form ini dapat dilihat pada gambar V.27.



Gambar V.27 Rancangan *form* tambah *user*
Sumber: Analisis Data (2018)

8. *Form* Gudang *User*

Form ini digunakan untuk melihat daftar pengguna sistem di bagian gudang. *form* ini dapat dilihat pada gambar V.28.

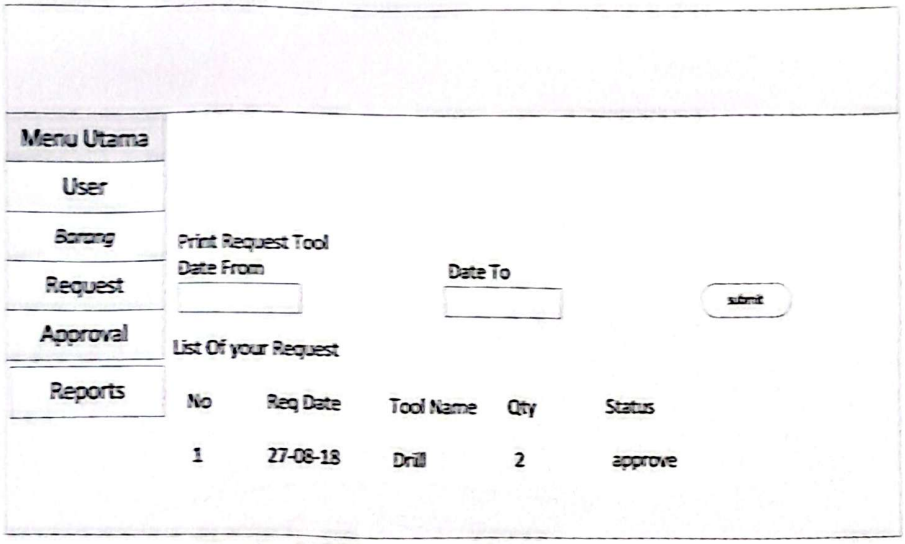


Gambar V.28 Rancangan *form* gudang *user*
Sumber: Analisis Data (2018)

9. *Form Reports*

Form ini digunakan untuk melihat laporan permintaan

Dipindai dengan CamScanner



Gambar V.29 Rancangan *form reports*
Sumber: Analisis Data (2018)

5.6 Implementasi Sistem *Software* dan *Hardware*

Untuk dapat menggunakan aplikasi, tentunya diperlukan suatu spesifikasi perangkat keras *(hardware)* ataupun perangkat lunak *(software)* yang mendukung agar aplikasi dapat berjalan dengan baik. Adapun spesifikasinya sebagai berikut:

1. Analisis kebutuhan *software*
   a. Sistem operasi : Minimal *Microsoft Windows* 7.
   b. *Database server* : MySQL versi 5.5.32.
   c. Bahasa pemrograman : Codeigniter.
   d. *Web Server* : *Apache.*
   e. *Web Browser* : *Mozilla Firefox, Google Chrome, Microsoft Edge*
2. Analisis kebutuhan *hardware*
   a. *Processor* : Minimal *processor pentium* IV.
   b. RAM : Minimal RAM 1024 MB.
   c. *Harddisk* : Minimal 100 GB.
   d. Keyboard, mouse, layar monitor.
   e. Printer sebagai media *output.*
   f. Wifi sebagai penghubung jaringan.

Dipindai dengan CamScanner

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil rancang bangun sistem yang dilakukan mengenai sistem informasi permintaan *tooling engine* di PT Mitsubishi Krama yudha Motors and Manufacturing dapat disimpulkan beberapa hal yaitu:

1. Membuat sistem basis data untuk permintaan *tooling engine* sehingga memudahkan dalam pencarian data dan mencegah terjadinya kehilangan data.
2. Merancang bangun sistem dengan menerapkan basis data agar memudahkan staff dalam memproses data permintaan *tooling*.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem informasi permintaan ini selanjutnya adalah sebagai berikut:

1. Untuk penerapan sistem baru terhadap sistem lama, sebaiknya dilakukan secara bertahap dan di perlukan sosialisasi kepada Bagian-bagian yamg terkait yang akan menggunakan sistem ini.
2. Sebaiknya dilakukan pemeliharaan aplikasi secara berkala sehingga aplikasi dapat berjalan dengan baik.
3. Diharapka bagi pembaca tugas akhir ini dapat membuat sistem informasi permintaan *tooling engine* menjadi sistem yang lebih baik dengan menambahkan beberapa fungsi tambahan, seperti penambahan fungsi *barcode* untuk meningkatkan kinerja pegawai dalam penginptan permintaan tooling engine

Dipindai dengan CamScanner

DAFTAR PUSTAKA

-----, Arsip PT Mitsubishi Krama Yudha Motors and Manufacturing

Bodnar, H. George & William S. Hopwood. 2000. *Sistem Informasi Akuntansi.* Yogyakarta: Salemba Empat.

