

**PERANCANGAN DAN IMPLEMENTASI METODE
ACTIVE SCHEDULE UNTUK MENGHASILKAN
URUTAN PROSES PRODUKSI PADA SISTEM
PENJADWALAN PRODUKSI DI PT GALIH AYOM
PARAMESTI**

TUGAS AKHIR

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

**OLEH
NURUL FIKRI
1315112**



**POLITEKNIK STMI
KEMENTERIAN PERINDUSTRIAN RI
JAKARTA
2019**

POLITEKNIK STMI
KEMENTERIAN PERINDUSTRIAN R.I

LEMBAR PENGESAHAN DOSEN PEMBIMBING

JUDUL TUGAS AKHIR:

**PERANCANGAN DAN IMPLEMENTASI METODE
ACTIVE SCHEDULE UNTUK MENGHASILKAN URUTAN PROSES
PRODUKSI PADA SISTEM PENJADWALAN PRODUKSI DI PT GALIH
AYOM PARAMESTI**

DISUSUN OLEH:

Nama : Nurul Fikri
Nim : 1315112
Program Studi : Sistem Informasi Industri Otomotif

Telah diperiksa dan disetujui untuk memenuhi sebagian syarat penyelesaian Program Sarjana Terapan Program Studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta

Jakarta, 20 Agustus 2019

Dosen Pembimbing



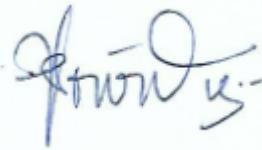
Ulil Hamida, ST,MT
NIP. 198103272005022001

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama	: Nurul Fikri
NIM	: <u>1315112</u>
Judul TA	: <u>Perancangan dan Implementasi Metode Active Schedule Untuk Menghasilkan Urutan Proses Produksi Pada Sistem Penjadwalan Produksi di PT Galih Ayom Paramesti</u>
Pembimbing	: <u>Ulil Hamida, ST, MT</u>

Tanggal	Bab	Keterangan	Paraf
20 Mei 2019	I	Bimbingan Bab I	✓
17 Juni 2019	I	Bimbingan Bab I & Metode	✓
20 Juni 2019	II	Revisi Bab II	✓
24 Juni 2019	I & II	Revisi Bab I & II	✓
25 Juni 2019		Konsultasi Program	✓
27 Juni 2019	III	Konsultasi Bab III	✓
4 Juli 2019	III & IV	Konsultasi Bab III & IV	✓
9 Juli 2019	III	Konsultasi Bab III	✓
22 Juli 2019		Konsultasi	✓
23 Juli 2019	IV & V	Bimbingan Bab IV & V (<i>Flowmap</i>)	✓
26 Juli 2019		Konsultasi Program	✓
6 Agustus 2019	V	Konsultasi & Revisi Bab V	✓
8 Agustus 2019	I – V	Pengecekan Keseluruhan	✓

Mengetahui,
 Ketua Program Studi Sistem Informasi
 Industri Otomotif



Noveriza Yuliasari, S.Si, M.T
 NIP. 197811212009012003

Dosen Pembimbing



Ulil Hamida, ST,MT
 NIP. 198103272005022001

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Nurul Fikri

NIM : 1315112

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

“Perancangan dan Implementasi Metode Active Schedule Untuk Menghasilkan Urutan Proses Produksi Pada Sistem Penjadwalan Produksi di PT Galih Ayom Paramesti”

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

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

Jakarta, 20 Agustus 2019



Nurul Fikri

ABSTRAK

PT Galih Ayom Paramesti merupakan salah satu perusahaan yang bergerak dalam bidang industri otomotif. Produk yang dihasilkan bervariasi seputar *engine parts* untuk kendaraan roda dua dan roda empat. Penjadwalan produksi yang terdapat pada perusahaan dilakukan berdasarkan dokumen PO (*purchase order*) & SD (*schedule delivery*) yang telah diterima dan disetujui oleh Bagian Pemasaran. Bagian PPC Manajer kemudian akan menginstruksikan Bagian Gudang untuk melakukan pengecekan ketersediaan bahan baku. Proses penjadwalan produksi akan ditunda apabila bahan baku yang dibutuhkan belum tersedia. Bahan baku tersebut selanjutnya akan dipesan oleh Bagian Pengadaan atas instruksi Bagian PPC Manajer. Proses penjadwalan produksi dibuat oleh Bagian PPC Manajer berdasarkan intuisi karyawan tersebut dan mengakibatkan jadwal yang dihasilkan bervariasi antara produk satu dengan yang lain. Untuk mengatasi masalah tersebut diperlukan suatu metode yang dapat diaplikasikan ke dalam penjadwalan produksi. Metode yang digunakan yaitu Metode *Active Schedule*. Metode *Active Schedule* merupakan salah satu metode penjadwalan produksi yang memprioritaskan proses pengerjaan dengan waktu terendah serta waktu tunggu minimal. Metodologi pengembangan sistem informasi yang digunakan pada penelitian ini adalah *evolutionary prototype*. Analisis dan perancangan sistem menggunakan UML (*Unified Modeling Language*) yang terdiri dari *use case diagram*, *activity diagram* dan *deployment diagram*. Pemodelan data dibuat menggunakan ERD (*Entity Relationship Diagram*) dan Kamus Data. Sistem informasi ini diimplementasikan menggunakan CodeIgniter 3.1.10 sebagai *framework* dan MariaDB 10.1.37 sebagai perangkat lunak manajemen basis data. Penggunaan Metode *Acitive Schedule* telah diterapkan pada sistem informasi yang dibuat. Implementasi tersebut dapat membantu proses penjadwalan produksi untuk menghasilkan urutan proses produksi di PT Galih Ayom Paramesti secara tepat dan cepat.

Kata kunci: Sistem Informasi, Penjadwalan Produksi, *PPC*, *PPIC*, Metode *Active Schedule*, Jadwal Produksi.

KATA PENGANTAR

Bismillahirahmanirahim

Puji syukur kepada Allah Subhanahu wa Ta'ala, atas rahmat dan karunia-Nya sehingga dapat diselesaikan Tugas Akhir dengan judul "**Perancangan dan Implementasi Metode Active Schedule Untuk Menghasilkan Urutan Proses Produksi Pada Sistem Penjadwalan Produksi di PT Galih Ayom Paramesti.**". Penulisan Tugas Akhir ini dibuat sebagai salah satu syarat penyelesaian Program Sarjana Terapan pada Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta.

Dalam pembuatan Tugas Akhir ini, banyak pihak yang senantiasa mendukung, membimbing dan membantu baik secara langsung maupun tidak. Oleh karena itu, saya mengucapkan terima kasih kepada:

- Bapak Dr. Mustofa, S.T., M.T. selaku Direktur Politeknik STMI Jakarta.
- Ibu Noveriza Yuliasari, S.Si., M.T. selaku Ketua Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta
- Ibu Ulil Hamida, S.T., M.T. selaku dosen pembimbing yang telah meluangkan waktu untuk membantu dan memberikan arahan serta masukan dalam penulisan dan penyusunan tugas akhir.
- Seluruh dosen yang ada di Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.
- Kedua orang tua saya yang selalu memberikan dukungan dan ridhonya agar saya dapat menjalankan pendidikan di Politeknik STMI Jakarta.
- Semua adik saya yang selalu memberikan dukungan untuk keberhasilan saya dalam menyelesaikan tugas akhir ini.
- Bapak Julius Handaru Pradipta selaku *Deputy Director* PT Galih Ayom Paramesti.
- Bapak Adi Alfianto selaku pembimbing yang telah membantu mengarahkan dan membimbing selama Kerja Lapangan di PT Galih Ayom Paramesti.

- Bapak Andreas Pramudyo selaku PPIC *Head/Manager* PT Galih Ayom Paramesti.
- Bapak Setyo Gunawan, Bapak Ardi, Bapak Dika, Bu Yanti, Bu Yatmi dan seluruh pegawai di PT Galih Ayom Paramesti yang telah membantu memberikan arahan serta informasi dalam pengumpulan data.
- Intan Rinjani dan Rizki Ahmad sebagai tempat konsultasi yang telah membantu memberikan saran atas pembuatan tugas akhir dan program.
- PKL Brother Fahri Bagus Firmansyah, Marcellino Jonathan, Rehan Egi Junior yang selalu ada disetiap waktu untuk membantu dan memberikan saran dalam segala hal terkait penyusunan tugas akhir.
- Teman-teman SII03 yang tidak dapat disebutkan namanya satu per satu atas bantuan, kebersamaan serta solidaritasnya selama 8 semester menimba ilmu bersama.
- Teman-teman mahasiswa/i Politeknik STMI Jakarta Program Studi Sistem Informasi 2015 atas kebersamaan dan motivasinya selama menjalankan pendidikan di Politeknik STMI Jakarta.
- Pihak lain yang saya tidak dapat sebutkan satu per satu.

Demikian yang dapat saya sampaikan, semoga Allah SWT membalas kebaikan semua pihak yang telah membantu dalam menyelesaikan Tugas Akhir ini. Telah disadari bahwa penulisan Laporan Tugas Akhir ini masih jauh dari sempurna, untuk itu dimohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan di masa yang akan datang. Terima kasih.

Jakarta, 20 Agustus 2019



Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL

LEMBAR PENGESAHAN DOSEN PEMBIMBING

LEMBAR BIMBINGAN iii

LEMBAR PERNYATANAAN KEASLIAN iv

ABSTRAK v

KATA PENGANTAR vi

DAFTAR ISI viii

DAFTAR GAMBAR xii

DAFTAR TABEL xiv

BAB I PENDAHULUAN 1

- 1.1. Latar Belakang 1
- 1.2. Pokok Permasalahan 3
- 1.3. Tujuan Penelitian 3
- 1.4. Batasan Masalah 3
- 1.5. Manfaat 4
- 1.6. Sistematika Penulisan 4

BAB II LANDASAN TEORI 6

- 2.1. Kajian Penelitian 6
- 2.2. Pengertian Implementasi 8
- 2.3. Pengertian Rancang Bangun 8
- 2.4. Konsep Dasar Sistem 8
 - 2.4.1. Pengertian Sistem 9
 - 2.4.2. Karakteristik Sistem 10
 - 2.4.3. Klasifikasi Sistem 11
 - 2.4.4. Daur Hidup Sistem 12
- 2.5. Informasi 14
 - 2.5.1. Pengertian Informasi 14

2.5.2.	Karakteristik Informasi	15
2.6.	Konsep Dasar Sistem Informasi	16
2.6.1.	Komponen Sistem Informasi	17
2.7.	Produksi	18
2.8.	Penjadwalan	19
2.9.	<i>Active Schedule</i>	21
2.9.1.	Kelebihan dan Kekurangan <i>Active Schedule</i>	22
2.10.	<i>System Development Life Cycle (SDLC)</i>	22
2.10.1.	Model <i>Prototype</i>	24
2.10.2.	Kelemahan dan Penyelesaian Model <i>Prototype</i>	24
2.10.3.	<i>Evolutionary Prototype</i>	25
2.11.	<i>Flowchart</i>	26
2.11.1.	Jenis-Jenis <i>Flowchart</i>	26
2.11.2.	Simbol-Simbol <i>Flowchart</i>	27
2.12.	<i>Unified Modeling Language (UML)</i>	31
2.13.	<i>Entity Relationship Diagram (ERD)</i>	37
2.14.	Kamus Data	39
2.15.	<i>Windows Navigation Diagram (WND)</i>	39
2.16.	<i>Hypertext Preprocessor (PHP)</i>	40
2.17.	<i>Hypertext Markup Language (HTML)</i>	40
2.18.	MariaDB	41
2.19.	<i>Framework</i>	41
2.20.	CodeIgniter	41
2.21.	MVC (<i>Model-View-Controller</i>)	44
2.22.	<i>Testing</i>	45
2.23.	<i>Behavioral (Black-box) Testing</i>	45
BAB III METODOLOGI PENELITIAN	46
3.1.	Metodologi Penelitian	46
3.2.	Jenis dan Sumber Data	46
3.3.	Metode Pengumpulan Data	46
3.4.	Kerangka Penelitian	47

3.4.1.	Data Input	48
3.4.2.	Metode <i>Active Schedule</i>	49
3.4.3.	Penjadwalan Metode <i>Active Schedule</i>	50
3.4.4.	Identifikasi Kebutuhan	53
3.4.5.	Pembuatan <i>Prototype</i>	53
3.4.6.	Evaluasi <i>Prototype</i>	54
3.4.7.	Gunakan Sistem	54
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA	55
4.1.	Sejarah Perusahaan	55
4.2.	Profil Perusahaan	55
4.3.	Makna Logo Perusahaan	55
4.4.	Struktur Organisasi Perusahaan	56
4.5.	Tugas dan Wewenang	58
4.6.	Struktur Organisasi PPC (<i>Production Planning and Control</i>) ..	60
4.7.	Tugas dan Wewenang PPC	61
4.8.	Dokumen Terkait Penjadwalan Produksi	62
4.9.	Sistem Informasi Penjadwalan Produksi yang Sedang Berjalan...	
	67
BAB V HASIL ANALISIS DAN PEMBAHASAN	70
5.1.	Analisis Kebutuhan Sistem	70
5.2.	Sistem Informasi Penjadwalan Produksi Usulan	72
5.3.	Analisis Sistem Usulan	73
5.3.1.	<i>Use Case Diagram</i>	74
5.3.2.	<i>Activity Diagram</i>	81
5.3.3.	<i>Deployment Diagram</i>	84
5.4.	Pemodelan Data Sistem Usulan	85
5.4.1.	<i>Entity Relationship Diagram (ERD)</i>	85
5.4.2.	Kamus Data	86
5.5.	<i>Windows Navigation Diagram (WND)</i>	89
5.6.	Perancangan Antarmuka Sistem	91
5.7.	Implementasi Metode <i>Active Schedule</i> Pada Sistem	99

5.8.	Implementasi Sistem	104
5.9.	Pengujian <i>Black Box Testing</i>	104
BAB VI	KESIMPULAN DAN SARAN	116
6.1.	Kesimpulan	116
6.2.	Saran	116
	DAFTAR PUSTAKA	117
	LAMPIRAN	

DAFTAR GAMBAR

Gambar II.1 Karakteristik Sistem	11
Gambar II.2 Daur Hidup Sistem	14
Gambar II.3 SDLC Model <i>Prototype</i>	23
Gambar II.4 Kerangka <i>Evolutionary Prototype</i>	25
Gambar II.5 Sistem <i>Client/Server Deployment Diagram</i>	36
Gambar II.6 Contoh <i>Windows Navigation Diagram</i>	37
Gambar II.7 Logo CodeIgniter	41
Gambar II.8 <i>Application Flowchart</i>	43
Gambar II.9 <i>Application Flowchart</i>	44
Gambar III.1 Kerangka Penelitian	48
Gambar IV.1 Logo PT Galih Ayom Paramesti	56
Gambar IV.2 Struktur Organisasi PT Galih Ayom Paramesti	57
Gambar IV.3 SO PPC PT Galih Ayom Paramesti	60
Gambar IV.4 Contoh PO PT Galih Ayom Paramesti	62
Gambar IV.5 Contoh <i>Schedule Delivery</i>	65
Gambar IV.6 Contoh Format Jadwal Produksi	66
Gambar IV.7 Bagan Alir Penjadwalan Produksi Keseluruhan	68
Gambar IV.8 Bagan Alir Pembuatan Urutan dan Jadwal Produksi	69
Gambar V.1 <i>Flowmap</i> Sistem Informasi Penjadwalan Produksi Usulan	73
Gambar V.2 <i>Use Case Diagram</i> Sistem Usulan	74
Gambar V.3 <i>Activity Diagram Log In</i>	81
Gambar V.4 <i>Activity Diagram</i> Mengelola Data Pengguna	82
Gambar V.5 <i>Activity Diagram</i> Mengelola Data Mesin	82
Gambar V.6 <i>Activity Diagram</i> Mengelola Data Produk	83
Gambar V.7 <i>Activity Diagram</i> Mengelola Jadwal Produksi	83
Gambar V.8 <i>Activity Diagram</i> Melihat dan Mencetak Hasil Urutan Proses Produksi	84

Gambar V.9 <i>Deployment Diagram</i> Sistem Usulan	84
Gambar V.10 <i>ERD</i> Sistem Usulan	86
Gambar V.11 <i>Windows Navigation Diagram</i> Sistem Usulan	90
Gambar V.12 <i>UI Log In</i>	91
Gambar V.13 <i>UI Dashboard Admin</i>	92
Gambar V.14 <i>UI Dashboard PPC Manager</i>	92
Gambar V.15 <i>UI Dashboard Engineer</i>	93
Gambar V.16 <i>UI Form</i> Data Pengguna	93
Gambar V.17 <i>UI Form</i> Tambah dan Ubah Data Pengguna	94
Gambar V.18 <i>UI Form</i> Data Mesin	94
Gambar V.19 <i>UI Form</i> Tambah dan Ubah Data Mesin	95
Gambar V.20 <i>UI Form</i> Data Produk	95
Gambar V.21 <i>UI Form</i> Tambah dan Ubah Data Produk	96
Gambar V.22 <i>UI Form</i> Data Detail Produk	96
Gambar V.23 <i>UI Form</i> Tambah Detail Produk	97
Gambar V.24 <i>UI Form</i> Data Jadwal	97
Gambar V.25 <i>UI Form</i> Tambah Jadwal	98
Gambar V.26 <i>UI Form</i> Data Hasil	98

DAFTAR TABEL

Tabel II.1 Simbol-Simbol <i>Flowchart</i>	28
Tabel II.2 Simbol-Simbol <i>Use Case Diagram</i>	32
Tabel II.3 Simbol-Simbol <i>Activity Diagram</i>	34
Tabel II.4 Simbol-Simbol <i>Deployment Diagram</i>	36
Tabel II.5 Simbol-Simbol ERD	38
Tabel III.1 Data Produk	48
Tabel III.2 Data Mesin	49
Tabel III.3 Data Urutan Mesin	49
Tabel III.4 Data Waktu Proses Produksi (detik)	49
Tabel III.5 Waktu Penyelesaian	50
Tabel III.6 Inisialisasi Pertama	50
Tabel III.7 Perhitungan Total Waktu Minimal	51
Tabel III.8 Pergantian Urutan Proses Terpilih	51
Tabel III.9 Penyelesaian Metode <i>Active Schedule Actual</i>	52
Tabel V.1 Analisis Kebutuhan Sistem Usulan	70
Tabel V.2 Analisis Kebutuhan Sistem <i>Functional Requirement</i>	71
Tabel V.3 Analisis Kebutuhan Sistem <i>Non Functional Requirement</i>	71
Tabel V.4 Definisi Aktor <i>Use Case Diagram</i> Usulan	75
Tabel V.5 Definisi <i>Use Case</i> Usulan	75
Tabel V.6 Skenario <i>Use Case</i> Melakukan <i>Log in</i>	76
Tabel V.7 Skenario <i>Use Case</i> Mengelola Data Pengguna	77
Tabel V.8 Skenario <i>Use Case</i> Mengelola Data Mesin	78
Tabel V.9 Skenario <i>Use Case</i> Mengelola Data Produk	79
Tabel V.10 Skenario <i>Use Case</i> Mengelola Jadwal Produksi	80
Tabel V.11 Skenario <i>Use Case</i> Melihat Hasil Urutan Proses Produksi	80
Tabel V.12 Tabel Pengguna	87
Tabel V.13 Tabel Detail Produk	87
Tabel V.14 Tabel Produk	87

Tabel V.15 Tabel Mesin	88
Tabel V.16 Tabel Penjadwalan	88
Tabel V.17 Tabel Hasil	89
Tabel V.18 Tabel Detail CProduk	89
Tabel V.19 <i>Test Case</i> Melakukan <i>Log in</i>	104
Tabel V.20 <i>Test Case</i> Mengelola Data Pengguna	106
Tabel V.21 <i>Test Case</i> Mengelola Data Mesin	108
Tabel V.22 <i>Test Case</i> Mengelola Data Produk	111
Tabel V.23 <i>Test Case</i> Mengelola Data Detail Produk	113

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PENGESAHAN

JUDUL TUGAS AKHIR:

**PERANCANGAN DAN IMPLEMENTASI METODE ACTIVE SCHEDULE
UNTUK MENGHASILKAN URUTAN PROSES PRODUKSI PADA
SISTEM PENJADWALAN PRODUKSI DI PT GALIH AYOM
PARAMESTI**

Disusun Oleh:

Nama : Nurul Fikri
NIM : 1315112
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 11 September 2019.