Hariyanto, Bambang. 2004. Rekayasa Sistem Berorientasi Objek. Jakarta : Informatika Bandung.

Hartono, Jogiyanto. 1999. *Analisa dan Desain Sistem Informasi: Pendekatan terstrktur teori dan praktis aplikasi bisnis.* Yogyakarta: Andi.

Jerry. 1981. *Fundamentals Of System Analysis.* New York: Edisi kedua, John Willey & Sons.

Jogiyanto. 2005. *Sistem Teknologi Informasi.* Yogyakarta: Andi.

Jogiyanto. 2005. Analisis & Desain Sistem Informasi: Pendekatan Terstruktur

Murdick, dkk. 1991. *Sistem Informasi Untuk Manajemen Modern.* Jakarta: Erlangga.

Mulyadi. 2001. *Sistem Akuntansi.* Edisi Ketiga. Cetakan Ketiga. Jakarta: Salemba Empat.

Nash, John F. 2003. *Sistem Informasi Akuntansi.* Bandung: Lembaga Informatika Akuntansi.

Scott M George. 2004. *Principles of Management Information Systems.* McGraw-Hill.

Simatupang, T. M. 1994. *Pemodelan Sistem.* Nindita, Klaten.

Steers R. M dan Porter L. W, 1987, *Motivation and Leadership at Work.* New York: McGraw Hill.

Stoner & Freeman. (1996). *Manajemen.* Jakarta: Prenhalindo.

Sutabri, Tata. 2012. *Sistem Informasi Manajemen.* Yogyakarta: Andi.

Dipindai dengan CamScanner

----------. *Kamus Besar Bahasa Indonesia.* [Online]. Tersedia di: https://kbbi.kemdikbud.go.id/. Diakses 18 Febuari 2018.

Abadi, Zainal. 2012. *Proses Produksi Permesinan.* Bandung: CV Pustaka Setia.

Ahyari, Agus. 2011. *Perencanaan dan Pengendalian Produksi.* Yogyakarta: BPFE.

Assauri, Sofjan. 2016. *Manajemen Operasi Produksi Pencapaian Sasaran Organisasi Berkesinambungan.* Cetakan ke-2. Jakarta: Rajawali Pers.

Endang, Mulyadi dan Mardiyatmo. 2013. *Ekonomi SMP Kelas VII.* Jakarta: Yudistira.

Fahmi, Irham. 2012. *Analisis Laporan Keuangan.* Cetakan ke-2. Bandung: Alfabeta.

Heizer, Jay dan Render, Barry. 2012. *Manajemen Operasi.* Jakarta: Bumi Aksara.

Ishak. 2010. *Manajemen Produksi dan Operasi.* Yogyakarta: Graha Ilmu.

Luis, Suwardi dan Biromo, Prima A. 2010. *Step by Step in Cascading Balanced Scorecad to Functional Scorecard.* Gramedia Pustaka Utama. Jakarta

Rusdi, Nur dan Arsyad, Suyuti, M. 2017. *Pengantar Sistem Manufaktur.* Yogyakarta: Deepublish.

Pratama, I Putu Agus Eka. 2016. *Integrasi dan Migrasi Sistem.* Cetakan Pertama. Bandung: Informatika.

Sinulingga, Sukaria. 2013. *Perencanaan dan Pengendalian Produksi.* Edisi Pertama. Yogyakarta: Graha Ilmu.

Sofyan, Diana Khairani. 2013. Perencanaan dan Pengendalian Produksi. Yogyakarta: Graha Ilmu.

Dipindai dengan CamScanner

Situs :

Anonim, Pengertian Proses, http://kbbi.web.id/proses, (Tanggal Akses; 12 Juli 2018).

Dipindai dengan CamScanner

LAMPIRAN 1

SOURCE CODE PROGRAM

1. Login

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Login extends CI_Controller {

        /**
         * Index Page for this controller.
         *
         * Maps to the following URL
         *              http://example.com/index.php/welcome
         *      - or -
         *              http://example.com/index.php/welcome/index
         *      - or -
         * Since this controller is set as the default controller in
         * config/routes.php, it's displayed at http://example.com/
         *
         * So any other public methods not prefixed with an underscore
will
         * map to /index.php/welcome/<method_name>
         * @see https://codeigniter.com/user_guide/general/urls.html
         */

        public function __construct(){
                parent::__construct();
                $this->load->model('Models');

        }
```

Dipindai dengan CamScanner

```
public function index()
{
        $this->load->view('login');
}

public function logincek(){

                $username = $this->input->post('username');
                $password = $this->input->post('password');
                $level = $this->input->post('level');



                if (isset($this->session->userdata['logged_in'])) {
                        redirect('beranda');
                }else{

                        $exec = $this->Models-
>cek_login($level,$username,$password);
                        if ($exec == TRUE) {

                                $result = $this->Models-
>getdatabyusername($level,$username);

                                if ($level == 'admin') {
                                        $session_data = array(

        'id_akun'=>$result[0]->id_admin,

        'username'=>$result[0]->username,
```

Dipindai dengan CamScanner

```
            'nama'=>$result[0]->nama_lengkap,
            'email'=>$result[0]->email,
            'jenis'=>'Administrator'
        );
                        }else if($level == 'gudang'){
                            $session_data = array(
            'id_akun'=>$result[0]->id_gudang,
            'username'=>$result[0]->username,
            'nama'=>$result[0]->nama_lengkap,
            'email'=>$result[0]->email,
            'jenis'=>'Gudang'
        );
                        }else if($level == 'produksi'){
                            $session_data = array(
            'id_akun'=>$result[0]->id_produksi,
            'username'=>$result[0]->username,
            'nama'=>$result[0]->nama_lengkap,
```

Dipindai dengan CamScanner

```
                'cmail'=>$result[0]->email,

                'jenis'=>'Produksi'

        );
                                        }

                                        $this->session-
>set_userdata('logged_in',$session_data);
                                        redirect('beranda');
                                }else{
                                        $this->session-
>set_flashdata('msg','<div class="alert alert-danger" role="alert"><button
type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-
hidden="true">&times;</span></button><strong class="d-block d-sm-
inline-block-force">Gagal!</strong> Cek username / password anda
.</div>');
                                        redirect('login');
                                }
                        }


                }

                public function logout(){

                        $sess_data = array(
                                                'username'=>'',
                                                'nama'=>'',
```

Dipindai dengan CamScanner

```
                                        'jenis'=>'',
                                        'email'=>''

                                    );

                        $this->session-
>unset_userdata('logged_in',$sess_data);
                        redirect('login');
                }
}
```

2. Beranda

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Beranda extends CI_Controller {

    /**
     * Index Page for this controller.
     *
     * Maps to the following URL
     *          http://example.com/index.php/welcome
     *    - or -
     *          http://example.com/index.php/welcome/index
     *    - or -
     * Since this controller is set as the default controller in
     * config/routes.php, it's displayed at http://example.com/
     *
     * So any other public methods not prefixed with an underscore
will
     * map to /index.php/welcome/<method_name>
```

Dipindai dengan CamScanner

```
	* @see https://codeigniter.com/user_guide/general/urls.html
	*/

	public function __construct(){
		parent::__construct();
		$this->load->library('upload');
		$this->load->model('Models');
	}

	public function index()
	{
		if($this->session->userdata('logged_in')){
			// $this->load->view('index');
			$where = array('status'=>'Aktif');
			$wherebarang = array('stok < '=>'1');
			$data['total_barang'] = $this->Models-
>get('barang')->num_rows();
			$data['total_admin'] = $this->Models-
>fetch($where,'admin')->num_rows();
			$data['total_gudang'] = $this->Models-
>fetch($where,'gudang')->num_rows();
			$data['total_produksi'] = $this->Models-
>fetch($where,'produksi')->num_rows();
			$data['total_stok_habis'] = $this->Models-
>fetch($wherebarang,'barang')->num_rows();
			$this->load->view('index',$data);
		}else{
			redirect('login');
		}
	}
```

Dipindai dengan CamScanner

3. Barang

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Barang extends CI_Controller {

        /**
         * Index Page for this controller.
         *
         * Maps to the following URL
         *              http://example.com/index.php/welcome
         *      - or -
         *              http://example.com/index.php/welcome/index
         *      - or -
         * Since this controller is set as the default controller in
         * config/routes.php, it's displayed at http://example.com/
         *
         * So any other public methods not prefixed with an underscore
will
         * map to /index.php/welcome/<method_name>
         * @see https://codeigniter.com/user_guide/general/urls.html
         */
        public function __construct(){
                parent::__construct();
                $this->load->library('upload');
                $this->load->model('Models');
        }

        public function index()
        {
                $data['barang'] = $this->Models->get('barang')->result();
```

Dipindai dengan CamScanner

```
        $this->load->view('list-barang',$data);
}

public function list_barang()
{
        $data = $this->Models->get('barang')->result();

        $output = '';
        $no = 1;
        if(count($data) > 0){
                $output .= '<div class="row">
                    <div class="col-12 table-responsive">
                     <table id="order-listing" class="table">
                      <thead>
                       <tr>
                        <th>No</th>
                        <th>Tool Name</th>
                        <th>Stock</th>
                        <th>Safety Stock</th>
                        <th>Shop Name(Remark)</th>
                        <th>Action</th>
                       </tr>
                      </thead>
                      <tbody>';
          foreach($data as $value){
                $output .= '<tr>
                 <td>'.$no.'</td>
                 <td>'.$value->nama_barang.'</td>
                 <td>'.$value->stok.'</td>
                 <td>'.($value->safety_stock).'</td>
                 <td>'.$value->shop_name.'</td>
```

Dipindai dengan CamScanner