Jakarta, 11 September 2019

Dosen Pembimbing

Ketua Penguji

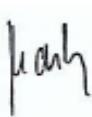


Ulil Hamida, ST, MT
NIP: 198103272005022001

Ahmad Juniar, S.Kom, MT
NIP : 197906052006041002

Dosen Penguji

Dosen Penguji



Dr. Ridzky Kramanandita, S.Kom, MT
NIP : 197403022002121001

Fifi Lailasari H, S.Kom, M.Kes
NIP : 197310162005022001

BAB I

PENDAHULUAN

1.1. Latar Belakang

PT Galih Ayom Paramesti merupakan perusahaan manufaktur yang bergerak di bidang industri otomotif. Terkenal akan riwayatnya dalam pembuatan produk dengan berbagai macam jenis, PT Galih Ayom Paramesti memiliki fokus produksi yaitu komponen *engine parts* untuk kendaraan roda dua dan roda empat. Produk tersebut merupakan produk yang dipesan khusus secara langsung oleh perusahaan utama penjual komponen otomotif yang tersebar di seluruh Indonesia.

Dalam menjalankan proses bisnisnya PT Galih Ayom Paramesti dibantu oleh Bagian *PPC (Production Planning and Control)* atau dalam perusahaan industri otomotif lain biasa disebut dengan Bagian *PPIC (Production Planning and Inventory Control)*. Bagian *PPC* memiliki tugas utama yaitu membuat jadwal produksi perusahaan. Proses pembuatan jadwal produksi dimulai ketika pesanan baru diterima oleh Bagian Pemasaran. Proses tersebut kemudian dilanjutkan dengan pengecekan kebutuhan-kebutuhan terkait proses produksi seperti bahan baku untuk memproduksi produk pesanan hingga bahan yang digunakan untuk membantu kinerja mesin seperti pelumas. Pada proses pengecekan tersebut akan dilihat, apabila bahan kebutuhan belum tersedia maka akan dilakukan pemesanan. Namun, apabila bahan kebutuhan sudah tersedia dan mencukupi maka akan dilakukan proses perencanaan produksi.

Proses perencanaan produksi merupakan suatu proses penting yang dilakukan sebelum membuat jadwal produksi. Pada proses tersebut dilakukan pertimbangan-pertimbangan seperti jumlah bahan baku yang dibutuhkan, waktu pengrajan suatu produk dari awal hingga akhir pembuatan serta kapasitas pengrajan mesin ketika digunakan untuk memproduksi produk tersebut. Pertimbangan tersebut dilakukan untuk mengetahui jumlah produk yang nantinya dapat dihasilkan pada satu sesi hingga pada satu hari kerja.

Hasil akhir dari tugas Bagian *PPC* yaitu berupa dokumen jadwal produksi. Dokumen tersebut nantinya akan dijadikan acuan oleh Bagian Produksi ketika memproduksi barang sehingga diperoleh jumlah yang sesuai dengan permintaan pelanggan. Secara garis besar dokumen jadwal produksi terdiri dari urutan proses penggerjaan produk, tanggal penggerjaan produk dan jumlah produk. Jumlah produk tersebut kemudian dibagi menjadi jumlah produk yang direncanakan sebelumnya dan jumlah produk yang dihasilkan pada proses produksi yang sedang dilakukan.

Pada setiap periodenya yang memiliki kisaran waktu satu bulan, PT Galih Ayom Paramesti selalu menghasilkan banyak komponen otomotif. Hal tersebut membuat rumitnya sistem penjadwalan produksi yang ada di PT Galih Ayom Paramesti. Tingkat kerumitan tersebut semakin bertambah karena dalam membuat penjadwalan produksi, Bagian *PPC* tidak menggunakan metode penjadwalan yang dapat mendukung dalam pengambilan keputusan yang tepat dan konsisten. Hingga saat ini Bagian *PPC* membuat jadwal produksi secara manual menggunakan intuisi karyawan.

Hal itu menyebabkan penentuan urutan proses produksi pada penjadwalan produksi menjadi tidak tepat sehingga terdapat mesin yang menunggu untuk digunakan. Berbagai kerumitan tersebut kembali dirasakan ketika Bagian *PPC* menanggapi terlalu banyaknya pesanan sehingga jenis komponen yang harus dihasilkan semakin banyak. Tak jarang Bagian Produksi melakukan kerja lembur agar produk yang dihasilkan sesuai dengan jumlah pesanan. Namun, terkadang PT Galih Ayom Paramesti tetap tidak sanggup untuk memenuhi pesanan sehingga terpaksa melakukan *outsource*. *Outsource* dilakukan dengan menyewa jasa perusahaan lain untuk mengerjakan sebagian proses produksi dari suatu produk.

Untuk mengatasi permasalahan tersebut, diperlukan suatu metode yang dapat membantu dalam membuat urutan proses produksi. Metode tersebut diharapkan dapat meningkatkan konsistensi dan efektifitas dengan mengurangi waktu yang dibutuhkan karyawan ketika hendak membuat jadwal produksi. Judul Tugas Akhir ini adalah “Perancangan dan Implementasi Metode *Active Schedule* Untuk Menghasilkan Urutan Proses Produksi Pada Sistem Penjadwalan Produksi di PT Galih Ayom Paramesti”.

1.2. Pokok Permasalahan

Berikut merupakan beberapa permasalahan yang terdapat pada bagian PPC di PT Galih Ayom Paramesti:

1. Proses pembuatan jadwal produksi sepenuhnya dilakukan berdasarkan intuisi karyawan. Hal tersebut dapat mempengaruhi konsistensi karyawan dalam menentukan urutan proses produksi yang berakibat adanya ketimpangan waktu produksi yang disebabkan oleh pengaturan mesin yang tidak tepat.
2. Pembuatan jadwal produksi yang dilakukan tanpa menggunakan metode apapun dapat menghambat kinerja Bagian *PPC* karena harus membuatnya secara manual dengan pendekatan yang berbeda-beda, sehingga membutuhkan waktu yang lama.

1.3. Tujuan Penelitian

Tujuan dilakukannya penelitian Tugas Akhir ini adalah sebagai berikut:

1. Melakukan perancangan sistem informasi penjadwalan produksi yang terkomputerisasi serta berbasis *web* agar pembuatan jadwal produksi tidak dilakukan secara manual dan dapat menghemat waktu dalam penentuan urutan proses produksi.
2. Melakukan implementasi Metode *Active Schedule* untuk menghitung urutan proses produksi pada sistem informasi penjadwalan produksi agar pembuatan jadwal produksi menggunakan pendekatan yang sama di setiap produknya sehingga tidak membutuhkan yang waktu lama.

1.4. Batasan Masalah

Batasan dalam penyusunan laporan Tugas Akhir adalah sebagai berikut:

1. Proses penelitian dilakukan pada PT Galih Ayom Paramesti yang memiliki alamat di Jalan Inspeksi Tarum Barat, Pekopen, Lambang Jaya, Tambun, Bekasi.
2. Jangka waktu dilakukannya penelitian yaitu selama tiga bulan dimulai pada bulan Oktober 2018 hingga Desember 2018.

3. Bagian dilakukannya penelitian yaitu Bagian *PPC* atau *Production Planning and Control* pada sub-bagian *PPC Head* atau *PPC Manajer*.
4. Produk yang dijadikan sebagai acuan ketika dilakukannya perhitungan menggunakan metode *Active Schedule* yaitu Sleeve 0068, Collar 2561, dan Shaft Kick 0014.
5. Proses yang dijadikan sebagai bahan acuan ketika implementasi metode *Active Schedule* hanyalah proses yang dikerjakan di dalam perusahaan (*in-house*), bukan yang dikerjakan di luar perusahaan (*out-source*).

1.5. Manfaat

Manfaat yang dapat diambil dari dibuatnya laporan Tugas Akhir ini antara lain adalah:

1. Bagi perusahaan
 - a. Penelitian ini diharapkan dapat menjadi acuan perbaikan pada perusahaan, khususnya pada PT Galih Ayom Paramesti di bagian *PPC Manager* dalam meningkatkan efektifitas ketika membuat jadwal produksi.
2. Bagi mahasiswa
 - a. Sebagai pembelajaran dalam menerapkan metode *Active Schedule* pada perusahaan industri otomotif.
 - b. Sebagai pembelajaran dalam perancangan dan implementasi aplikasi pada perusahaan industri otomotif.
3. Bagi pihak lain
 - a. Hasil penelitian ini diharapkan dapat menjadi sumbangan pemikiran dan sebagai referensi akan penelitian yang serupa di masa yang akan datang.

1.6. Sistematika Penulisan

Laporan penelitian Tugas Akhir ini tersusun atas beberapa tahapan dan langkah-langkah yang diperoleh berdasarkan hasil pengamatan yang telah dilakukan selama Praktik Kerja Lapangan. Berikut ini merupakan tahapan dalam penyusunan laporan ini:

BAB I PENDAHULUAN

Pada bab ini berisi penjelasan mengenai gambaran Tugas Akhir yang dilaksanakan dan mencangkup deskripsi latar belakang, permasalahan yang diangkat, tujuan dari penelitian, batasan masalah, manfaat serta sistematika penulisan yang sesuai dengan isi laporan penelitian.

BAB II LANDASAN TEORI

Pada bab ini menjelaskan mengenai teori-teori yang dirangkum dari jurnal, buku-buku dan berbagai referensi literatur lainnya serta kajian jurnal yang berkaitan dengan tema penelitian yang diambil. Teori-teori kajian yang dipaparkan pada laporan ini antara lain adalah metode *Active Schedule*, penjadwalan produksi, artikel jurnal yang sesuai dengan topik yang diambil, metode pengembangan sistem, *Unified Modeling Language (UML)*, *Flowchart*, Hypertext Preprocessor (PHP), Code Igniter dan MariaDB.

BAB III METODOLOGI PENELITIAN

Pada bab ini diuraikan metode yang digunakan dalam pengumpulan data, metode dalam melakukan pengembangan sistem, metode pemecahan masalah serta langkah-langkah dalam penggunaan metode yang telah dipilih.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Pada bab ini menjelaskan mengenai data-data yang telah diperoleh di PT Galih Ayom Paramesti pada saat dilakukannya pengamatan.

BAB V ANALISIS DAN PEMBAHASAN

Pada bab ini berisi analisis rinci dari pengolahan data yang dilakukan. Pembahasan meliputi interpretasi hasil analisis yang diperoleh dengan menggunakan metode yang telah ditentukan, adapun dalam Tugas Akhir ini menggunakan metode *Active Schedule*.

BAB VI PENUTUP

Pada bab ini diuraikan beberapa kesimpulan dari penelitian serta saran-saran bagi pihak perusahaan dengan ruang lingkup sistem penjadwalan produksi yang telah diteliti.

BAB II

LANDASAN TEORI

2.1. Kajian Penelitian

Pada penelitian perancangan dan implementasi metode *Active Schedule* dilakukan dengan mempelajari metode dari penelitian sejenis. Terdapat beberapa referensi yang dikutip berdasarkan jurnal penelitian yang kemudian dirangkum untuk selanjutnya dijadikan acuan dalam penggerjaan tugas akhir ini yang antara lain.

1. *Penjadwalan Waktu Beban Kerja dengan Metode Algoritma Active Schedule dan Heuristic Schedule untuk Efisiensi Daya Listrik (I Gede Suputra Widharma, I Made Sajayasa, 2014)*

Pada jurnal yang dilakukan oleh (Widharma & Sajayasa, 2014) mengenai perhitungan untuk meningkatkan efisiensi daya listrik menggunakan metode *active schedule* dan *heuristic schedule*. Objek penelitian dilakukan pada *power house* tenaga listrik di Politeknik Negeri Bali, yaitu pada *transformator* distribusi 630 KVA yang digunakan untuk memasok daya bagi institusi yang terbagi dalam 21 grup waktu kerja. Penelitian dilakukan dengan memperhatikan lokasi serta waktu kerja yang jadwalnya dapat diatur ulang, yaitu di Laboratorium Refrigerasi pada hari selasa, hari rabu dan hari jumat.

Pada penelitian ini digunakan aturan prioritas MWKR (*Most Work Remaining*) dimana prioritas tertinggi diberikan kepada operasi suatu job yang mempunyai sisa waktu proses terlama. Sebelum dilakukannya penelitian diketahui bahwa daya yang dilewati sebesar 345 KVA. Setelah dilakukan penelitian diperoleh hasil tingkat pembebanan puncak sebesar 375 KVA. Hal itu diperoleh dengan mengatur jadwal praktik hari Selasa pukul 11.00 ke hari Senin pukul 08.30, hari Rabu pukul 11.00 ke pukul 13.00 dan hari Jumat pukul 10.30 ke hari Kamis pukul 08.00. Besarnya daya yang dapat diatur pada pembebanan transformator distribusi 630 KVA sebesar 36 KVA, sehingga diperoleh efisiensi daya listrik.

2. *Job Scheduling Menggunakan Metode Algoritma Active, Algoritma Non Delay dan Heuristic Schedule Generation* (Suseno, Bian Indrakusuma, 2014)

Pada jurnal penelitian ini, peneliti (Suseno & Indrakusuma, 2014) membuat perhitungan untuk menentukan metode yang paling optimal ketika digunakan pada objek penelitian yaitu Borobudur Knitting yang terdapat di Yogyakarta. Metode yang digunakan sebagai acuan yaitu Algoritma *Active Schedule Generation*, Algoritma *Non-Delay Generation* dan Algoritma *Heuristic Generation*. Data penelitian yang digunakan yaitu data yang diperoleh pada bulan Maret 2014. Data yang digunakan untuk penelitian antara lain Data Produk dan Jumlah Produksi, Data Mesin, Data *Routing* Mesin dan Data Waktu Proses.

Kesimpulan yang diperoleh dari analisis penyelesaian yaitu: Penjadwalan kondisi aktual menghasilkan waktu penyelesaian keseluruhan produk sebesar 7059,12 menit. Penjadwalan menggunakan Algoritma *Active Schedule Generation* menghasilkan waktu sebesar 6659,77 menit. Penjadwalan menggunakan Algoritma *Non-Delay Schedule Generation* menghasilkan waktu sebesar 6659,77 menit serta penjadwalan menggunakan Algoritma *Heuristic Schedule Generation* menghasilkan waktu sebesar 7249,14 menit. Berdasarkan perolehan waktu yang didapat maka metode yang paling optimal yaitu Algoritma *Active Schedule Generation* dan Algoritma *Non-Delay Schedule Generation* dengan perolehan waktu 6669,77 menit.

3. *Optimasi Penjadwalan Produksi pada Game Hay Day dengan Kriteria Minimalisasi Waktu, Minimalisasi Biaya dan Experience Menggunakan Metode Algoritma Active Schedule Generation* (Muhammad Ainul Yaqin, Qothrun Nada Munawaroh, Fani Dwi Cahyanti, M Bagus Syaifullah, 2014)

Pada jurnal penelitian yang dilakukan oleh (Yaqin *et al*, 2014) mengenai implementasi metode *Active Schedule Generation* pada *Game Hay Day*. Tujuan dari penelitian ini yaitu optimasi penjadwalan produksi dengan kriteria Minimalisasi Waktu, Minimalisasi Biaya dan Experience dengan data yang

diperoleh dari permainan tersebut. Sampel data yang digunakan diambil dari Hay Day pada *level* 1-25 yang melihat pesanan konsumen untuk menghitung optimasi penjadwalan dengan melihat proses produksi dan keuntungan yang didapatkan.

Proses yang dilakukan dalam analisis perhitungan menggunakan Algoritma *Active Schedule* diawali dengan menentukan produk yang dipesan, setelah produk ditentukan dilanjutkan dengan analisis waktu proses produksi, bahan kebutuhan dan biaya produksi agar mendapatkan penjadwalan yang optimal. Hasil dari analisis diperoleh bahwa jadwal yang optimal yaitu waktu proses produksi selama 505 menit dengan mendapatkan 454 *experience* dan mengeluarkan biaya produksi 84 koin. Sedangkan waktu yang diperoleh sebelum optimasi yaitu 775 menit dan waktu mendekati optimal menggunakan metode *Heuristic* yaitu 615 menit.