```
                    <td>
                     <a
href="'.site_url('barang/hapus_barang/').$value->id_barang.'" class="btn
btn-outline-danger">Delete</a>
                    </td>
                   </tr>';
            $no++;
              }
                    $output .= '</tbody></table></div></div>';
            }else{
                    $output .= 'No Data Found';
            }

            echo json_encode($output);
    }

    public function add_new_barang(){
            $this->load->view('add-barang');
    }

    public function hapus_barang($id_barang){
            $where = array('id_barang'=>$id_barang);
            $result = $this->Models->delete($where,'barang');
            if($result){
                    // $msg['success'] = true;
                    $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-success"><strong>Well done!</strong>
Record has been deleted.</div>');
                    // list_barang();
            }
            redirect('barang');
```

Dipindai dengan CamScanner

```
    }

    public function update_stok(){
        $id = $this->input->post('nama_barang');
        $jumlah = $this->input->post('jumlah');
        $cur_stok = 0;
        $total_stok = 0;
        // $msg['success'] = false;
        $where = array('id_barang'=>$id);
        $barang = $this->Models->fetch($where,'barang')->result();
            foreach ($barang as $value) {
                $cur_stok = $value->stok;
            }

        $total_stok = $jumlah + $cur_stok;
        $data = array('stok'=>$total_stok);
        $updatestok = $this->Models-
>update($where,$data,'barang');
        // if($result){
        $msg['success'] = true;
        $this->session->set_flashdata('msg','<div role="alert"
class="alert alert-success"><strong>Well done!</strong> New record has
been saved.</div>');
            // list_barang();
        // }
        echo json_encode($msg);
    }

    public function add_barang(){
        $nama = $this->input->post('nama');
        $stok = $this->input->post('stok');
```

Dipindai dengan CamScanner

```
                $shop_name = $this->input->post('shop_name');
                $msg['success'] = false;

                $data = array(
                                'nama_barang'=>$name,
                                'stok'=>$stok,
                                'shop_name'=>$shop_name,
                                'safety_stock' => '2',
                                'id_gudang'=>$this->session-
>userdata['logged_in']['id_akun']
                                );

                $result = $this->Models->insert($data,'barang');
                if($result){
                        $msg['success'] = true;
                        $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-success"><strong>Well done!</strong>
Stock has been updated.</div>');
                        // list_barang();
                }
                echo json_encode($msg);
                // redirect('barang');
        }
}
```

4. **Request**

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Request extends CI_Controller {
```

Dipindai dengan CamScanner

```
	/**
	 * Index Page for this controller.
	 *
	 * Maps to the following URL
	 *		http://example.com/index.php/welcome
	 *	- or -
	 *		http://example.com/index.php/welcome/index
	 *	- or -
	 * Since this controller is set as the default controller in
	 * config/routes.php, it's displayed at http://example.com/
	 *
	 * So any other public methods not prefixed with an underscore
will
	 * map to /index.php/welcome/<method_name>
	 * @see https://codeigniter.com/user_guide/general/urls.html
	 */

	public function __construct(){
		parent::__construct();
		$this->load->library('upload');
		$this->load->model('Models');
	}

	public function input_request()
	{
		if($this->session->userdata('logged_in')){
			// $this->load->view('index');
			$wherebarang = array('stok > '=>'0');
			$data['barang'] = $this->Models-
>fetch($wherebarang,'barang')->result();
			$this->load->view('form-request',$data);
```

Dipindai dengan CamScanner

```
                // print_r($data);
            }else{
                redirect('login');
            }
        }

        public function request_history(){
            $where = array('produksi.id_produksi'=>$this->session->userdata['logged_in']['id_akun']);
            $data['history'] = $this->Models->request_history($where)->result();
            $this->load->view('request-history',$data);
        }

        public function input_data_request(){
            date_default_timezone_set("Asia/Jakarta");
            $idbarang = $this->input->post('nama_barang');
            $jumlah = $this->input->post('jumlah');
            $date = date("Y-m-d");
            // $remark = $this->input->post('remark');
            $arr = [];
            for ($i = 0; $i < count($idbarang); $i++){
                $arr[] = $idbarang[$i];
            }


            $cek_dupes = $this->Models->array_has_dupes($arr);
            if($cek_dupes == 1){
                $this->session->set_flashdata('msg','<div role="alert" class="alert alert-danger"><strong>Error!</strong> Barang yang direquest tidak boleh ada kesamaan.</div>');
```

Dipindai dengan CamScanner

```
}else{
        for ($j = 0; $j < count($idbarang); $j++){

            $query = $this->db->query('SELECT stok
FROM barang WHERE id_barang = '.$idbarang[$j].");
            $row = $query->row();

            if($jumlah[$j] >= ($row->stok - 2))
            {
                $this->session-
>set_flashdata('msg','<div role="alert" class="alert alert-
danger"><strong>Error!</strong> Input over Stock.</div>');
                redirect('request/input_request');
                exit;
            }
            else
            {
                $data = array(

    'id_barang'=>$idbarang[$j],

    'jumlah_permintaan'=>$jumlah[$j],

    'tanggal_permintaan'=>$date,
                                                    //
 'remark'=>$remark[$j],

    'id_produksi'=>$this->session->userdata['logged_in']['id_akun']
                                                );
```