2.2. Pengertian Implementasi

Implementasi memiliki arti yaitu penerapan atau pelaksanaan. Suatu implementasi berakhir pada aktivitas, adanya aksi, atau mekanisme suatu sistem (Firdianti, 2018).

2.3. Pengertian Rancang Bangun

Rancang bangun adalah suatu istilah umum untuk membuat atau mendesain suatu objek dari awal pembuatan sampai akhir pembuatan (Fajriyah, Josi, & Fisika, 2017).

2.4. Konsep Dasar Sistem

Menurut (Sutabri, 2012) terdapat dua kelompok pendekatan dalam pendefinisian sistem, yaitu kelompok yang menekankan pada prosedur dan kelompok yang menekankan pada elemen atau komponennya. Pendekatan yang menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Pendekatan yang lebih menekankan pada elemen atau komponen mendefinisikan sistem sebagai kumpulan elemen yang berinteraksi untuk mencapai suatu tujuan

tertentu. Kedua kelompok definisi ini adalah benar dan tidak bertentangan. Yang berbeda adalah cara pendekatannya.

Selain terdapat kemiripan antara karakteristik mengenai sistem yang diungkapkan para ahli, secara garis besar prinsip sistem hampir sama dengan konsep dasar sistem pada umumnya. Hal itu diungkapkan (Fatta, 2017) yang menjelaskan secara ringkas bahwa sistem adalah:

1. Komponen-komponen yang saling berhubungan antara satu dengan yang lain.
2. Suatu keseluruhan tanpa memisahkan komponen pembentuknya.
3. Bersama-sama dalam mencapai tujuan.
4. Memiliki *input* dan *output* yang dibutuhkan oleh sistem lainnya.
5. Terdapat proses yang mengubah *input* menjadi *output*.
6. Memiliki aturan.
7. Memiliki subsistem yang lebih kecil.
8. Memiliki diferensiasi antar subsistem.
9. Memiliki tujuan yang sama meskipun dimulai secara berbeda.

2.4.1. Pengertian Sistem

Secara garis besar sistem adalah suatu kumpulan dari dua atau lebih komponen yang saling berinteraksi dan bekerja sama satu dengan yang lain untuk mencapai suatu kegiatan atau tujuan tertentu. Hal tersebut seperti yang diutarakan oleh (Yakub, 2012) bahwa suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan tertentu.

Pendapat serupa juga dikemukakan oleh (Susanto, 2013) yang menyatakan bahwa sistem adalah suatu kumpulan atau grup dari sistem atau bagian atau komponen apapun baik fisik atau non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu. Suatu sistem dapat diterapkan pada berbagai bidang seperti bidang komputer berupa sistem informasi, bidang keuangan berupa sistem akuntasi, dan sebagainya.

2.4.2. Karakteristik Sistem

Suatu ciri yang khas yang terdapat pada seseorang atau sesuatu disebut dengan karakteristik. Suatu sistem memiliki ciri khas yang berbeda dari unsur yang lain. Karakteristik sebuah sistem dikemukakan oleh (Sutabri, 2012) yang menjelaskan bahwa sistem memiliki karakteristik yang sederhana ataupun yang rumit. Karakteristik yang sederhana dari sebuah sistem adalah bahwa suatu hal dapat disebut dengan sistem apabila terdiri dari *input*, *process*, dan *output*. Selain dari 3 komponen tersebut, karakteristik sistem dapat di-breakdown menjadi hal yang lebih kompleks, yaitu:

1. Mempunyai Komponen Sistem (*Component System*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan.

2. Mempunyai Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya.

3. Mempunyai Lingkungan Luar Sistem (*Environment*)

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

4. Mempunyai Penghubung (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut dengan penghubung sistem atau interface.

5. Mempunyai Masukan (*Input*)

Masukan atau *input* merupakan *energy* yang dimasukan dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah suatu bahan yang dimasukan agar sistem dapat beroperasi, sedangkan *signal input* adalah suatu masukan yang diproses untuk mendapatkan keluaran. Sebagai contoh dalam sistem komputer, *program* merupakan *maintenance input* yang digunakan

untuk mengoperasikan komputer sedangkan *data* adalah *signal input* yang dapat diolah menjadi informasi.

6. Mempunyai Pengolahan (*Processing*)

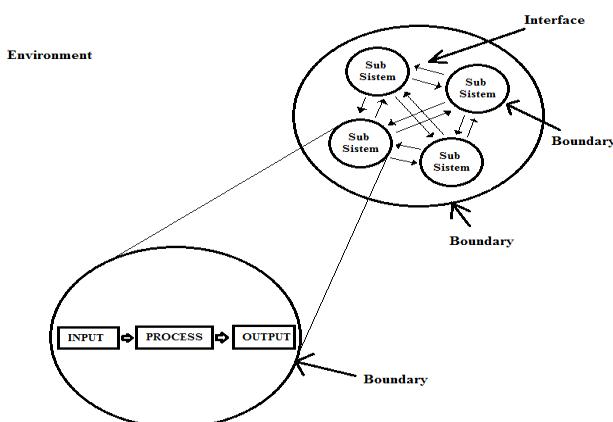
Pengolahan sistem (*process*) merupakan bagian yang melakukan perubahan dari masukan untuk dijadikan keluaran yang diinginkan.

7. Mempunyai Keluaran (*Output*)

Keluaran atau *output* sistem merupakan hasil dari pemrosesan. Keluaran dapat berupa informasi sebagai masukan dari perubahan untuk dijadikan keluaran lain yang diinginkan.

8. Mempunyai Sasaran (*Objective*) dan Tujuan

Suatu sistem mempunyai tujuan dan sasaran yang pasti dan bersifat deterministik. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.



Gambar II.1 Karakteristik Sistem
Sumber: (Sutabri, 2012)

2.4.3. Klasifikasi Sistem

Selain karakteristik (Sutabri, 2012) menjelaskan bahwa suatu sistem dapat diklasifikasikan dari berbagai sudut pandang, di antaranya adalah:

1. Sistem Abstrak (*Abstract System*) dan Sistem Fisik (*Physical System*)

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

Salah satu contohnya adalah sistem komputer, sistem produksi, sistem penjualan, dsb.

2. Sistem Alamiah (*Natural System*) dan Sistem Buatan Manusia (*Human Made System*).

Sistem alamiah adalah sistem yang keberadaannya terjadi karena proses alam, bukan buatan manusia seperti sistem perputaran bumi atau sistem pergantian musim. Sedangkan sistem buatan manusia (*human made system*) adalah sistem yang terjadi melalui rancangan atau campur tangan manusia.

3. Sistem Tentu (*Deterministic System*) dan Sistem Tak Tentu (*Probabilistic System*)

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut dengan sistem deterministik. Sedangkan sistem probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

4. Sistem Tertutup (*Closed System*) dan Sistem Terbuka (*Open System*)

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya.

2.4.4. Daur Hidup Sistem

Menurut (Rusdiana & Irfan, 2014), daur hidup sistem terbagi menjadi lima poin penting, yaitu:

1. Mengenali Adanya Kebutuhan

Sebelum segala sesuatu terjadi, akan timbul kebutuhan yang harus dapat dikenali sebagaimana adanya. Kebutuhan dapat terjadi sebagai hasil perkembangan dari organisasi dan *volume* yang meningkat melebihi kapasitas sistem yang ada. Semua data ini harus didefinisikan dengan jelas. Tanpa adanya kejelasan dari kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektivitas.

2. Pembangunan Sistem

Suatu proses atau seperangkat prosedur yang harus diikuti untuk menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk dapat memenuhi kebutuhan tersebut.

3. Pemasangan Sistem

Setelah tahap pembangunan sistem selesai, sistem akan dioperasikan. Pemasangan sistem merupakan tahap yang penting dalam daur hidup sistem. Peralihan dari tahap pembangunan menuju tahap operasional terjadi pemasangan sistem sebenarnya, yang merupakan langkah akhir pembangunan sistem.

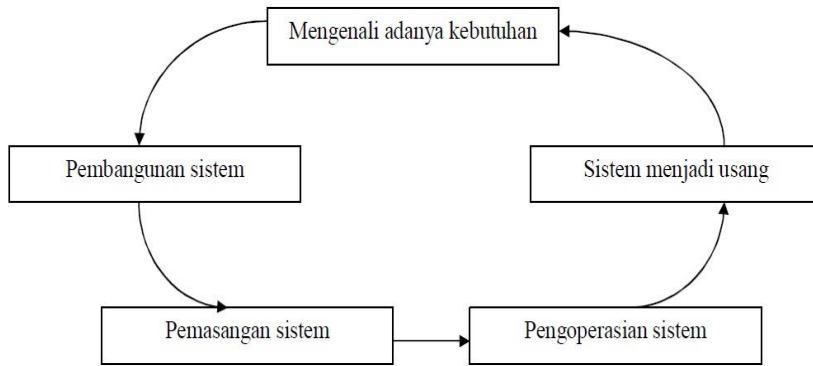
4. Pengoperasian Sistem

Program-program komputer dan prosedur-prosedur pengoperasian yang membentuk sistem informasi semuanya bersifat statis, sedangkan organisasi yang ditunjang oleh sistem informasi selalu mengalami perubahan-perubahan karena pertumbuhan kegiatan bisnis, perubahan peraturan, dan kebijaksanaan ataupun kemajuan teknologi. Untuk mengatasi perubahan-perubahan tersebut, sistem harus diperbaiki dan diperbarui.

5. Sistem Menjadi Usang

Terkadang perubahan terjadi begitu cepat sehingga tidak dapat diatasi hanya dengan melakukan perbaikan-perbaikan pada sistem yang berjalan. Dengan demikian, secara ekonomis dan teknis sistem yang ada tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya.

Sistem informasi kemudian akan melanjutkan daur hidupnya. Sistem dibangun untuk memenuhi kebutuhan. Sistem beradaptasi terhadap aneka perubahan lingkungan yang dinamis hingga kemudian sampai pada kondisi di mana sistem tidak dapat lagi beradaptasi. Sistem baru kemudian dibangun untuk menggantikannya. Daur sistem ini dapat dilihat pada Gambar II.2 berdasarkan (Sutabri, 2012).



Gambar II.2 Daur Hidup Sistem

Sumber: (Sutabri, 2012)

2.5. Informasi

Informasi merupakan hal penting yang harus dimiliki oleh seseorang atau perusahaan atau organisasi terkait bidang yang dijalankan. Pada bidang otomotif seperti teknisi kendaraan menganggap spesifikasi, riwayat perbaikan dan kondisi kendaraan sebagai informasi penting. Hal tersebut dianggap berbeda oleh orang yang berkecimpung dibidang kedokteran karena memiliki fokus yang berbeda.

2.5.1. Pengertian Informasi

Menurut (Sutarman, 2012), Informasi adalah sekumpulan fakta atau data yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi si penerima. Pendapat lain yang serupa juga dikemukakan oleh Davis, Gordon B yang telah dialih bahasa oleh (Mardi, 2014) bahwa informasi adalah data yang telah diolah ke dalam suatu bentuk yang berguna bagi penerimanya dan nyata atau berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa depan.

Dari kedua definisi diatas, dapat digaris bawahi bahwa suatu informasi dapat memiliki arti bagi penerima apabila informasi tersebut dapat diolah oleh penerima. Suatu fakta atau data tidak dapat menjadi informasi apabila penerima tidak dapat mengolah data tersebut. Hal tersebut seperti yang telah dijelaskan sebelumnya menggunakan perumpamaan teknisi kendaraan yang menganggap data kendaraan sebagai sebagai suatu informasi penting tetapi tidak demikian dengan dokter karena dokter tidak dapat mengolah data tersebut.

2.5.2. Karakteristik Informasi

Informasi merupakan hasil dari proses pengolahan dari data atau fakta. Informasi yang baik adalah informasi yang memiliki karakteristik tertentu di dalamnya. Menurut Romney yang dialih telah dialih bahasa oleh (Mardi, 2014) informasi memiliki karakteristik sebagai berikut:

1. Relevan

Informasi harus memiliki makna yang tinggi sehingga tidak menimbulkan keraguan bagi yang menggunakannya dan dapat digunakan secara tepat untuk membuat keputusan.

2. Andal

Suatu informasi harus memiliki keterandalan yang tinggi. Informasi yang dijadikan alat pengambilan keputusan merupakan kejadian nyata dalam aktivitas perusahaan. Informasi yang tidak andal tidak dapat dijadikan sebagai acuan dalam pengambilan keputusan.

3. Lengkap

Informasi harus memiliki penjelasan yang rinci dan jelas dari setiap aspek peristiwa yang dilakukannya.

4. Tepat Waktu

Setiap informasi harus dalam kondisi yang terbaru dan tidak dalam bentuk yang usang, sehingga penting untuk digunakan sebagai pengambil keputusan.

5. Dapat Dipahami

Informasi yang disajikan dalam bentuk yang jelas akan memudahkan orang dalam menginterpretasikannya.

6. Dapat Diverifikasi

Informasi yang baik tidak memiliki arti yang ambigu. Informasi harus memiliki kesamaan pengertian dalam pemakaianya.

Pendapat yang serupa akan karakteristik dari suatu informasi juga dikemukakan oleh Mc Leod yang pendapatnya kemudian dialihbahasakan oleh (Susanto, 2013) mengatakan bahwa suatu informasi harus memiliki ciri-ciri sebagai berikut:

1. Akurat
Memiliki arti bahwa Informasi harus mencerminkan keadaan yang sebenarnya.
2. Tepat Waktu
Memiliki arti bahwa informasi harus tersedia atau ada pada saat informasi tersebut diperlukan.
3. Relevan
Memiliki arti bahwa informasi yang diberikan harus sesuai dengan yang dibutuhkan oleh individu yang menerimanya.
4. Lengkap
Memiliki arti bahwa informasi harus diberikan secara lengkap. Seperti contohnya, suatu informasi mengenai penjualan harus mencantumkan periode penjualan.

2.6. Konsep Dasar Sistem Informasi

Suatu organisasi terdiri dari banyak unsur yang di dalamnya terdapat orang yang masing-masing memiliki tugas, wewenang dan tanggung jawab yang berbeda-beda. Meskipun memiliki tugas dan wewenang yang berbeda suatu organisasi memiliki tujuan yang sama yaitu keberhasilan organisasinya. Dengan tujuan yang sama tersebut diperlukan suatu sistem yang dapat menghubungkan informasi dalam organisasi tersebut. Oleh karena itu, organisasi mengimplementasikan sistem informasi. Sistem informasi merupakan penerapan sistem di dalam organisasi untuk mendukung informasi yang dibutuhkan oleh semua tingkat manajemen.

Menurut (Gelinas & Dull, 2012) sistem informasi adalah suatu kumpulan elemen atau komponen yang berupa manusia, prosedur, database dan alat yang saling terkait untuk memproses, menyimpan serta menghasilkan informasi untuk mencapai suatu tujuan. Selain itu, (Stair & Reynolds, 2010) mengatakan bahwa sistem informasi merupakan suatu perangkat elemen atau komponen yang saling terkait satu sama lain, yang dapat mengumpulkan, mengolah, menyimpan dan juga menyebarkan data dan juga informasi, serta mampu untuk memberikan *feedback* untuk memenuhi tujuan suatu organisasi.

Berdasarkan beberapa pendapat oleh para ahli sebelumnya, dapat diambil kesimpulan bahwa sistem informasi merupakan kumpulan atau perangkat elemen dari suatu komponen yang saling terkait untuk melakukan tugasnya dengan hasil akhir yaitu mencapai tujuan yang telah ditentukan. Tiap-tiap perusahaan mempunyai perangkat sistem informasi yang berbeda-beda dengan tujuan yang berbeda pula, meskipun begitu tak bisa dipungkiri bahwa sistem informasi merupakan suatu hal yang tidak dapat dipisahkan dari perusahaan.

2.6.1. Komponen Sistem Informasi

Menurut (Sutabri, 2012) sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*). Komponen tersebut terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. Keenam blok tersebut kemudian saling berinteraksi satu dengan yang lain dan membentuk satu kesatuan untuk mencapai sasaran.

1. Blok Masukan (*Input Block*)

Input mewakili data yang masuk ke dalam sistem informasi. Yang dimaksud dengan *input* di sini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen- dokumen dasar.

2. Blok Model (*Model Block*)

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

3. Blok Keluaran (*Output Block*)

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

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan *toolbox* dalam sistem informasi. Teknologi digunakan untuk menerima masukan, 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 (*Database Block*)

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan perangkat lunak yang digunakan untuk dimanipulasi. Data perlu disimpan 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 penyimpanan. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*database management system*).

6. Blok Kendali (*Control Block*)

Banyak hal dapat merusak sistem informasi baik itu yang berasal dari manusia (*human error*) atau yang berasal dari lingkungan (*environment*). Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dicegah dan apabila sudah terjadi maka kesalahan-kesalahan dapat dengan cepat diatasi.

2.7. Produksi

Produksi adalah kegiatan menambah nilai guna/manfaat dari suatu faktor produksi masukan atau membuat barang dan jasa baru untuk memenuhi kebutuhan. Penjelasan produksi tersebut seperti yang dikemukakan oleh (Heizer & Render, 2011) yaitu merupakan suatu kegiatan dalam menghasilkan barang atau jasa dengan cara merubah faktor-faktor produksi masukan menjadi keluaran. Fakto-faktor masukan tersebut antara lain adalah sebagai berikut:

1. Faktor Sumber Daya Alam
2. Faktor Sumber Daya Manusia
3. Faktor Modal
4. Faktor Kewirausahaan
5. Faktor Sumber Daya Informasi

2.8. Penjadwalan

Menurut (Pinedo, 2016) penjadwalan merupakan sebuah fungsi pengambilan keputusan yang biasa digunakan banyak perusahaan manufaktur atau jasa, yang berhubungan dengan alokasi sumber daya untuk mengerjakan tugas selama waktu tertentu dan tujuannya adalah untuk mengoptimalkan satu atau lebih tujuan pengoptimalan. Penjadwalan produksi dapat diklasifikasikan menjadi beberapa kriteria, yaitu:

1. Penjadwalan produksi secara umum

a. Penjadwalan maju (*forward scheduling*)

Operasi penjadwalan dimulai dari tanggal penerimaan pesanan secara maju. Konsekuensinya terjadi persediaan bahan baku sampai pada gilirannya diproses.

b. Penjadwalan mundur (*backward scheduling*)

Teknik penjadwalan dimulai dari waktu penyelesaian operasi terakhir. Keuntungannya adalah mengurangi persediaan barang setengah jadi (*work in process*).

2. Penjadwalan produksi berdasarkan mesin yang digunakan

a. Model *Single-Machine*

Model *single-machine* hanya berisi sejumlah m mesin yang sejenis, yang harus memproses sejumlah pekerjaan yang terdiri dari satu operasi. Setiap pekerjaan dapat dikerjakan pada salah satu mesin yang ada tersebut. Model ini banyak digunakan pada penjadwalan yang menggunakan metode dekomposisi, yaitu suatu model penjadwalan yang memecahkan permasalahan penjadwalan yang kompleks ke dalam sejumlah permasalahan yang lebih sederhana dalam bentuk *single-machine*.

b. Model *Parallel-Machine*

Merupakan suatu bentuk umum dari model *single-machine*. Model ini terdapat pada industri yang proses produksinya terdiri dari berbagai tahapan, di mana tiap tahapan tersebut terdiri dari sejumlah mesin sejenis yang tersusun secara paralel. Setiap pekerjaan dapat dikerjakan

pada salah satu mesin yang ada. Ada suatu kondisi di mana mesin-mesin yang tersusun secara paralel tersebut tidak identik satu sama lain dalam hal kemampuan berprosesnya.

c. Model *Flow Shop*

Suatu proses manufaktur seringkali harus melewati banyak operasi yang membutuhkan jenis mesin yang berbeda pada tiap operasinya. Jika rute yang harus dilewati untuk setiap pekerjaan adalah sama, maka bentuk konfigurasi ini disebut juga dengan model *flow shop*. Mesin-mesin pada model ini disusun secara seri dan pada saat sebuah pekerjaan selesai diproses pada satu mesin, maka pekerjaan tersebut akan meninggalkan mesin tersebut untuk kemudian mengisi antrian pada mesin berikutnya untuk diproses.

d. Model *Job Shop*

Dalam suatu proses manufaktur yang memerlukan banyak operasi, seringkali rute yang harus dilalui setiap pekerjaan (*job*) tidak sama. Model seperti ini disebut dengan model *job shop*. Bentuk sederhana dari model ini mengasumsikan bahwa setiap pekerjaan hanya melewati satu jenis mesin sebanyak satu kali dalam rutennya pada proses tersebut. Namun ada juga model lainnya di mana setiap pekerjaan diperbolehkan untuk melewati mesin sejenis lebih dari satu kali pada rutennya. Model ini disebut juga dengan *job shop recirculation* (pengulangan).

3. Penjadwalan produksi berdasarkan pola kedatangan pekerjaan

a. Penjadwalan Statis

Menggunakan acuan pekerjaan yang datang bersamaan dan siap dikerjakan pada mesin yang tidak bekerja. Kondisi semua stasiun kerja dan perlengkapannya selalu tersedia pada saat itu.

b. Penjadwalan Dinamis

Menggunakan acuan pekerjaan yang datang secara terus-menerus pada waktu yang berbeda-beda. Pendekatan yang sering digunakan pada penjadwalan ini adalah penggunaan aturan *dispatching* yang berbeda untuk setiap stasiun kerja.

4. Penjadwalan produksi berdasarkan sifat informasi yang diterima
 - a. Penjadwalan Deterministik
Informasi yang diperoleh bersifat pasti, seperti waktu kedatangan *job*, waktu *setup* dan waktu proses.
 - b. Penjadwalan Stokastik
Informasi yang diperoleh tidak pasti, tetapi memiliki kecenderungan yang jelas atau menyangkut adanya distribusi probabilitas tertentu, misalnya kedatangan pekerjaan bersifat acak.

2.9. Active Schedule

Metode penjadwalan aktif (*active schedule*) adalah metode penjadwalan dengan satu set jadwal yang tidak memungkinkan lagi untuk dilakukan geser kiri global. Metode penjadwalan aktif sering kali digunakan pada penjadwalan dengan jenis model *Job Shop* yang dalam masing-masing proses produksi pada tiap produknya berbeda. Adapun langkah-langkah dalam penggunaan metode penjadwalan aktif adalah sebagai berikut (Aryono, 2011):

1. Memulai pekerjaan pada $t = 0$, karena belum ada proses pekerjaan yang dijadwalkan.
2. Menentukan awal job, operasi dan mesin yang akan dijadwalkan pada PSt.
3. Berdasarkan job yang akan dijadwalkan tentukan waktu mulai dan waktu proses sehingga diketahui waktu penyelesaiannya.
4. Menentukan waktu penyelesaian (r_j) yang paling minimum untuk job yang akan direalisasikan.
5. Menentukan job yang akan direalisasikan dengan waktu penyelesaian yang paling minimum pada *stage 1* dan dimasukkan dalam PSt.
6. Mengeluarkan job yang telah dijadwalkan dan memasukkan operasi selanjutnya dari job yang sama ke dalam St.
7. Kembali ke langkah 2 untuk setiap alternatif jadwal parsial PSt yang dapat dibuat pada langkah ke 3. Lanjutkan proses ini sampai selesai.

Keterangan akan notasi yang terdapat dalam penjelasan metode *Active Schedule* adalah sebagai berikut:

- *Stage*: Nomor tahapan.
- Mesin: Jumlah mesin yang dimiliki.
- *St*: Kolom dari operasi yang dijadwalkan.
- *Tij*: Waktu penyelesaian produksi.
- *rj*: Total waktu penyelesaian.
- *r**: Waktu paling minimal dalam deretan *rj*.
- *m**: Mesin yang digunakan.
- *Pst*: Jadwal Parsial dari waktu yang terjadwalkan.

2.9.1. Kelebihan dan Kekurangan *Active Schedule*

Adapun kelebihan dan kekurangan metode penjadwalan *active schedule* adalah sebagai berikut:

Kelebihan:

- Metode *Active Schedule* merupakan metode yang mudah dipahami karna alur perhitungannya yang menggunakan prioritas nilai terkecil sehingga tidak diperlukan perhitungan khusus untuk mencari nilai terlebih dahulu.

Kekurangan:

- Minimnya tulisan atau literatur yang menjelaskan aturan baku akan tahapan-tahapan perhitungan dalam penggunaan metode *active schedule* secara *detail*. Hal tersebut dapat dilihat dari banyaknya jurnal-jurnal yang memiliki teknis perhitungan yang berbeda-beda.
- Tidak dapat digunakan pada suatu sistem dengan mesin paralel (tanpa menggabungkannya dengan metode lain).

2.10. System Development Life Cycle (SDLC)

System development life cycle (SDLC) adalah proses untuk memahami bagaimana sebuah sistem informasi dapat mendukung kebutuhan bisnis dengan merancang suatu sistem, membangun sistem tersebut kemudian menyampaikannya kepada pengguna (Dennis, Wixom, & Roth, 2013).

SDLC terdiri dari beberapa fase seperti perencanaan, analisis, desain, implementasi serta pemeliharaan. Setiap fase tersebut memiliki langkah beserta penjelasan yang berbeda dalam menghasilkan tujuan yang ingin dicapai. Berikut merupakan penjelasan dari tiap-tiap fase yang terdapat pada SDLC.

1. Perencanaan (*Planning*)

Merupakan tahap awal pengembangan sistem dengan mendefinisikan perkiraan kebutuhan-kebutuhan sumber daya. Dalam tahapan ini juga ditentukan langkah-langkah berupa: Mendefinisikan masalah, menentukan tujuan sistem, mengidentifikasi kendala sistem dan membuat studi kelayakan.

2. Analisis (*Analysis*)

Tahap mempelajari sistem informasi yang sedang berjalan. Berguna untuk mengetahui sebab dan akibat yang ditimbulkan oleh masalah, sehingga akan menghasilkan informasi yang mengungkapkan suatu permasalahan yang dapat atau sedang terjadi.

3. Perancangan (*Design*)

Tahap selanjutnya yaitu merancang sistem. Perancangan suatu sistem dilakukan dengan memahami bagaimana menterjemahkan keinginan dan kebutuhan pemakai dari sistem informasi tersebut ke dalam bahasa komputer. Dalam memulai perancangan dibutuhkan alat bantu UML yaitu dengan menggunakan *Software Visual Paradigm* seperti *Use Case Diagram*, *Class Diagram*, *Sequence Diagram*, *Statechart Diagram* dan *Activity Diagram*.

4. Penerapan (*Implementation*)

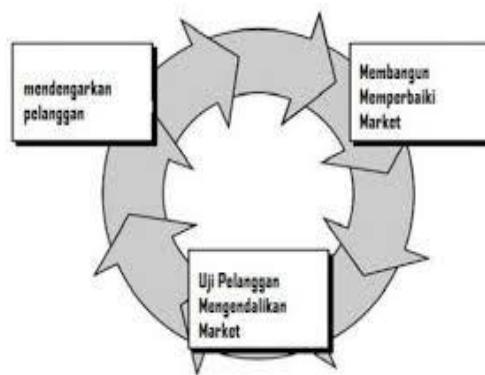
Setelah dilakukan perancangan maka selanjutnya dilakukan tahap penerapan sistem tersebut. Suatu sistem informasi yang telah disusun kemudian dibuat menjadi sebuah *software* komputer yang siap digunakan oleh pengguna sesuai dengan spesifikasi atau kebutuhan yang telah diminta.

5. Pemeliharaan (*Maintenance*)

Pemeliharaan *System Development Life Cycle* (SDLC) yang dilakukan adalah dengan melakukan perbaikan pada kesalahan atau kegagalan yang timbul pada saat pengguna menggunakan sistem informasi tersebut.

2.10.1. Model *Prototype*

Model *prototype* merupakan salah satu jenis yang sering digunakan ketika hendak membuat sistem. Suatu model *prototype* dapat digunakan untuk menyambungkan ketidakpahaman pelanggan terhadap hal teknis yang memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Rosa & Salahuddin, 2011). Beberapa tahapan (dapat dilihat pada Gambar II.3) pada model pengembangan sistem jenis *prototype* adalah sebagai berikut:



Gambar II.3 SDLC Model *Prototype*

Sumber: (Rosa & Salahuddin, 2011)

1. Mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat.
2. Membuat *prototype* agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program *prototype* merupakan suatu program yang belum sempurna, meskipun begitu terdapat fungsi-fungsi utama yang diinginkan pelanggan.
3. Program *prototype* selanjutnya dievaluasi oleh pelanggan atau pengguna sampai program tersebut sesuai dengan spesifikasi yang diinginkan.

2.10.2. Kelemahan dan Penyelesaian Model *Prototype*

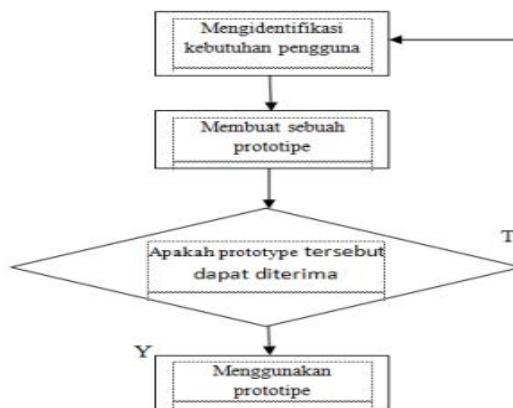
Terlepas dari kelebihan untuk pelanggan atau pengguna berupa dapat dengan mudah dipahami hasil akhir dari suatu program. Model *prototype* juga memiliki kelemahan tersendiri jika dibandingkan dengan model lain seperti model *waterfall*. Kelemahan model *prototype* antara lain adalah sebagai berikut (Rosa & Salahuddin, 2011):

- Pelanggan dapat sering mengubah atau menambah spesifikasi kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangkan. Hal tersebut sering membuat pengembang mengalah kepada pengguna yang kemudian melakukan penambahan spesifikasi terlepas dari kebutuhan pengguna pada awal perjanjian.

Beberapa penyelesaian dalam pembuatan *prototype* yang dapat digunakan oleh pengembang antara lain adalah sebagai berikut:

- Menentukan poin-poin utama dari permintaan pelanggan. Pengembang dapat mencatat poin utama permintaan pelanggan dalam nota kesepakatan (seperti *memorandum of understanding*) sehingga pelanggan tidak dapat semena-mena dalam mengubah atau menambah fitur dari program.
- Memberikan tenggat waktu untuk mendiskusikan fitur yang terdapat pada program. Tenggat waktu juga digunakan agar membuat pengembang dapat fokus dalam pembuatan program ketika spesifikasi sudah dijelaskan oleh pelanggan.

2.10.3. *Evolutionary Prototype*



Gambar II.4 Kerangka *Evolutionary Prototype*
Sumber: (Darmawan & Fauzi, 2013)

Evolutionary prototype merupakan salah satu jenis *prototype* (dapat dilihat pada Gambar II.4) yang terus menerus disempurnakan hingga memiliki fungsionalitas yang dibutuhkan pengguna. Menurut (Darmawan & Fauzi, 2013) terdapat empat langkah dalam pembuatan sistem menggunakan *evolutionary prototype*, yaitu:

1. Mengidentifikasi kebutuhan pengguna. Pengembang mewawancara pengguna untuk mendapatkan ide mengenai sistem seperti apa yang diminta.
2. Membuat satu *prototype*. Pengembang mempergunakan satu alat *prototyping* atau lebih untuk membuat *prototype*.
3. Mendemonstrasikan *prototype* kepada para pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan, jika belum maka *prototype* direvisi ulang hingga sesuai dengan kebutuhan pengguna.
4. Menggunakan *prototype* tersebut.

2.11. *Flowchart*

Flowchart (bagan alir) merupakan gambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program (Indrajani, 2011). Pendapat lain juga dikemukakan oleh (Romney & Steinbart, 2014) bahwa bagan alir merupakan teknik analisis bergambar yang digunakan untuk menjelaskan tentang prosedur yang terjadi di dalam perusahaan secara ringkas dan jelas. Berdasarkan pendapat para ahli dapat disimpulkan bahwa *flowchart* merupakan bagan alir yang menggunakan simbol dalam menampilkan gambaran dari langkah dan urutan suatu prosedur.

Simbol-simbol yang terdapat pada bagan alir dibagi menjadi 4 kategori seperti simbol masukan atau keluaran, simbol pemrosesan, simbol penyimpanan serta simbol lain-lain seperti simbol arus. Simbol masukan atau keluaran menampilkan masukan atau keluaran yang terdapat dalam sistem. Simbol pemrosesan menampilkan data yang sedang diproses di dalam sistem. Simbol penyimpanan menampilkan data yang disimpan perusahaan. Simbol lain-lain berisi simbol-simbol pelengkap dan keterangan akan data dalam sistem.

2.11.1. Jenis-Jenis *Flowchart*

Penjelasan akan jenis-jenis *flowchart* (bagan alir) yang umum digunakan saat ini adalah sebagai berikut:

1. Bagan Alir Sistem (*System Flowchart*)

Merupakan bagan alir yang menunjukkan arus pekerjaan secara keseluruhan dari suatu sistem.

2. Bagan Alir Dokumen (*Document Flowchart*)

Merupakan bagan alir yang menunjukkan arus dari laporan, formulir dan berkas lain termasuk tembusan-tembusannya. Bagan alir ini menggunakan simbol yang sama seperti bagan alir sistem. Bagan alir dokumen dan bagan alir sistem merupakan bagan alir yang seringkali digunakan pada saat pembuatan bagan alir (*flowchart*).

3. Bagan Alir Skematik (*Schematic Flowchart*)

Merupakan bagan alir yang mirip dengan bagan alir sistem. Hal yang membedakannya yaitu adanya penggunaan simbol tambahan berupa gambar komputer serta peralatan lain yang digunakan.

4. Bagan Alir Program (*Program Flowchart*)

Merupakan bagan alir yang terdiri dari dua macam, yaitu logika program (*program logic flowchart*) dan program komputer terinci (*detailed computer program flowchart*). Bagan alir logika program digunakan untuk menggambar langkah yang ada di dalam program komputer secara logika. Berbeda dengan program komputer terinci yang berisi bahasa pemrograman komputer. Bagan alir logika program dipersiapkan oleh analis sistem.

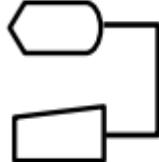
5. Bagan Alir Proses (*Process Flowchart*)

Merupakan bagan alir yang sering digunakan pada perusahaan yang di dalamnya terdapat hal-hal teknik industri. Bagan alir ini berguna dalam menggambarkan proses dalam suatu prosedur.

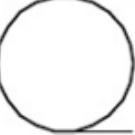
2.11.2. Simbol-Simbol *Flowchart*

Flowchart terdiri dari simbol-simbol dengan arti dan penjelasan yang berbeda-beda. Simbol tersebut memiliki fungsi sebagai alat bantu dalam menggambarkan proses-proses yang berkaitan dengan alur dalam suatu sistem. Meskipun bagan alir (*flowchart*) terdiri dari berbagai macam jenis, namun simbol-simbol yang terdapat di dalamnya memiliki beberapa persamaan. Berikut ini merupakan simbol-simbol dari bagan alir (dapat dilihat pada Tabel II.1) yang sering digunakan menurut (Romney & Steinbart, 2014).