Dipindai dengan CamScanner

```
                                    $this->Models-
>insert($data,'permintaan');
                                    $this->session-
>set_flashdata('msg','<div role="alert" class="alert alert-
success"><strong>Well done!</strong> New Request has been
inserted.</div>');
                                    redirect('request/input_request');
                                    exit;
                                }
                            }


                        }
                    }
                }
```

5. **Approval**

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Approval extends CI_Controller {

    /**
     * Index Page for this controller.
     *
     * Maps to the following URL
     *          http://example.com/index.php/welcome
     *      - or -
     *          http://example.com/index.php/welcome/index
     *      - or -
     * Since this controller is set as the default controller in
```

Dipindai dengan CamScanner

```
	* config/routes.php, it's displayed at http://example.com/
	*
	* So any other public methods not prefixed with an underscore
will
	* map to /index.php/welcome/<method_name>
	* @see https://codeigniter.com/user_guide/general/urls.html
	*/

	public function __construct(){
		parent::__construct();
		$this->load->library('upload');
		$this->load->model('Models');
	}

	public function index()
	{
		if($this->session->userdata('logged_in')){
			// $this->load->view('index');
			$where = array('status_permintaan'=>'Pending');
			$data['request'] = $this->Models-
>request_list($where)->result();
			$this->load->view('request-list',$data);
		}else{
			redirect('login');
		}
	}

	public function update_to_approve($id_permintaan){
		$where = array('id_permintaan'=>$id_permintaan);
		$cur_stok = 0;
```

Dipindai dengan CamScanner

```
                $getpermintaan = $this->Models-
>fetch($where,'permintaan')->result();
                        foreach ($getpermintaan as $value) {
                                $idbarang = $value->id_barang;
                                $jumlah = $value->jumlah_permintaan;
                        }
                $wherebarang = array('id_barang'=>$idbarang);
                $barang = $this->Models->fetch($wherebarang,'barang')-
>result();
                        foreach ($barang as $value) {
                                $cur_stok = $value->stok;
                        }
                $stokakhir = $cur_stok - $jumlah;

                $data_permintaan =
array('status_permintaan'=>'Approve','id_gudang'=>$this->session-
>userdata['logged_in']['id_akun']);
                $data_barang = array('stok'=>$stokakhir);
                $updatepermintaan = $this->Models-
>update($where,$data_permintaan,'permintaan');
                $updatebarang = $this->Models-
>update($wherebarang,$data_barang,'barang');

                $this->session->set_flashdata('msg','<div role="alert"
class="alert alert-success"><strong>Well done!</strong> Request has
been approved.</div>');
                redirect('approval');
        }

        public function update_to_reject($id_permintaan){
```

Dipindai dengan CamScanner

```
                $data_permintaan =
array('status_permintaan'=>'Reject','id_gudang'=>$this->session-
>userdata['logged_in']['id_akun']);
                $where = array('id_permintaan'=>$id_permintaan);
                $updatepermintaan = $this->Models-
>update($where,$data_permintaan,'permintaan');

                $this->session->set_flashdata('msg','<div role="alert"
class="alert alert-success"><strong>Well done!</strong> Request has
been rejected.</div>');
                redirect('approval');
        }
}
```

6. **Report**

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Report extends CI_Controller {

        /**
         * Index Page for this controller.
         *
         * Maps to the following URL
         *              http://example.com/index.php/welcome
         *      - or -
         *              http://example.com/index.php/welcome/index
         *      - or -
         * Since this controller is set as the default controller in
         * config/routes.php, it's displayed at http://example.com/
         *
```

Dipindai dengan CamScanner

```
	* So any other public methods not prefixed with an underscore
will
	* map to /index.php/welcome/<method_name>
	* @see https://codeigniter.com/user_guide/general/urls.html
	*/

	public function __construct(){
		parent::__construct();
		$this->load->library('upload');
		$this->load->model('Models');
	}

	public function index(){
		if($this->session->userdata('logged_in')){
			$where = array('id_produksi'=>$this->session-
>userdata['logged_in']['id_akun']);
			$data['history'] = $this->Models->request_history()-
>result();
			$this->load->view('report',$data);
		}else{
			redirect('login');
		}
	}

	public function cetak_report(){
		$datefrom = $this->input->post('datefrom');
		$dateto = $this->input->post('dateto');
		// $where = array()
		// $datefrom = "2018-09-01";
		// $dateto = "2018-09-22";
		// $where = array('tanggal_permintaan >='=>$datefrom);
```