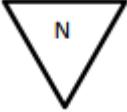
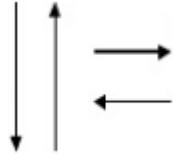
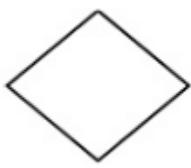
Tabel II.1 Simbol-Simbol *Flowchart*

No	Gambar Simbol	Nama Simbol	Deskripsi
Simbol Masukan atau Keluaran			
1		Simbol Dokumen	Merupakan simbol yang menyatakan masukan/ <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> yang dicetak ke kertas.
2		Simbol Dokumen beserta Tembusan	Merupakan simbol dokumen serta tembusan yang dituliskan dengan penomoran pada sisi dokumennya.
3		Simbol Masukan / Keluaran / Jurnal / Buku Besar	Merupakan simbol masukan atau keluaran yang dapat digunakan pada <i>program flowchart</i> . Simbol ini juga dapat digunakan sebagai simbol jurnal atau buku besar pada <i>document flowchart</i> .
4		Simbol Keluaran Elektronik	Merupakan simbol untuk menampilkan informasi ke terminal, monitor atau layar.
5		Simbol Masukan dan Keluaran Elektronik	Merupakan simbol yang digunakan untuk menunjukkan alat yang digunakan untuk keduanya.
6		Simbol Masukan Elektronik	Merupakan simbol yang digunakan untuk memasukkan data ke dalam komputer, monitor ataupun layar

Tabel II.1 Simbol-Simbol *Flowchart* (Lanjutan)

No	Gambar Simbol	Nama Simbol	Deskripsi
Simbol Pemrosesan			
1		Simbol Pemrosesan Komputer	Merupakan simbol pemrosesan yang dilakukan secara komputerisasi.
2		Simbol Pemrosesan Manual	Merupakan simbol pemrosesan yang dilakukan secara manual.
Simbol Penyimpanan			
1		Simbol Pita Magnetis	Merupakan simbol yang menyatakan data yang disimpan dalam pita magnetis.
2		Simbol Jurnal	Merupakan simbol catatan akuntansi berupa jurnal atau buku besar

Tabel II.1 Simbol-Simbol *Flowchart* (Lanjutan)

No	Gambar Simbol	Nama Simbol	Deskripsi
Lanjutan Simbol Penyimpanan			
3		Simbol Basis Data	Merupakan simbol yang menggambarkan basis data yang digunakan untuk menyimpan data.
4		Simbol Arsip Dokumen Sementara	Merupakan simbol yang menunjukkan arsip penyimpanan sementara. Ditandai dengan Nomor, Abjad atau Tanggal.
Simbol Lain-Lain			
1		Simbol Alur atau Arus	Merupakan simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan yang lain.
2		Simbol Keputusan	Merupakan simbol yang digunakan ketika terdapat pemilihan pada suatu proses.

Tabel II.1 Simbol-Simbol *Flowchart* (Lanjutan)

No	Gambar Simbol	Nama Simbol	Deskripsi
Lanjutan Simbol Lain-Lain			
3		Simbol Konektor Dalam-halaman	Merupakan simbol untuk keluar – masuk atau penyambung proses dalam lembar/halaman yang sama.
4		Simbol Konektor Luar-halaman	Merupakan simbol untuk keluar – masuk atau penyambung proses pada lembar/halaman yang berbeda.
5		Simbol Terminal	Merupakan simbol untuk memulai (<i>start</i>) atau mengakhiri (<i>stop</i>) suatu kegiatan atau sistem dalam suatu <i>flowchart</i> .

Sumber: (Romney & Steinbart, 2014)

2.12. Unified Modeling Language (UML)

Menurut (Rosa & Shalahuddin, 2014) UML (*Unified Modeling Language*) adalah suatu standar bahasa yang digunakan di dunia industri untuk mendefinisikan kebutuhan, membuat analisa dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Pendapat serupa juga dikemukakan oleh (Gata & Gata, 2013) bahwa UML (*Unified Modeling Language*) merupakan bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

Berdasarkan pendapat tersebut dapat disimpulkan bahwa UML merupakan suatu bahasa standar yang digunakan untuk merepresentasikan proses-proses ke dalam perangkat lunak yang berorientasi objek. Para pengembang menggunakan berbagai alat bantu berupa diagram dalam merancang perangkat lunak berorientasi objek menggunakan UML. Berikut ini merupakan beberapa jenis diagram beserta penjelasannya:

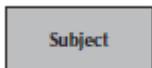
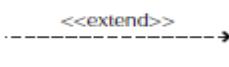
1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk menjelaskan tingkah laku akan sistem informasi yang hendak dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Rosa & Shalahuddin, 2014). Berikut adalah simbol-simbol yang ada pada diagram *use case* (Dennis, Wixom, & Tegarden, 2015):

Tabel II.2 Simbol-Simbol *Use Case Diagram*

No	Gambar Simbol	Nama dan Deskripsi Simbol
1	 Actor: <ul style="list-style-type: none"> • Sebuah orang atau sistem yang mengambil keuntungan dan merupakan bagian luar subjek. • Ditandai sesuai dengan perannya. • Diletakkan di luar batasan subjek (<i>boundary</i>). • Digambarkan antara menggunakan figur tongkat (<i>default</i>) atau apabila aktor bukan merupakan orang maka digambarkan dengan bangun kotak dengan tulisan <<aktor>> di dalamnya (<i>alternative</i>). 	
2	 Use Case: <ul style="list-style-type: none"> • Menggambarkan poin utama dalam sistem yang memiliki fungsi • Dapat memperpanjang dengan <i>use case</i> lain (<i>extend</i>). • Dapat menyertakan <i>use case</i> lain (<i>include</i>). • Diletakkan di dalam batasan subjek (<i>boundary</i>). <p>Diberi label berupa frase kata kerja deskriptif – kata benda.</p>	

Tabel II.2 Simbol-Simbol *Use Case Diagram* (Lanjutan)

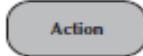
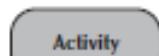
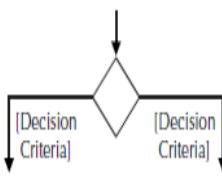
No	Gambar Simbol	Nama dan Deskripsi Simbol
3		Batasan Subjek (Boundary): <ul style="list-style-type: none"> Terdapat nama dari subjek di atas atau di dalam <i>boundary</i> Merepresentasikan ruang lingkup dari subjek misalkan suatu sistem atau bisnis proses individu.
3		Hubungan Asosiasi (Association): <ul style="list-style-type: none"> Menghubungkan aktor dengan <i>use case</i> yang digunakannya untuk berinteraksi.
4		Hubungan Penyertaan (Include) <ul style="list-style-type: none"> Merupakan penyertaan fungsionalitas dari suatu <i>use case</i> di dalam yang lain Terdapat tanda panah yang ditarik dari <i>use case</i> dasar menuju ke <i>use case</i> yang disertakan.
5		Hubungan Perpanjangan (Extend) <ul style="list-style-type: none"> Merupakan ekstensi dari <i>use case</i> untuk memasukkan perilaku opsional. Terdapat tanda panah yang ditarik dari <i>use case</i> ekstensi ke <i>use case</i> dasar.
6		Hubungan Generalisasi (Generalization) <ul style="list-style-type: none"> Menggambarkan <i>use case</i> yang khusus menuju yang lebih umum. Terdapat tanda panah yang ditarik dari <i>use case</i> khusus ke <i>use case</i> dasar.

Sumber: (Dennis, Wixom, & Tegarden, 2015)

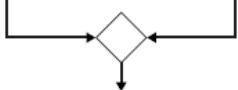
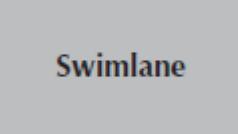
2. *Activity Diagram*

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis yang ada pada perangkat lunak. Perlu diperhatikan bahwa *activity diagram* menggambarkan aktivitas dari sebuah sistem bukan apa yang dilakukan oleh aktor (Rosa & Shalahuddin, 2014). Berikut ini merupakan simbol yang terdapat pada *activity diagram* (Dennis, Wixom, & Tegarden, 2015):

Tabel II.3 Simbol-Simbol *Activity Diagram*

No	Gambar Simbol	Nama dan Deskripsi Simbol
1		<p>Action Symbol:</p> <ul style="list-style-type: none"> • Merupakan simbol yang berisi perilaku sederhana yang tidak dapat dikompromikan. • Diberi label sesuai namanya.
1		<p>Activity Symbol:</p> <ul style="list-style-type: none"> • Merupakan simbol yang digunakan untuk mewakili serangkaian tindakan. • Diberi label sesuai dengan namanya.
2		<p>Initial Node:</p> <ul style="list-style-type: none"> • Merupakan simbol yang digunakan untuk menandai awal dimulainya tindakan atau kegiatan.
3		<p>Final-Activity Node:</p> <ul style="list-style-type: none"> • Merupakan simbol yang digunakan untuk menghentikan semua tindakan atau kegiatan.
4		<p>Final-Flow Node:</p> <ul style="list-style-type: none"> • Merupakan simbol yang digunakan untuk menhentikan aliran kontrol atau aliran objek secara khusus.
5		<p>Decision Node:</p> <ul style="list-style-type: none"> • Merupakan simbol yang digunakan untuk menggambarkan kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur. • Diberi label kriteria keputusan untuk melanjutkan ke jalur tertentu.

Tabel II.3 Simbol-Simbol *Activity Diagram* (Lanjutan)

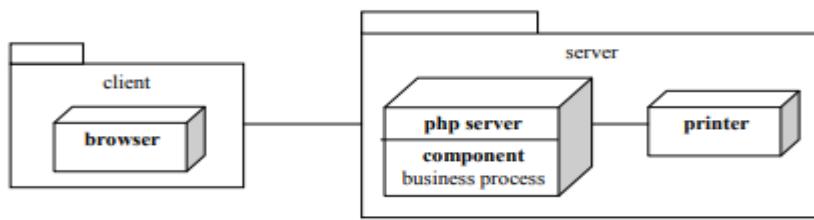
No	Gambar Simbol	Nama dan Deskripsi Simbol
6		<p>Merge Node:</p> <ul style="list-style-type: none"> Digunakan untuk menyambungkan jalur kondisi pengujian yang berbeda menjadi satu jalur.
7		<p>Fork Node:</p> <ul style="list-style-type: none"> Digunakan untuk membagi perilaku menjadi serangkaian kegiatan atau aktivitas paralel.
8		<p>Join Node:</p> <ul style="list-style-type: none"> Digunakan untuk menyambungkan serangkaian kegiatan atau aktivitas paralel menjadi satu.
9		<p>Swimlane:</p> <ul style="list-style-type: none"> Digunakan untuk memecah <i>activity diagram</i> menjadi baris dan kolom untuk menetapkan aktivitas individu kepada inividu atau objek yang bertanggung jawab untuk menjalankan aktivitas tersebut. Diberi label dengan nama individu atau objek yang bertanggung jawab.

Sumber: (Dennis, Wixom, & Tegarden, 2015)

3. Deployment Diagram

Deployment Diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. *Deployment diagram* juga dapat digunakan untuk memodelkan hal-hal berikut (Rosa & Shalahuddin, 2013):

- Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node* dan *hardware*.
- Sistem *client/server* seperti Gambar II.5 berikut:



Gambar II.5 Sistem *Client/Server Deployment Diagram*

Sumber: (Rosa & Shalahuddin, 2013)

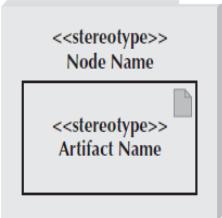
- c. Sistem terdistribusi murni.
- d. Rekayasa ulang aplikasi

Deployment diagram terdiri dari simbol sebagai penyusun diagramnya. Berikut merupakan simbol beserta deskripsi yang terdapat pada *deployment diagram* (Dennis, Wixom, & Tegarden, 2015).

Tabel II.4 Simbol-Simbol *Deployment Diagram*

No	Gambar Simbol	Nama dan Deskripsi Simbol
1		<p>Node Symbol:</p> <ul style="list-style-type: none"> • Merupakan sebuah sumber daya komputasi seperti komputer klien, server, jaringan terpisah atau perangkat jaringan individual. • Diberi label sesuai namanya. • Dapat berisi stereotip yang secara spesifik melabeli jenis node yang diwakili, misalnya perangkat, stasiun kerja klien, server aplikasi, perangkat seluler, dsb.
2		<p>Artifact Symbol:</p> <ul style="list-style-type: none"> • Merupakan spesifikasi perangkat lunak atau basis data misalnya, basis data, tabel, tampilan basis data, komponen perangkat lunak. • Diberi label nama. • Dapat berisi stereotip yang secara spesifik melabeli jenis artefak misalnya, file sumber, tabel basis data, file yang dapat dieksekusi, dll.

Tabel II.4 Simbol-Simbol *Deployment Diagram* (Lanjutan)

No	Gambar Simbol	Nama dan Deskripsi Simbol
3		<p>Node with Deployed Artifact Symbol: Menggambarkan artefak yang ditempatkan di node fisik.</p>
4	<u><<stereotype>></u>	<p>Communication Path Symbol:</p> <ul style="list-style-type: none"> • Merupakan hubungan antara dua <i>Node</i>. • Mengizinkan <i>Node</i> untuk bertukar pesan. • Dapat berisi stereotip yang secara spesifik melabeli jenis jalur komunikasi yang diwakili seperti LAN, Internet, serial, dan paralel.

Sumber: (Dennis, Wixom, & Tegarden, 2015)

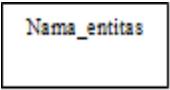
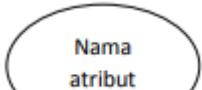
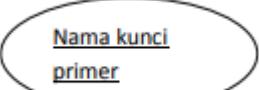
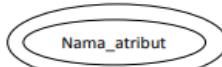
2.13. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan bentuk paling awal dalam melakukan perancangan basis data relasional. Penggunaan ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. Penggunaan himpunan bidang matematika yang umumnya digunakan yaitu hubungan *binary*. Pada hubungan *binary* digambarkan dengan hubungan dua buah entitas pada satu buah relasi (Rosa & Shalahuddin, 2013).

Meskipun begitu, penggunaan hubungan *ternary* yang menggambarkan hubungan tiga buah entitas ke dalam satu relasi atau *N-ary* yang menggambarkan hubungan banyak entitas ke dalam satu relasi dapat ditoleransi pada beberapa jenis metode perancangan ERD. Beberapa jenis aliran notasi yang dikenal dan digunakan pada pembuatan ERD antara lain adalah aliran notasi Chen yang dikembangkan oleh Peter Chen, aliran notasi Barker yang dikembangkan oleh Richard Barker, Ian Palmer dan Harry Ellis, dan aliran notasi lainnya. Berdasarkan aliran notasi tersebut, aliran notasi Chen merupakan notasi yang saat ini seringkali digunakan dalam pembuatan *Entity Relationship Diagram*.

Berikut merupakan notasi dan simbol yang terdapat pada notasi Chen (dapat dilihat pada Tabel II.5):

Tabel II.5 Simbol-simbol ERD

No	Simbol	Keterangan
1	Entitas 	Entitas merupakan data inti yang akan disimpan; penamaan entitas biasanya dituliskan menggunakan kata benda.
2	Atribut 	Atribut merupakan kolom data yang terdapat dalam suatu entitas.
3	Atribut Kunci Primer 	Atribut Primer merupakan kolom data yang terdapat dalam suatu entitas dan digunakan sebagai kunci akses dalam suatu entitas.
4	Atribut Multinilai 	Atribut Multinilai merupakan kolom data yang terdapat dalam suatu entitas yang dapat memiliki lebih dari satu nilai.
5	Relasi 	Relasi merupakan simbol yang menjelaskan hubungan antar entitas; biasanya berupa kata kerja.
6	Asosiasi 	Asosiasi merupakan garis penghubung antara relasi dengan entitas dimana di kedua ujungnya terdapat <i>multiplicity</i> yang menjelaskan akan jumlah penggunaan dari entitas terkait. Sebagai contoh terdapat kardinalitas (<i>multiplicity</i>) 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan sebuah entitas A dengan N entitas B.

Sumber: (Rosa & Shalahuddin, 2013)

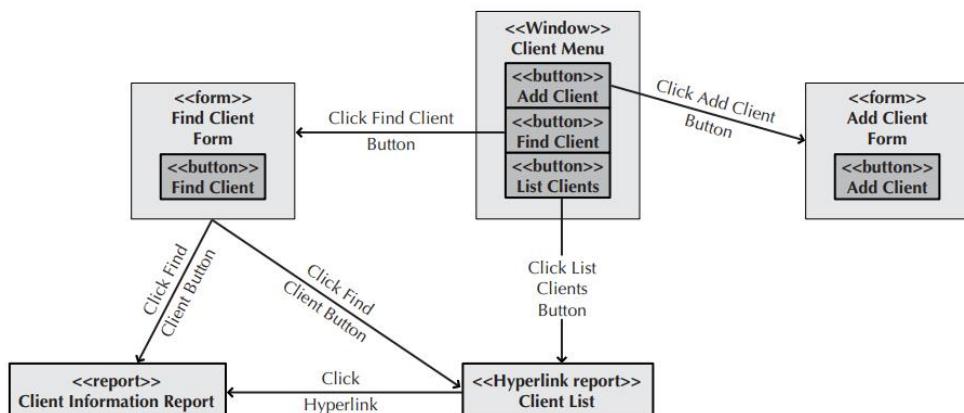
2.14. Kamus Data

Pada pemodelan sebuah sistem, kamus data digunakan sebagai alat untuk menjelaskan isi yang terkandung dalam basis data secara lengkap. Penjelasan akan kamus data yaitu merupakan kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan dan keluaran dapat dipahami secara umum (Rosa & Shalahuddin, 2013). Pada penggunaan kamus data, umumnya berisi:

- Nama dari sebuah data.
- Proses yang terkait dengan sebuah data.
- Deskripsi atau penjelasan dari sebuah data.
- Informasi tambahan seperti tipe data, nilai data, batas nilai data dan komponen yang membentuk sebuah data.

2.15. Windows Navigation Diagram (WND)

Windows Navigation Diagram merupakan struktur navigasi dalam sebuah program yang menentukan cara kerja fungsionalitas pada pengguna. WND (*Windows Navigation Diagram*) digunakan untuk menunjukkan bagaimana semua *interface*, *form*, dan *report* yang digunakan oleh sistem terkait dan bagaimana pengguna berpindah dari satu *interface* ke *interface* yang lain (Dennis, Wixom, & Tegarden, 2015). Contoh dari WND dapat dilihat pada Gambar II.6.



Gambar II.6 Contoh Windows Navigation Diagram
Sumber: (Dennis, Wixom, & Tegarden, 2015)

2.16. Hypertext Preprocessor (PHP)

PHP, singkatan dari PHP: *Hypertext Preprocessor* yaitu merupakan bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*serverside HTML embedded scripting*). PHP sering digunakan sebagai bahasa pemrograman untuk mengembangkan situs *web* dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu terbaru atau *up to date*. Semua *script* PHP dieksekusi pada *server* di mana *script* tersebut dijalankan (Anhar, 2010).

Kelebihan PHP dari Bahasa Pemrograman Lain:

- a. PHP adalah bahasa pemrograman yang tidak memerlukan kompilasi dalam penggunaanya.
- b. Terdapat banyak *Web Server* yang mendukung PHP seperti Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
- c. Berdasarkan segi pengembangan, penggunaan PHP termasuk mudah, karena banyaknya milis dan pengembang yang siap membantu.
- d. Berdasarkan segi pemahaman, PHP merupakan bahasa pemrograman yang paling mudah karena memiliki banyak referensi.
- e. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan melalui konsol serta juga dapat menjalankan perintah-perintah sistem.

2.17. Hypertext Markup Language (HTML)

HTML merupakan singkatan dari *Hypertext Markup Language*. HTML terdiri dari sekumpulan simbol-simbol atau *tag-tag* yang dituliskan dalam sebuah *file* yang digunakan untuk menampilkan halaman pada *web browser*. *Tag* pada HTML selalu diawali dengan <x> dan diakhiri dengan </x> dimana x *tag* HTML berisi seperti b (untuk menebalkan tulisan), i (untuk mencetak tulisan miring), u (untuk memberikan garis bawah pada tulisan), dsb (Anhar, 2010). Umumnya, penggunaan HTML berfungsi sebagai kerangka awal dari sebuah program yang di dalamnya dapat berisi Bahasa Pemrograman lain.

2.18. MariaDB

MariaDB adalah sebuah *database* manajemen sistem yang di kembangkan oleh pengembang MYSQL. Pengembang MYSQL membuat MariaDB karena MYSQL telah diakuisisi oleh perusahaan basis data bernama ORACLE sehingga MYSQL menjadi produk berlisensi dan akan menjadi produk komersial. Sebenarnya MariaDb ini mirip sekali dengan MYSQL karena di kembangkan dari MYSQL itu sendiri selain interfacenya mirip, mesin dan API nya sangat kompatible dengan MYSQL, Artinya semua connector, library dan aplikasi yang bekerja pada MySQL, dapat bekerja pada MariaDB. (Sidik, 2017).

2.19. Framework

Framework merupakan kerangka kerja atau sekumpulan *file-file* yang sudah ter-*include* yang mana di dalam *file* tersebut terdapat perintah kode program dan fungsi dasar untuk melakukan tugas tertentu (Purbadian, 2016). Berdasarkan pendapat tersebut dapat diambil kesimpulan bahwa seorang *programmer* tidak harus selalu menghafal Bahasa Pemrograman dari nol melainkan mengerti bagaimana logika suatu *program* bekerja dan melakukan penambahan berikutnya.

Kesimpulan tersebut seperti yang dikemukakan oleh (Saputra, 2011) bahwa seorang *programmer* tidak perlu membuat kode dari nol, karena *framework* telah menyediakan kerangka kerja dalam aplikasi *web* yang dimana di dalamnya memiliki potongan-potongan program yang telah disusun.

2.20. CodeIgniter

CodeIgniter adalah sebuah *framework* PHP yang dapat membantu mempercepat pengembang dalam mengembangkan aplikasi *web* berbasis PHP dibandingkan jika menulis semua kode program dari awal (Hakim, 2010).



Gambar II.7 Logo CodeIgniter
Sumber (Hakim, 2010)

CodeIgniter pertama kali dibuat oleh Rick Ellis, CEO Ellislab, Inc. (<http://ellislab.com>), sebuah perusahaan yang memproduksi CMS (*Content Management System*) yang cukup handal, yaitu Expression Engine (<http://www.expressionengine.com>). Saat ini, CodeIgniter dikembangkan dan dipelihara oleh *Expression Engine Development Team*.

Adapun beberapa keuntungan menggunakan CodeIgniter, diantaranya:

1. Gratis

CodeIgniter berlisensi di bawah Apache atau BSD opensource.

2. Ditulis menggunakan PHP 4

Meskipun CodeIgniter dapat berjalan di PHP 5, namun sampai saat ini kode program CodeIgniter masih dibuat dengan menggunakan PHP 4.

3. Berukuran Kecil

Ukuran CodeIgniter lebih kecil jika dibandingkan dengan *framework* lain.

4. Menggunakan Konsep MVC

CodeIgniter menggunakan konsep MVC yang memungkinkan pemisahan *layer application-logic* dan *presentation*.

5. URL yang Sederhana

Secara *default*, URL yang dihasilkan oleh CodeIgniter sangat bersih dan *Search Engine Friendly* (SEF).

6. Memiliki Paket *Library* yang Lengkap

CodeIgniter mempunyai *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis *web*.

7. *Extensible*

Sistem dapat dikembangkan dengan mudah menggunakan *plugin* dan *helper* atau dengan menggunakan *hooks*.

8. Tidak Memerlukan *Template Engine*

Meskipun CodeIgniter dilengkapi dengan *template parser* sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakaninya.

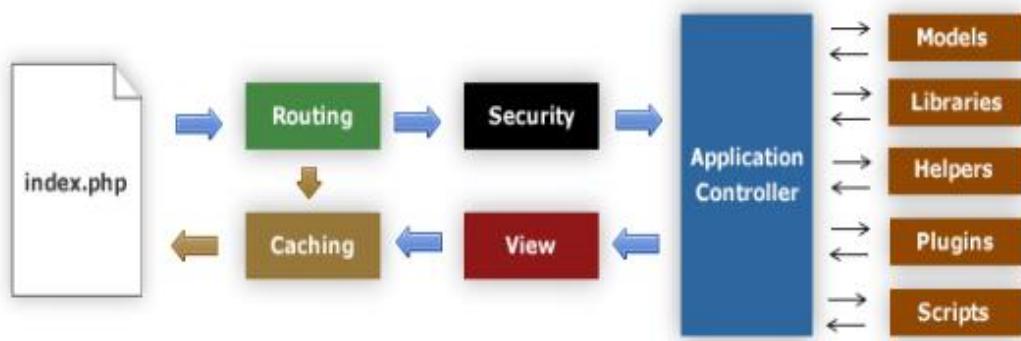
9. Dokumentasi Lengkap dan Jelas

CodeIgniter merupakan satu-satunya *framework* dengan dokumentasi yang lengkap dan jelas jika dibandingkan dengan *framework* lainnya.

10. Komunitas

Komunitas CodeIgniter saat ini berkembang pesat. Salah satu komunitasnya dapat dilihat di (<http://codeigniter.com/forum/>).

Proses aliran data aplikasi pada sistem dapat diilustrasikan seperti pada Gambar II.8 dibawah ini.



Gambar II.8 Application Flowchart

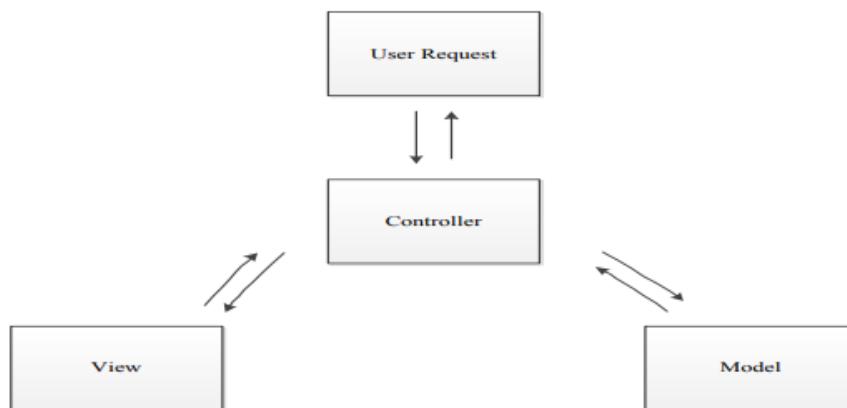
Sumber (Hakim, 2010)

Keterangan:

1. Index.php berfungsi sebagai *front controller*, menginisialisasi *base resource* untuk menjalankan CodeIgniter.
2. *Router* memeriksa HTTP *request* untuk menentukan apa yang harus dilakukan dengannya.
3. Jika *cache* aktif, maka hasilnya akan langsung dikirimkan ke *browser* dengan mengabaikan aliran data normal.
4. *Security*. Sebelum *controller* dimuat, HTTP *request* dan data yang dikirimkan *user* akan disaring untuk keamanan.
5. *Controller* memuat *model*, *core libraries*, *plugins*, *helpers*, dan semua *resource* lain yang diperlukan untuk memproses *request*.
6. Pada tahap akhir *view* akan dikirimkan ke *browser*. Jika *cache* aktif, maka *view* akan disimpan sebagai *cache* dahulu, sehingga pada *request* berikutnya langsung dapat ditampilkan.

2.21. MVC (*Model-View-Controller*)

CodeIgniter adalah *framework* PHP yang dibuat berdasarkan kaidah *model-view-controller*. Dengan MVC, maka memungkinkan pemisahan antara *layer application-logic* dan *presentation*. Sehingga, dalam sebuah pengembangan *web* seorang *programmer* bisa berkonsentrasi pada *core-system*, sedangkan *web designer* bisa berkonsentrasi pada tampilan *web*. Adapun alur program aplikasi berbasis *framework* CodeIgniter dapat dilihat pada Gambar II.9.



Gambar II.9 Application Flowchart

Sumber (Hakim, 2010)

Gambar diatas menampilkan bahwa ketika datang sebuah *user request*, maka akan ditangani oleh *controller*, kemudian *controller* akan memanggil *model* jika memang diperlukan operasi *database*. Hasil dari *query* oleh *model* kemudian dikembalikan ke *controller*. Pada tahapan selanjutnya *controller* akan memanggil *view* yang tepat dan mengkombinasikannya dengan hasil *query model*. Hasil akhir dari operasi ini akan ditampilkan dalam *browser*.

Dalam konteks CodeIgniter dan aplikasi berbasis *web*, penerapan konsep MVC mengakibatkan kode program dapat dibagi menjadi tiga kategori, yaitu:

1. *Model*

Kode program (berupa OOP *class*) yang digunakan untuk memanipulasi *database*.

2. *View*

Berupa *template* html/xml/php yang memiliki fungsi khusus untuk menampilkan data pada *browser*.

3. *Controller*

Kode program (berupa OOP *class*) yang digunakan untuk mengontrol aliran aplikasi (sebagai pengontrol *model* dan *view*).

2.22. *Testing*

Menurut (Nidhra & Dondeti, 2012) pengujian perangkat lunak merupakan teknik yang sering digunakan untuk verifikasi dan validasi kualitas suatu perangkat lunak. Pengujian perangkat lunak adalah prosedur untuk eksekusi sebuah program atau sistem dengan tujuan untuk menemukan kesalahan.

Kesalahan yang telah ditemukan setelah dilakukan *software testing* kemudian akan dipilah-pilah untuk kemudian diambil keputusan penyelesaian pada masing-masing jenis secara spesifik. Salah satu contoh metode untuk melakukan pencarian kesalahan (*testing*) yang sering digunakan ketika hendak mencari kesalahan suatu perangkat lunak atau sistem yaitu *Black-box Testing* (*Behavioral Testing*).

2.23. *Behavioral (Black-Box) Testing*

Black-box Testing merupakan jenis *testing* yang sering digunakan pada saat program sudah masuk ke dalam tahap *final*. Pada pengujian *black-box testing*, seorang *tester* (orang yang melakukan uji kelayakan) akan menguji semua fungsi yang terdapat dalam perangkat lunak.

Penjelasan tersebut seperti yang dikemukakan oleh (Nidhra & Dondeti, 2012) yang menyatakan bahwa *black-box testing* memiliki sebutan lain berupa *functional testing*, sebuah teknik pengujian fungsional yang merancang *test case* berdasarkan informasi dari spesifikasi perangkat lunak. Masing-masing *test case* akan mengangkat sebuah spesifikasi atau fitur utama dalam perangkat lunak yang terkandung dalam *use case* perangkat lunak.

BAB III

METODOLOGI PENELITIAN

3.1. Metodologi Penelitian

Metodologi penelitian adalah suatu cara yang dilakukan peneliti untuk memperoleh hasil atas penelitian yang dilakukan dengan menggunakan tata cara, rumus dan urutan-urutan yang bersifat ilmiah dan tidak mengada-ada.

3.2. Jenis dan Sumber Data

Data yang diperoleh dari dilakukannya penelitian ini ada dua jenis, yaitu data primer dan data sekunder. Sumber dari kedua jenis data tersebut diperoleh dari tempat dilakukannya praktik kerja lapangan yaitu PT Galih Ayom Paramesti:

1. Data Primer

Data primer adalah data yang diperoleh secara langsung di PT Galih Ayom Paramesti, dimana pengumpulan data atau informasi dilakukan dengan menanyakannya secara langsung kepada orang atau pihak terkait. Contoh data primer pada proses penjadwalan produksi antara lain, proses penjadwalan produksi saat ini, jenis prioritas penjadwalan produksi, proses bisnis sistem yang ada saat ini dan yang akan diusulkan, serta kebutuhan pengguna sistem.

2. Data Sekunder

Data sekunder adalah data yang diperoleh secara tidak langsung di PT Galih Ayom Paramesti, dimana pengumpulan data dilakukan dengan cara melakukan observasi dokumen atau literatur terkait perusahaan. Data sekunder yang terdapat pada laporan yaitu data umum perusahaan, profil perusahaan dan struktur organisasi perusahaan.

3.3. Metode Pengumpulan Data

Data yang diperoleh dikumpulkan dengan menggunakan beberapa metode serta pembahasan masalah selama dilakukannya praktik kerja lapangan pada PT Galih Ayom Paramesti. Metode yang digunakan dalam mengumpulkan data tersebut adalah sebagai berikut:

1. Studi Lapangan

Studi lapangan (*field research*) adalah pengumpulan data ke lapangan secara langsung pada objek yang diteliti dengan menggunakan metode sebagai berikut:

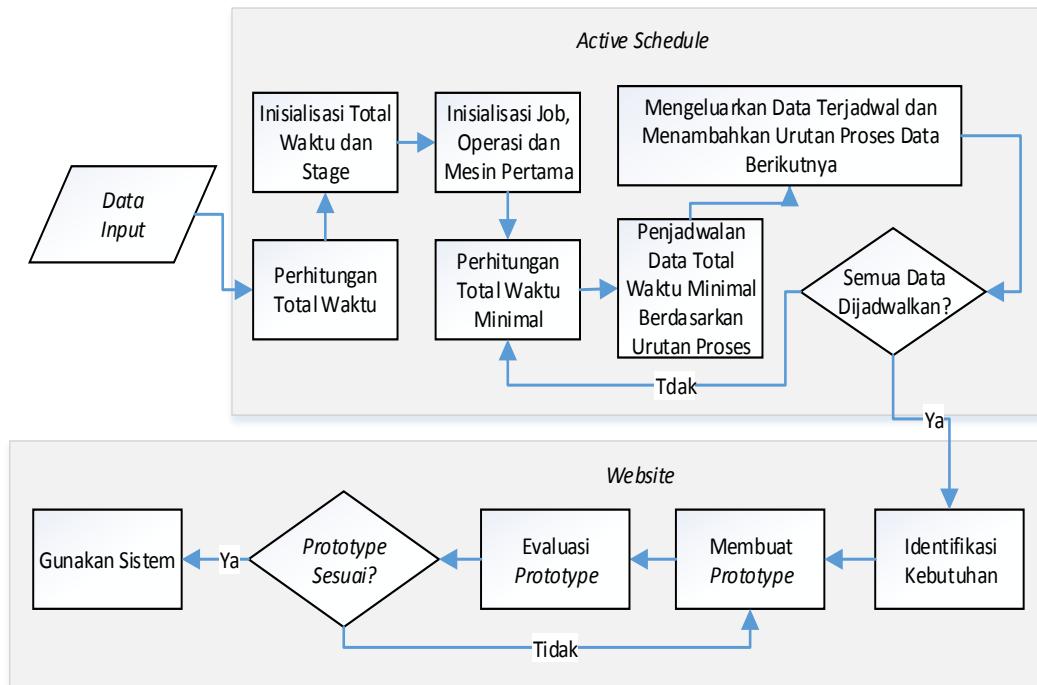
- a. Metode Pengamatan, metode ini dilakukan secara langsung dengan mengamati proses pembuatan jadwal pada PT Galih Ayom Paramesti.
- b. Metode Wawancara, metode ini dilakukan dengan memberikan pertanyaan terkait hal-hal yang dibutuhkan selama pembuatan Tugas Akhir. Wawancara dilakukan kepada Kepala Bagian PPC serta Wakil Direktur PT Galih Ayom Paramesti.
- c. Analisis Dokumen, metode ini dilakukan dengan mengamati dokumen yang ada di PT Galih Ayom Paramesti terkait sistem penjadwalan produksi.

2. Studi Pustaka

Studi pustaka (*literature review*) adalah metode pengumpulan data dengan mengamati literatur yang terdapat pada internet atau fisik yang memiliki hubungan dengan topik dan judul yang akan dibahas pada Tugas Akhir.

3.4. Kerangka Penelitian

Pada laporan Tugas Akhir ini, diuraikan langkah-langkah atau tahapan dilakukannya perancangan dan implementasi metode *Active Schedule* untuk menghasilkan urutan proses produksi pada sistem penjadwalan produksi di PT Galih Ayom Paramesti. Tahapan proses pada penelitian ini (dapat dilihat pada Gambar III.1) dimulai dari data *input*, perhitungan total waktu, inisialisasi total waktu dan stage, inisialisasi job, operasi dan mesin pertama, perhitungan total waktu minimal, penjadwalan data dengan total waktu minimal berdasarkan urutan proses, mengeluarkan data terpilih dan memasukkan data berdasarkan urutan proses berikutnya, melakukan perhitungan kembali hingga semua data terjadwalkan, analisis kebutuhan, perancangan *prototype*, evaluasi *prototype*, implementasi sistem, *testing*, evaluasi sistem serta menggunakan sistem.