Dipindai dengan CamScanner

```
                // $data2 = $datefrom.$dateto;
                $output = '';
                $data = $this->Models->request_report($datefrom,$dateto)-
>result();
                $no = 1;
                if(count($data) > 0){
                        $output .= '<div class="row">
              <div class="col-12 table-responsive">
               <table id="order-listing" class="table">
                <thead>
                 <tr>
                  <th>No</th>
                  <th>Request Date</th>
                  <th>Tool Name</th>
                  <th>Qty</th>
                  <th>User Name</th>
                  <th>Status</th>
                 </tr>
                </thead>
                <tbody>';
          foreach ($data as $value) {
                $output .= '<tr>
                 <td>'.$no.'</td>
                 <td>'.$value->tanggal_permintaan.'</td>
                 <td>'.$value->nama_barang.'</td>
                 <td>'.$value->jumlah_permintaan.'</td>
                 <td>'.$value->nama_lengkap.'</td>
                 <td>
                   <label class="badge badge-danger">'.$value-
>status_permintaan.'</label>
                 </td>
```

Dipindai dengan CamScanner

```
            </tr>';
          $no++;
        }
        $output .='<tbody></table></div></div>';
            }
                // $this->load->view('report',$data);
            // $output = '<p>Tes</p>';
            // $output .= '<p>'.strtotime($datefrom).'</p>';
            echo json_encode($output);
        }
    }
```

7. Users

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Users extends CI_Controller {

    /**
     * Index Page for this controller.
     *
     * Maps to the following URL
     *          http://example.com/index.php/welcome
     *      - or -
     *          http://example.com/index.php/welcome/index
     *      - or -
     * Since this controller is set as the default controller in
     * config/routes.php, it's displayed at http://example.com/
     *
     * So any other public methods not prefixed with an underscore
will
```

Dipindai dengan CamScanner

```
	 * map to /index.php/welcome/<method_name>
	 * @see https://codeigniter.com/user_guide/general/urls.html
	 */

	public function __construct(){
		parent::__construct();
		$this->load->library('upload');
		$this->load->model('Models');
	}

	public function index()
	{
		if($this->session->userdata('logged_in')){
			$this->load->view('index');
		}else{
			redirect('login');
		}

	}

	public function update_admin($id_admin){
		$where = array('id_admin'=>$id_admin);
		$data['admin'] = $this->Models->fetch($where,'admin')-
>result();
		$this->load->view('update-admin',$data);
	}

	public function update_gudang($id_gudang){
		$where = array('id_gudang'=>$id_gudang);
		$data['gudang'] = $this->Models->fetch($where,'gudang')-
>result();
```

Dipindai dengan CamScanner

```
            $this->load->view('update-gudang',$data);
        }

        public function update_produksi($id_produksi){
            $where = array('id_produksi'=>$id_produksi);
            $data['produksi'] = $this->Models-
>fetch($where,'produksi')->result();
            $this->load->view('update-produksi',$data);
        }

        public function delete_admin($id_admin){
            $where = array('id_admin'=>$id_admin);
            $result = $this->Models->delete($where,'admin');
            if($result){
                $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-success"><strong>Well done!</strong>
Record has been deleted.</div>');
            }else{
                $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-danger"><strong>Oh Snap!</strong> This
record contain error (s), you should do double check.</div>');
            }

            redirect('users/admin_user');
        }

        public function delete_gudang($id_gudang){
            $where = array('id_gudang'=>$id_gudang);
            $result = $this->Models->delete($where,'gudang');
            if($result){
```

Dipindai dengan CamScanner

```
                $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-success"><strong>Well done!</strong>
Record has been deleted.</div>');
            }else{
                $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-danger"><strong>Oh Snap!</strong> This
record contain error (s), you should do double check.</div>');
            }

            redirect('users/gudang_user');
        }

        public function delete_produksi($id_produksi){
            $where = array('id_produksi'=>$id_produksi);
            $result = $this->Models->delete($where,'gudang');
            if($result){
                $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-success"><strong>Well done!</strong>
Record has been deleted.</div>');
            }else{
                $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-danger"><strong>Oh Snap!</strong> This
record contain error (s), you should do double check.</div>');
            }

            redirect('users/produksi_user');
        }

        public function update_data_gudang($id_gudang){

            $where = array('id_gudang'=>$id_gudang);
```

Dipindai dengan CamScanner

```
$username = $this->input->post('username');
$password = $this->input->post('password');
$nama_lengkap = $this->input->post('nama_lengkap');
$email = $this->input->post('email');
$notelp = $this->input->post('notelp');
$data = array(
                    'username'=>$username,
                    'password'=>$password,
                    'nama_lengkap'=>$nama_lengkap,
                    'email'=>$email
                );
$result = $this->Models->update($where,$data,'gudang');
if($result){
        $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-success"><strong>Well done!</strong>
New record has been saved.</div>');
}else{
        $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-danger"><strong>Oh Snap!</strong> This
record contain error (s), you should do double check.</div>');
}

redirect('users/gudang_user');
}

public function update_data_produksi($id_produksi){

$where = array('id_produksi'=>$id_produksi);