Gambar III.1 Kerangka Penelitian
Sumber: Hasil Analisis (2019)

3.4.1. Data Input

Penelitian ini menggunakan data yang berasal dari dua bagian pada PT Galih Ayom Paramesti. Bagian tersebut yaitu Bagian PPC Manajer (*Head*) dan Bagian Teknisi atau Insinyur (*Engineering*). Selama dilakukannya penelitian, rincian data yang diperoleh yaitu data mesin berjumlah 11 mesin serta data produk berjumlah 3 produk beserta detail yang terkandung di dalamnya seperti data urutan proses dan waktu proses di setiap mesin. Data yang telah diperoleh selama dilakukannya penelitian pada PT Galih Ayom Paramesti dapat dilihat pada tabel di bawah ini:

Tabel III.1 Data Produk

No	Nama	Kode Job	Jumlah Produk
1	SLEEVE 0068	J1	200
2	SHAFT KICK 0014	J2	150
3	COLLAR 2561	J3	300

Sumber: Hasil Analisis (2018)

Tabel III.2 Data Mesin

No	Nama Mesin	Jumlah	Kode Mesin
1	BENCH LATE SD 32	2	M1
2	FONG HO	2	M2
3	CNC FOCUS 120	3	M3
4	BENCH LATE SD 25	6	M4
5	AMADA	1	M5
6	HOBING MC 2X	5	M6
7	CNC VICTOR	1	M7
8	WEST LAKES	2	M8
9	ROLLING MC	2	M9
10	CF 12A	5	M10
11	CNC FOCUS	2	M11

Sumber: Hasil Analisis (2018)

Tabel III.3 Data Urutan Mesin

Job	Proses								
	OP1	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9
J1	M1	M10							
J2	M2	M4	M6	M7	M11	M8	M9	M7	M10
J3	M2	M4	M3	M3	M5	M4			

Sumber: Hasil Analisis (2018)

Tabel III.4 Data Waktu Proses Produksi (detik)

Job	Waktu Proses (Detik)								
	OP1	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9
J1	28	26,4							
J2	28,8	30,5	184	54	60	25	18	24	59,4
J3	12,08	33	19,2	19,2	15,6	16,94			

Sumber: Hasil Analisis (2018)

3.4.2. Metode Active Schedule

Secara garis besar, tahapan dari metode *active schedule* yaitu perhitungan total waktu penyelesaian (t penyelesaian) dan penjadwalan metode *active schedule* berdasarkan total waktu penyelesaian yang telah diperoleh. Tahapan awal pada penjadwalan metode *active schedule* yaitu melakukan inisialisasi job, proses dan mesin yang digunakan pertama kali untuk kemudian dicari waktu minimalnya.

Waktu minimal tersebut diperoleh dengan membandingkan total waktu penyelesaian yang telah dihitung sebelumnya. Jika waktu penyelesaian minimal tersebut telah ditemukan maka job, proses dan mesin terkait waktu minimal tersebut akan dikeluarkan dan dimasukkan ke dalam jadwal. Data yang dikeluarkan tersebut kemudian akan diisi dengan urutan proses berikutnya pada job yang sama.

Proses pencarian tersebut dilakukan secara terus menerus hingga semua data berhasil dikeluarkan dan dimasukkan ke dalam jadwal. Dibawah ini merupakan rumus untuk mencari total waktu penyelesaian (t penyelesaian) beserta hasil keseluruhan pada data yang telah diperoleh setelah dilakukannya perhitungan t penyelesaian (dapat dilihat pada Tabel III.5).

$$TP_{Penyelesaian} = T_{proses} \times \frac{Jumlah\ Produksi}{Jumlah\ Mesin}$$

$$T(J1.OP1) = 28 \times \frac{200}{2}$$

J1.OP1 = Proses 1 dari JOB1 (Produk 1)

Tabel III.5 Waktu Penyelesaian

Job	Waktu Proses (Detik)								
	OP1	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9
J1	2800	1056							
J2	2160	762,5	5520	8100	4500	1875	1350	3600	1782
J3	1812	1650	1920	1920	4680	847			

Sumber: Hasil Analisis (2018)

3.4.3. Penjadwalan Metode *Active Schedule*

Tahapan awal dalam penyusunan jadwal dengan metode *active schedule* yaitu melakukan inisialisasi proses, job dan mesin pertama yang akan dijadikan acuan untuk selanjutnya dilakukan pencarian nilai minimal. Proses, job dan mesin pertama yang akan dijadikan acuan yaitu:

Tabel III.6 Inisialisasi Pertama

Job	Proses
	OP1
J1	2800
J2	2160
J3	1812

Sumber: Hasil Analisis (2019)

Proses tersebut kemudian dilanjutkan dengan mencari total waktu minimal berdasarkan urutan pertama yang telah diinisialisasi sebelumnya. Total waktu minimal diperoleh dengan menjumlahkan nilai dari waktu sebelumnya yang berkaitan dengan total waktu penyelesaian pada masing-masing urutan (dapat dilihat pada Tabel III.7).

Tabel III.7 Perhitungan Total Waktu Minimal

Stage	Mesin											St	Cj	Tij	rj	r*	m*	Pst
	1	2	3	4	5	6	7	8	9	10	11							
	0	0	0	0	0	0	0	0	0	0	0	111	0	2800	2800			312
												212	0	2160	2160			
												312	0	1812	1812	1812	2	

Sumber: Hasil Analisis (2019)

Adapun penjelasan akan notasi yang terdapat dalam metode *Active Schedule* adalah sebagai berikut:

- *Stage*: Nomor tahapan.
- Mesin: Jumlah mesin yang dimiliki.
- *St*: Kolom dari operasi yang dijadwalkan.
- *Tij*: Waktu penyelesaian produksi.
- *rj*: Total waktu penyelesaian.
- *r**: Waktu paling minimal dalam deretan *rj*.
- *m**: Mesin yang digunakan.
- *Pst*: Jadwal Parsial dari waktu yang terjadwalkan.

Proses yang memiliki nilai minimal kemudian dikeluarkan dan diganti dengan proses urutan berikutnya (dapat dilihat pada Tabel III.8). Nilai yang telah terpilih sebelumnya juga dijumlahkan pada job dan mesin yang sama. Proses ini dilakukan hingga semua job proses dijadwalkan (dapat dilihat pada Tabel III.9).

Tabel III.8 Pergantian Urutan Proses Terpilih

Stage	Mesin											St	Cj	Tij	rj	r*	m*	Pst
	1	2	3	4	5	6	7	8	9	10	11							
	0	1812	0	0	0	0	0	0	0	0	0	111	0	2800	2800	2800	1	111
												212	1812	2160	3972			
												324	1812	1650	3462			

Sumber: Hasil Analisis (2019)

Tabel III.9 Penyelesaian Metode Active Schedule Actual

Stage	Mesin											St	Cj	Tij	rj	r*	m*	Pst
	1	2	3	4	5	6	7	8	9	10	11							
0												111	0	2800	2800			312
	0	0	0	0	0	0	0	0	0	0	0	212	0	2160	2160			
1												312	0	1812	1812	1812	2	
	0	1811	0	0	0	0	0	0	0	0	0	212	1812	2160	3972			111
2												324	1812	1650	3462			
	2800	1811	0	0	0	0	0	0	0	0	0	212	1812	2160	3972			324
3												324	1812	1650	3462	3462	4	
	2800	1811	0	3462	0	0	0	0	0	0	0	212	1812	2160	3972			1210
4												333	3462	1920	5382			
	2800	1811	0	3462	0	0	0	0	0	0	0	212	1812	2160	3972	3972	2	212
5												333	3462	1920	5382			
	2800	3972	0	3462	0	0	0	0	0	0	0	224	3972	762.5	4734.5	4734.5	4	224
6												333	3462	1920	5382			
	2800	3972	0	4734.5	0	0	0	0	0	0	0	236	4734.5	5520	10254.5			333
7												333	3462	1920	5382	5382	3	343
	2800	3972	5382	4734.5	0	0	0	0	0	0	0	236	4734.5	5520	10254.5			
8												343	5382	1920	7302	7302	3	343
	2800	3972	7302	4734.5	0	0	0	0	0	0	0	236	4734.5	5520	10254.5	10254.5	6	236
9												355	7302	4680	11982			
	2800	3972	7302	4734.5	0	10254.5	0	0	0	0	0	247	10254.5	8100	18354.5			
10												355	7302	4680	11982	11982	5	355
	2800	3972	7302	4734.5	11982	10254.5	0	0	0	0	0	247	10254.5	8100	18354.5			
11												364	11982	847	12829	12829	4	364
	2800	3972	7302	12828	11982	10254.5	0	0	0	0	0	247	10254.5	8100	18354.5	18354.5	7	247
12												364	11982	847	12829	12829		
	2800	3972	7302	12828	11982	10254.5	18354.5	0	0	0	0	2511	18354.5	4500	22854.5	22854.5	11	2511
13												364	11982	847	12829	12829	8	268
	2800	3972	7302	12828	11982	10254.5	18354.5	0	0	0	0	268	22854.5	1875	24729.5	24729.5		
14												364	22854.5	1875	24729.5	24729.5	9	279
	2800	3972	7302	12828	11982	10254.5	18354.5	24729.5	0	0	0	279	24729.5	1350	26079.5	26079.5		
15												364	24729.5	1350	26079.5	26079.5	7	287
	2800	3972	7302	12828	11982	10254.5	18354.5	24729.5	26079.5	3856	22854.5	287	26079.5	3600	29679.5	29679.5		
16												364	26079.5	3600	29679.5	29679.5	10	2910
	2800	3972	7302	12828	11982	10254.5	29679.5	24729.5	26079.5	3856	22854.5	2910	29679.5	1782	31461.5	31461.5		
17												364	29679.5	1782	31461.5	31461.5	10	2910
	2800	3972	7302	12828	11982	10254.5	29679.5	24729.5	26079.5	31461.5	22854.5							

Sumber: (Hasil Analisis, 2019)

Berdasarkan perhitungan menggunakan metode *Active Schedule* pada data yang diperoleh dari penelitian langsung di lapangan, dapat diambil kesimpulan bahwa:

1. Urutan penggeraan proses yang direkomendasikan adalah 312*, 111, 324, 1210, 212, 224, 333, 343, 236, 355, 364, 247, 2511, 268, 279, 287, 2910.
2. Waktu total yang diperlukan untuk menyelesaikan pekerjaan produk Sleeve 0068, Shaft Kick 0014 dan Collar 2561 apabila dikerjakan secara bersama adalah sebesar 31.461,5 detik atau 524,35 menit atau ±8,7 jam.

* Dibaca job, urutan, mesin. Contoh 312 dibaca job ke-3, urutan nomor 1 dan mesin 2 (Fong Ho)

3.4.4. Identifikasi Kebutuhan

Pada tahap identifikasi kebutuhan akan didefinisikan format keseluruhan dari kebutuhan sistem dan perangkat lunak yang hendak dibuat. Proses pengumpulan data kebutuhan ini dilakukan dengan melakukan wawancara, observasi serta analisis dari dokumen terkait. Hasil dari proses pengumpulan data yaitu berupa permasalahan yang nantinya akan dijelaskan beserta proses bisnis yang sedang berjalan akan digambarkan dengan *flowmap*.

Permasalahan tersebut nantinya akan digunakan sebagai acuan untuk membantu dalam menjelaskan kebutuhan sistem yang baru. Gambaran dari kebutuhan sistem yang baru tersebut akan dijelaskan menggunakan *Unified Modeling Language (UML)* seperti *Use-case Diagram*, *Activity Diagram* serta *Deployment Diagram*.

3.4.5. Pembuatan Prototype

Pada tahap ini, pembuatan *prototype* akan dibuat perancangan basis data menggunakan *Entity Relationship Diagram (ERD)* dan Kamus Data, struktur navigasi menggunakan *Windows Navigation Diagram (WND)* serta perancangan tampilan pada program (*user interface*) sementara.

3.4.6. Evaluasi *Prototype*

Pada metode pengembangan sistem berjenis *evolutionary prototype*, evaluasi dilakukan dengan menambahkan perbaikan-perbaikan pada *prototype* yang sudah dibuat sebelumnya menjadi sebuah program. Program yang dibuat berdasarkan *prototype* sebelumnya dibuat menggunakan Bahasa Pemrograman PHP serta alat pendukung berupa *framework* bernama CodeIgniter.

Pada saat *prototype* awal dibuat ke dalam sebuah program akan dilakukan pengecekan akan kebutuhan dan spesifikasi yang diminta oleh perusahaan. Program yang memiliki kekurangan akan ditambahkan fitur baru namun apabila program sudah sesuai akan masuk ke tahap akhir. Program yang sudah pada tahap akhir kemudian akan dilakukan *testing* menggunakan *behavioural testing* atau *black-box testing* untuk mengetahui kinerja atau fungsi keseluruhan dari sebuah program.

3.4.7. Menggunakan Sistem

Program yang sudah melewati tahap *behavioural testing* dan tidak lagi memiliki kesalahan-kesalahan atau kekurangan fungsi dan fitur yang dibutuhkan perusahaan maka sudah dapat digunakan.

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1. Sejarah Perusahaan

PT Galih Ayom Paramesti merupakan perusahaan manufaktur yang bergerak di bidang industri otomotif. Didirikan sejak tahun 1992, PT Galih Ayom Paramesti memiliki fokus produksi yaitu pada *engine parts* untuk kendaraan roda dua dan roda empat. PT Galih Ayom Paramesti berlokasi pada Jl. Inspeksi Tarum Barat, Pekopen Lambang Jaya, Tambun, Bekasi, Jawa Barat. Berkat tingginya standar dan kualitas produk yang dihasilkan, PT Galih Ayom Paramesti diberikan kepercayaan oleh perusahaan yang telah dikenal oleh khalayak luas di Indonesia seperti PT Kawasaki Motor Indonesia.

4.2. Profil Perusahaan

Nama Perusahaan	: PT Galih Ayom Paramesti
Alamat Kantor	: Jl. Inspeksi Tarum Barat, Pekopen Lambang Jaya, Tambun, Bekasi, 17510, Jawa Barat, Indonesia
Telepon	: 021-8837 4577
Fax	: 021-8837 4576
Email	: Galihayom@yahoo.com
Produksi	: <i>Component Machining Automobile and Motorcycle</i>
Status	: Perseroan Terbatas
Luas Tanah	: 2500 m^2

4.3. Makna Logo Perusahaan

Berikut ini adalah logo perusahaan PT Galih Ayom Paramesti (dapat dilihat pada Gambar IV.1). Logo memiliki kegunaan yaitu sebagai identitas perusahaan serta cerminan dari visi misi yang ada di PT Galih Ayom Paramesti.



Gambar IV.1 Logo PT Galih Ayom Paramesti

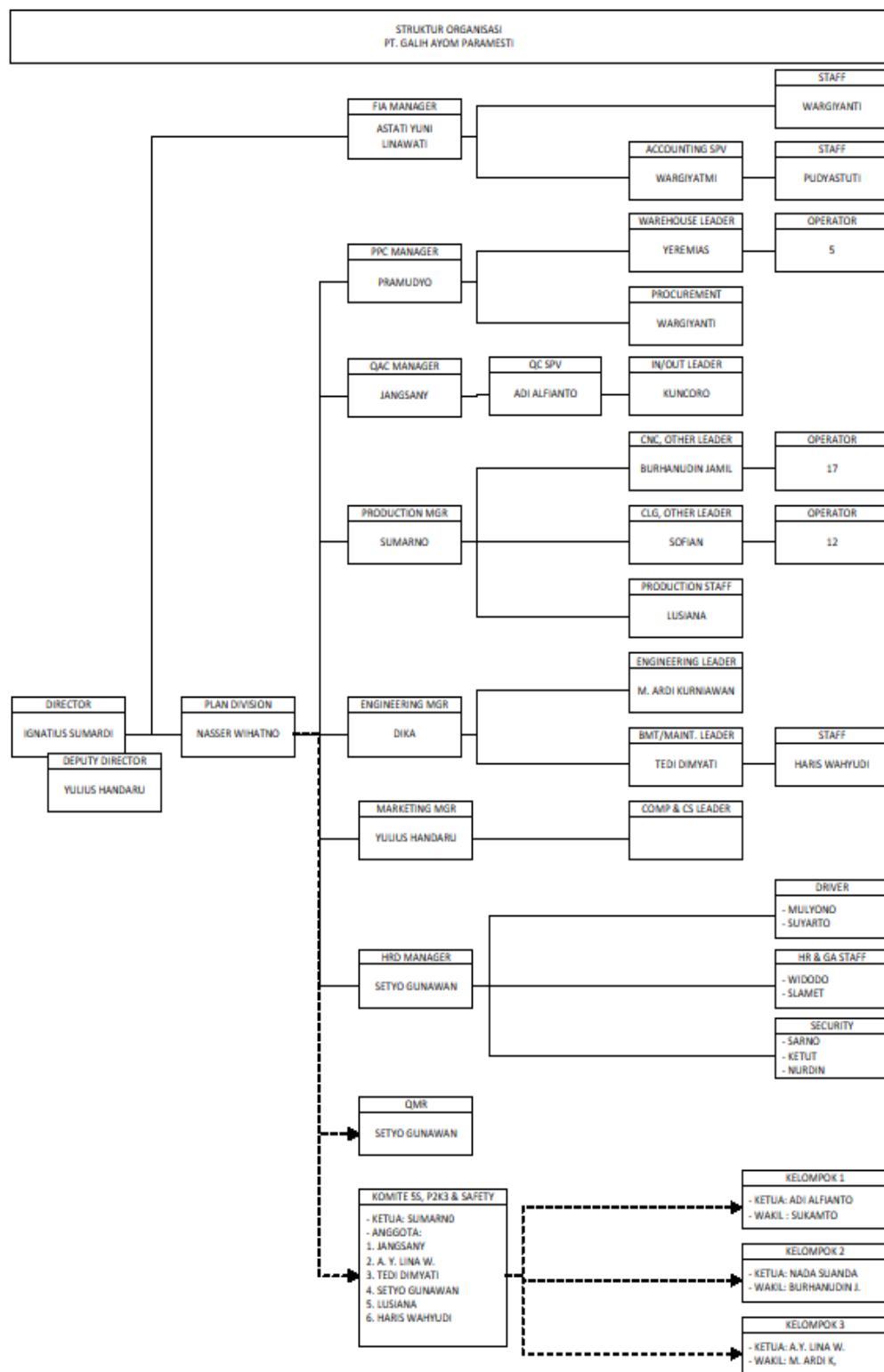
Sumber: PT Galih Ayom Paramesti (2018)

Pada logo tersebut digambarkan dengan lingkaran dan tulisan GAP yang masing-masing memiliki arti dan makna disetiap hurufnya. Huruf pertama memiliki arti Galih yang bermakna mencari dan melatih potensi diri. Huruf kedua memiliki arti Ayom yang memiliki makna mengayomi dengan segenap hati. Huruf terakhir memiliki arti Paramesti yaitu menjunjung nama Tuhan YME. Secara keseluruhan Galih Ayom Paramesti bermakna mencari dan melatih potensi diri, dengan cara mengayomi menggunakan segenap hati serta selalu menjunjung tinggi nama Tuhan YME.