$username = $this->input->post('username');
```

Dipindai dengan CamScanner

```
            $password = $this->input->post('password');
            $nama_lengkap = $this->input->post('nama_lengkap');
            $email = $this->input->post('email');
            $notelp = $this->input->post('notelp');
            $data = array(
                        'username'=>$username,
                        'password'=>$password,
                        'nama_lengkap'=>$nama_lengkap,
                        'email'=>$email
                    );
            $result = $this->Models->update($where,$data,'produksi');
            if($result){
                $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-success"><strong>Well done!</strong>
New record has been saved.</div>');
            }else{
                $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-danger"><strong>Oh Snap!</strong> This
record contain error (s), you should do double check.</div>');
            }

            redirect('users/produksi_user');
        }

        public function update_data_user($id_admin){

            $where = array('id_admin'=>$id_admin);

            $username = $this->input->post('username');
            $password = $this->input->post('password');
            $nama_lengkap = $this->input->post('nama_lengkap');
```

Dipindai dengan CamScanner

```
            $email = $this->input->post('email');
            $notelp = $this->input->post('notelp');
            $data = array(
                                'username'=>$username,
                                'password'=>$password,
                                'nama_lengkap'=>$nama_lengkap,
                                'email'=>$email
                        );
            $result = $this->Models->update($where,$data,'admin');
            if($result){
                    $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-success"><strong>Well done!</strong>
New record has been saved.</div>');
            }else{
                    $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-danger"><strong>Oh Snap!</strong> This
record contain error (s), you should do double check.</div>');
            }

            redirect('users/admin_user');
        }

        public function admin_user(){
            if($this->session->userdata('logged_in')){
                    $data['admin'] = $this->Models->get('admin')-
>result();
                    $this->load->view('admin-user',$data);
            }else{
                    redirect('login');
            }
```

Dipindai dengan CamScanner

```
    }

    public function gudang_user(){
        if($this->session->userdata('logged_in')){
            $data['gudang'] = $this->Models->get('gudang')-
>result();
            $this->load->view('gudang-user',$data);
        }else{
            redirect('login');
        }

    }

    public function produksi_user(){
        if($this->session->userdata('logged_in')){
            $data['produksi'] = $this->Models->get('produksi')-
>result();
            $this->load->view('produksi-user',$data);
        }else{
            redirect('login');
        }

    }

    public function add_new_user(){
        if($this->session->userdata('logged_in')){
            $this->load->view('add-user');
        }else{
            redirect('login');
        }
```

Dipindai dengan CamScanner

```
}

public function add_user(){

        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $nama_lengkap = $this->input->post('nama_lengkap');
        $email = $this->input->post('email');
        $notelp = $this->input->post('notelp');
        $level = $this->input->post('level');

        $data = array(
                        'username'=>$username,
                        'password'=>$password,
                        'nama_lengkap'=>$nama_lengkap,
                        'email'=>$email,
                        'notelp'=>$notelp
                );

        if($level == 'admin'){
                $result = $this->Models->insert($data,'admin');
        }else if($level == 'gudang'){
                $result = $this->Models->insert($data,'gudang');
        }else if($level == 'produksi'){
                $result = $this->Models->insert($data,'produksi');
        }

        if($result){
                $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-success"><strong>Well done!</strong>
New record has been saved.</div>');
```

Dipindai dengan CamScanner

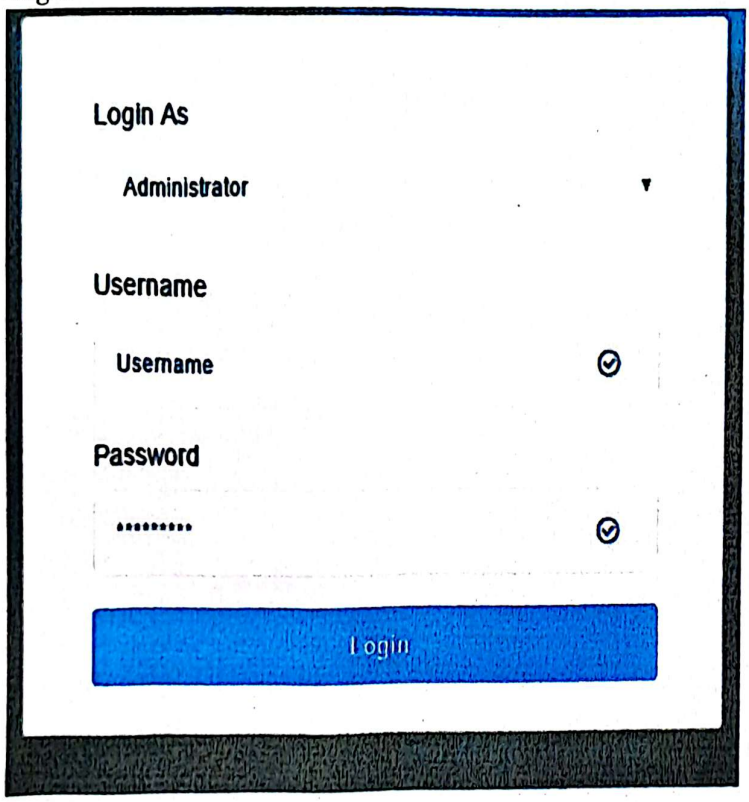
```
            }else{
                    $this->session->set_flashdata('msg','<div
role="alert" class="alert alert-danger"><strong>Oh Snap!</strong> This
record contain error (s), you should do double check.</div>');
            }

            redirect('users/add_new_user');
        }
    }
```

Dipindai dengan CamScanner

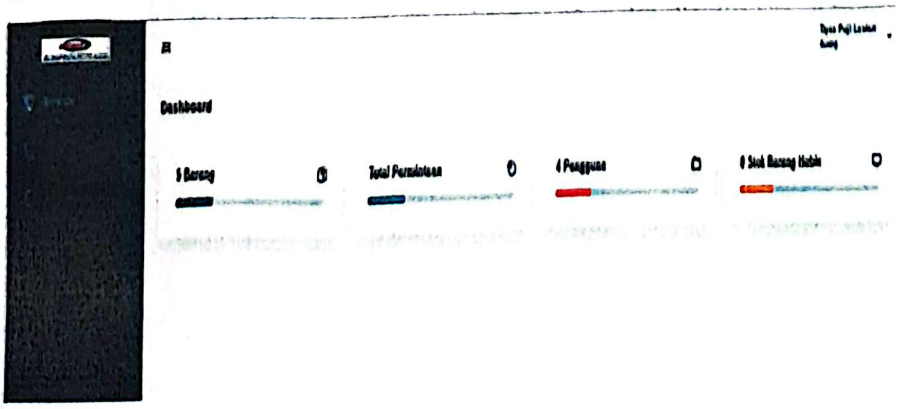
LAMPIRAN 2

TAMPILAN PROGRAM

1. *Login*



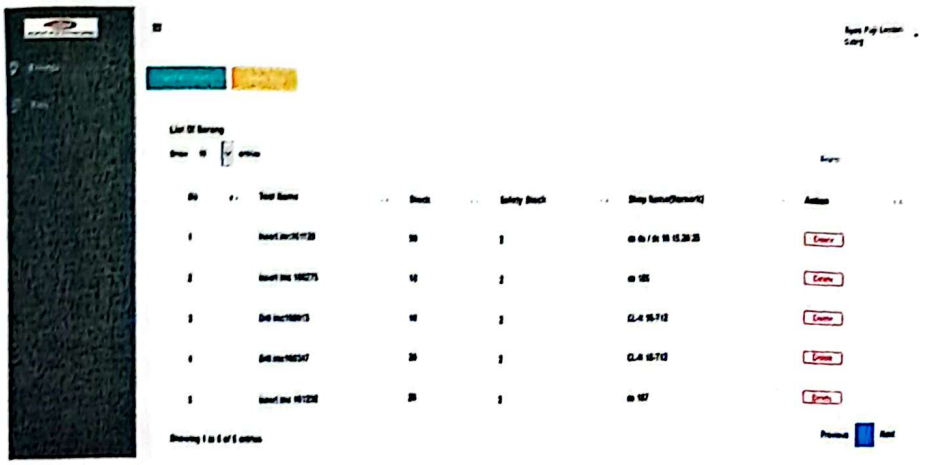
2. *Form* Beranda



Dipindai dengan CamScanner

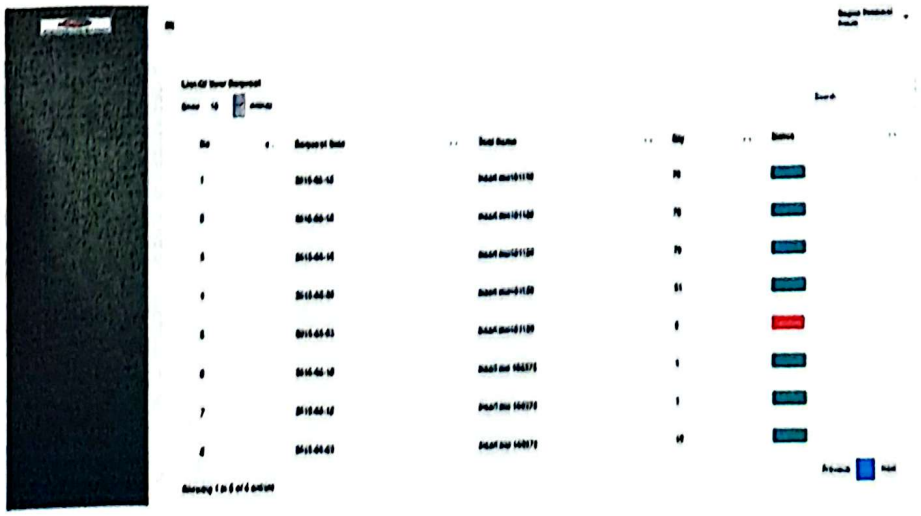
Dipindai dengan CamScanner

. *Form* data barang



7. *Form Request* barang

8. *Form Request History*



Dipindai dengan CamScanner