4.4. Struktur Organisasi Perusahaan

Setiap perusahaan memerlukan suatu struktur organisasi yang baik. Tanpa adanya pembagian kerja dan struktur organisasi, tidak akan dapat terjadi suatu kegiatan produksi. Sebaliknya, struktur organisasi yang tersusun dengan baik akan memudahkan koordinasi, integrasi, serta meningkatkan efektifitas dan efisiensi suatu perusahaan untuk mewujudkan visi dan misinya.

Hal tersebut tak terkecuali PT Galih Ayom Paramesti yang memerlukan struktur organisasi sehingga pegawai dari perusahaan tidak mengalami tumpang tindih dengan tugas dan kewajiban yang dimiliki. Berdasarkan penjelasan akan pentingnya struktur organisasi bagi suatu perusahaan yang sudah diterangkan, berikut ini merupakan struktur organisasi PT Galih Ayom Paramesti yang dapat dilihat pada Gambar IV.2.



Gambar IV.2 Struktur Organisasi PT Galih Ayom Paramesti
Sumber: PT Galih Ayom Paramesti (2018)

4.5. Tugas dan Wewenang

Dengan adanya struktur organisasi perusahaan diharapkan suatu perusahaan dapat berfungsi dengan baik. Di bawah ini merupakan uraian tugas dan wewenang secara garis besar dari bagian yang ada di PT Galih Ayom Paramesti:

1. Direktur

Direktur merupakan pimpinan tertinggi dalam suatu perusahaan. Seseorang yang menjabat sebagai direktur memiliki tugas untuk menuntun perusahaan agar sejalan dengan visi misi yang telah ditentukan. Seorang direktur juga bertugas menentukan peraturan dan kebijakan perusahaan serta bertanggung jawab atas keuntungan dan kerugian yang dihadapi oleh perusahaan.

2. Wakil Direktur

Merupakan wakil dari direktur yang bertugas membantu direktur dalam pelaksanaan tugasnya. Wakil direktur memiliki tugas yang secara langsung mengawasi kinerja yang membawahinya untuk kemudian dipertanggungjawabkan kepada direktur, serta memberikan arahan atau revisi jika diperlukan.

3. *Plant Division*

Sebuah bagian strategis yang mengemban tanggung jawab dan tanggung gugat atas manajemen aset berupa *equipment* dan memastikan ketersediaannya yang layak untuk dapat dipergunakan oleh bagian lain. Tugas dari seorang *plant manager* adalah memimpin, mengontrol, dan mengawasi keseluruhan dari operasi yang berjalan di pabrik dan juga bertanggung jawab untuk semua kegiatan di pabrik.

4. *FIA (Finance and Accounting)*

Sesuai dengan namanya, bagian FIA atau bagian keuangan memiliki tugas utama yaitu mengatur keuangan yang pada PT Galih Ayom Paramesti. Bagian FIA memeriksa keuangan yang masuk dan keluar serta melakukan pembelian mengenai kebutuhan di PT Galih Ayom Paramesti. Seorang bagian keuangan yang baik memiliki data keuangan lengkap perusahaan yang kemudian disusun rapi sehingga dapat memudahkan bagian lain apabila hendak menggunakannya.

5. *PPC (Production Planning and Control)*

Secara garis besar, Bagian PPC memiliki tugas dan wewenang yang berkaitan dengan membantu kelancaran Bagian Produksi. Beberapa tugas tersebut seperti melakukan pemesanan barang baku, membuat jadwal produksi, kemudian menjaga barang baku yang telah dipesan sebelumnya di dalam gudang. Tugas tersebut dilakukan oleh bagian yang berbeda meskipun termasuk dalam Bagian PPC. Bagian-bagian tersebut antara lain adalah: *PPC Head* atau *PPC Manager*, *PPC Warehouse* atau *Warehouse Leader*, *PPC Purchasing* atau *PPC Procurement* dan kemudian dibantu oleh *Operator*.

6. *QAC (Quality Assurance & Control)*

Bagian *Quality Assurance & Control* merupakan bagian yang bertugas menjaga dan mengontrol kualitas dari suatu barang baik barang baku yang telah dipesan oleh pemasok hingga barang jadi yang merupakan hasil produksi. Bagian QAC terdiri dari tiga bagian utama yaitu *QAC Manager*, *QAC Supervisor* dan *QAC Staff*.

7. *Production*

Bagian produksi merupakan bagian yang memiliki tugas utama dalam memproduksi produk. Semua produk yang diproduksi PT Galih Ayom Paramesti merupakan produk yang telah ditentukan spesifikasinya masing-masing seperti ketebalannya, kebulatannya serta kehalusan dari produk tersebut. Hal tersebut membuat masing-masing pegawai produksi harus memiliki keahlian dalam memproduksi barang yang tepat sesuai dengan spesifikasi yang diminta.

8. *Engineering*

Bagian *engineering* merupakan bagian yang ada dalam perusahaan manufaktur yang memiliki tugas untuk melakukan pengecekan terhadap kondisi mesin, memahami mesin dan memperbaiki mesin apabila kondisi mesin rusak. Selain itu bagian *engineering* juga memiliki tugas untuk melihat dan membuat gambar teknik baik itu ketika pelanggan melakukan pemesanan atau ketika produksi sedang dijalankan.

9. *Marketing*

Bagian Pemasaran merupakan ujung tombak dari suatu perusahaan, karena mereka harus memiliki koneksi dan cara yang banyak agar dapat memberikan pelanggan kepada perusahaan. Tanpa adanya suatu Bagian Pemasaran maka perusahaan tidak akan mendapat pemasukan sehingga segala hal dapat terhenti.

10. HRD & GA (*Human Resources and General Affairs*)

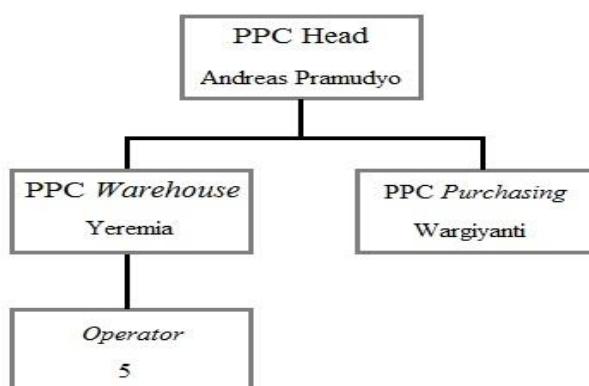
Merupakan bagian yang memiliki tugas dalam melakukan perekrutan karyawan baru. Selain itu tugas lain dari Bagian HRD & GA yaitu mengurus segala bentuk perizinan perusahaan, pengadaan dan distribusi ATK, mengurus segala bentuk kebutuhan operasional perusahaan serta mengurus karyawan *out-sourcing* seperti *office boy, security, dan driver*)

11. QMR (*Quality Management Representative*)

QMR adalah suatu bagian yang memiliki tugas dalam pengawasan mutu/kualitas dari suatu perusahaan. Tugas tersebut seperti halnya berkoordinasi dengan badan sertifikasi, *review* kebijakan mutu secara berkala, serta mengukur dan mengawasi kinerja proses perusahaan.

4.6. Struktur Organisasi PPC (*Production Planning and Control*)

Berikut struktur organisasi Bagian PPC pada PT Galih Ayom Paramesti (dapat dilihat pada Gambar IV.3).



Gambar IV.3 SO PPIC PT Galih Ayom Paramesti
Sumber: PT Galih Ayom Paramesti (2018)

4.7. Tugas dan Wewenang PPC

Tugas dan wewenang dari struktur organisasi Bagian PPC pada PT Galih Ayom Paramesti dapat dijelaskan sebagai berikut:

- **Bagian PPC Head/Manager**

Adapun tugas dan wewenang dari Bagian PPC *Head* adalah sebagai berikut:

- Membuat *production planning* harian, bulanan serta tahunan.
- Menerima, menyimpan dan merawat serta mengirim barang sesuai dengan jumlah yang telah ditentukan.
- Mengkoordinasikan kegiatan operasional dengan produksi.
- Mengontrol *delivery* dari pemasok dan ke pelanggan.
- Mencari pemasok baru sesuai dengan kebutuhan.
- Menentukan *loading* mesin dan perencanaan kerja.
- Menentukan waktu *delivery* sesuai dengan permintaan pelanggan.

- **Bagian PPC Warehouse**

Tugas dan wewenang dari Bagian PPC *Warehouse* adalah sebagai berikut:

- Memeriksa kelengkapan barang yang masuk dari pemasok (surat jalan, *check sheet, mill sheet, label*, dll).
- Memeriksa kelengkapan dokumen dari barang keluar.
- Membuat *invoice* ke pelanggan.

Selain itu PPC *Warehouse* memiliki tanggung jawab seperti:

- Mendaftarkan serta menjamin ketersediaan stok barang baku (*raw material*).
- Menjamin ketersediaan alat-alat penunjang proses produksi.
- Memastikan barang yang keluar/masuk diperiksa jumlah, kelengkapan serta kondisinya bersama dengan Bagian QAC (*Quality Assurance & Control*).

- **Bagian PPC Purchasing**

Beberapa tugas dan wewenang Bagian PPC *Purchasing* di antaranya:

- Membuat permintaan barang/alat/jasa ke pemasok sesuai dengan permintaan bagian terkait.
- Melaksanakan proses pembelian/membuat PO.
- Membuat data/grafik performa setiap pemasok.

Selain itu PPC *Purchasing* memiliki tanggung jawab, di antaranya adalah:

- Memenuhi/membeli kebutuhan dalam perencanaan produksi.
- Memenuhi/membeli alat ukur dan penunjang dalam proses inspeksi.
- Memenuhi/membeli kebutuhan bagian lain dalam rangka menunjang kegiatan manufaktur.

4.8. Dokumen Terkait Penjadwalan Produksi

Terdapat dokumen-dokumen yang terkait dalam proses penjadwalan produksi pada PT Galih Ayom Paramesti antara lain adalah sebagai berikut:

a. *Purchase Order*

Purchase Order merupakan dokumen yang diberikan oleh pelanggan kepada Bagian Pemasaran (dapat dilihat pada Gambar IV.4). Berperan sebagai surat pembelian atau pemesanan yang dilakukan pelanggan kepada PT Galih Ayom Paramesti. *Purchase Order* diterima oleh Bagian Pemasaran untuk kemudian digunakan sebagai acuan pada *meeting* bulanan untuk dibuatkan *Production Planning* oleh Bagian PPC.

		PURCHASE ORDER						Nomor Order RA1234	Date 28/06/2018	Buyer PUR004	Page 1		
		ARRIVAL		QUANTITY	U/M	U/C	UNIT PRICE						
LINE NO	PART NO	DESCRIPTION	A/T	DATE	LOCATION								
1	13066-001	SHAFT-KICK		12:00	8/08/2018	KSPINV	20	pcs	IDR	20,000.00	400,000.00		
2	13066-006	SHAFT-KICK		12:00	9/08/2018	KSPINV	40	pcs	IDR	25,000.00	1,000,000.00		
3	42036-006	SLEEVE		12:00	8/08/2018	KSPINV	100	pcs	IDR	15,000.00	1,500,000.00		
4	42036-124	SLEEVE 10X16X19		12:00	8/08/2018	KSPINV	40	pcs	IDR	1,500.00	60,000.00		
5	42036-135	SLEEVE		12:00	8/08/2018	KSPINV	100	pcs	IDR	8,000.00	800,000.00		
6	92043-010	PIN 8.2X10X4		12:00	8/08/2018	KSPINV	120	pcs	IDR	450.00	54,000.00		
7	92143-143	COLLAR 9.5X12.7X20		12:00	8/08/2018	KSPINV	10	pcs	IDR	550.00	5,500.00		
8	92152-073	COLLAR		12:00	8/08/2018	KSPINV	20	pcs	IDR	5,000.00	100,000.00		
								TOTAL		3,919,500.00			
								PPN		391,950.00			
								GRAND TOTAL		4,311,450.00			

Gambar IV.4 Contoh PO PT Galih Ayom Paramesti

Sumber: PT Galih Ayom Paramesti (2018)

Keterangan-keterangan pada Gambar IV.4 dijelaskan sebagai berikut:

- *Logo*
Merupakan logo dari pelanggan yang memesan dan membuat PO.
- *Nomor Order*
Berisi nomor order atau nomor PO yang dibuat oleh pelanggan.
- *Date*
Berisi tanggal dikirimkannya PO tersebut kepada pemasok.
- *Buyer*
Berisi kode pembeli daripada produk yang nantinya sudah dibuat.
- *Company Address*
Berisi alamat dari pabrik/kantor tempat PO tersebut dituju.
- *Reference*
Berisi nama referensi dokumen tersebut dikirimkan. Pada kolom referensi terdapat kode dari nama orang yang dituju.
- *Terms of Payment*
Berisi ketentuan pembayaran yang dilakukan kepada pemasok. Dalam contoh PO di atas tertulis bahwa pembayaran dilakukan dalam 30 hari kerja.
- *Line No.*
Berisi nomor produk yang dipesan.
- *Parts No.*
Berisi jenis barang yang dipesan dari suatu produk (yang dibedakan menggunakan nomor).
- *Description*
Berisi nama barang yang dipesan pada suatu PO.
- *A/T (Arrival Time)*
Berisi waktu kedatangan produk tersebut ke pelanggan.
- *Arrival Date*
Berisi tanggal kedatangan produk tersebut dari perusahaan menuju ke pelanggan.

- *Arrival Location*
Berisi lokasi kedatangan produk tersebut yang ditentukan oleh pelanggan.
 - *Quantity*
Berisi jumlah barang yang dipesan pada masing-masing produk.
 - U/M
Berisi jenis satuan jumlah dari barang yang dipesan.
 - U/C
Berisi jenis mata uang (*currency*) yang digunakan sebagai alat pembayaran.
 - *Unit Price*
Berisi harga dari barang tiap satuannya.
 - *Total Price*
Berisi harga total yang didapatkan dari barang tiap satuannya kemudian dikalikan dari jumlah kuantitas jenis barang yang dipesan.
 - Total
Berisi harga total yang didapatkan setelah menjumlahkan semua harga pada *total price*.
 - PPN (Pajak Pertambahan Nilai)
Berisi jumlah pajak sejumlah 10% dari harga total.
 - *Grand Total*
Berisi harga total berupa jumlah total yang ditambahkan dari harga tiap-tiap barang yang kemudian dijumlahkan oleh PPN yang sudah dihitung.
- b. Dokumen Jadwal Pengiriman (*Schedule Delivery*)
Merupakan dokumen yang berisi data mengenai tenggat waktu pengiriman produk yang dipesan pelanggan. *Schedule Delivery* (dapat dilihat pada Gambar IV.5) diberikan oleh pelanggan kepada PT Galih Ayom Paramesti ketika sudah melakukan proses produksi massal (*mass production*). Dengan diketahuinya jumlah yang harus dikirimkan kepada konsumen, Bagian PPC dapat memperhitungkan dan menjadwalkan proses produksi dan seberapa banyak jumlah produk yang harus diproduksi pada jangka waktu tertentu.

SCHEDULE DELIVERY PT GALIH AYOM PARAMESTI												
3 FEBRUARI												
Vendor Name	Start on Delivery Due Date	Item No.	Item Description	Buyer	PO No.	Line No.	Order Qty	Recv Qty	Date Delivery	Plan Qty	Un-Delivery	Reason
GAP	3/02/2016	12020-0015	SHAFT-ROCKER	EDK	PA0001	18	2.600	0				
GAP	3/02/2016	13066-0014	SHAFT-KICK	EDK	PA0001	19	1.020	0				
GAP	3/02/2016	13116-0706	ROD-PUSH	EDK	PA0001	20	1.300	0				
GAP	3/02/2016	42036-0068	SLEEVE	EDK	PA0001	21	1.200	0				
GAP	3/02/2016	42036-1248	SLEEVE	EDK	PA0001	22	2.000	0				
GAP	3/02/2016	42036-1351	SLEEVE	EDK	PA0001	23	1.000	0				
GAP	3/02/2016	92026-1605 Z	SPACER	EDK	PA0001	24	100	0				
GAP	3/02/2016	92043-0099	PIN	EDK	PA0001	25	9.200	0				
GAP	3/02/2016	92043-0100	PIN	EDK	PA0001	26	5.300	0				
GAP	3/02/2016	92043-0744	PIN	EDK	PA0001	27	2.500	0				
GAP	3/02/2016	92043-080	PIN	EDK	PA0001	28	2.500	0				
GAP	3/02/2016	92043-0834	PIN	EDK	PA0001	29	1.200	0				
GAP	3/02/2016	92043-1294 1A	PIN	EDK	PA0001	30	300	0				
GAP	3/02/2016	92152-0736	COLLAR	EDK	PA0001	31	1.500	0				
GAP	3/02/2016	92152-1139	COLLAR	EDK	PA0001	32	1.000	0				

Gambar IV.5 Contoh *Schedule Delivery*

Sumber: PT Galih Ayom Paramesti (2018)

Keterangan *field-field* pada Gambar IV.5:

- *Vendor Name*:
Berisi nama perusahaan pengirim atau pembuat produk tersebut
- *Start on Delivery Due Date*:
Berisi tanggal pengiriman pesanan kepada perusahaan pembeli (pembuat *schedule delivery*).
- *Item No.*:
Berisi nomor atau jenis dari suatu produk.
- *Item Description*:
Berisi nama produk yang dipesan oleh pembeli.
- *Buyer*:
Berisi nama pembeli produk pesanan.
- *PO No.*:
Berisi nomor *purchase order* yang dikirimkan oleh pelanggan.
- *Order Qty*:
Berisi jumlah produk yang dipesan oleh pelanggan ke *vendor*.
- *Received Qty*:
Berisi jumlah produk yang sudah diterima oleh pembeli.

- *Date Delivery*

Berisi tanggal pengiriman produk yang sudah dikirimkan.

- *Un-delivery*

Berisi jumlah produk yang tidak dikirim.

- *Reason*

Berisi alasan mengapa produk tidak dikirimkan oleh *vendor*.

c. Dokumen Perencanaan/Penjadwalan Produksi (*Planning Production*)

Merupakan dokumen yang berisi *detail* jumlah yang harus diproduksi dalam jangka waktu tertentu (dapat dilihat pada Gambar IV.6). Selain itu, dalam dokumen tersebut terdapat keterangan proses produksi yang dilakukan, jumlah produksi yang telah dilakukan, tanggal jumlah produksi tidak sesuai kuota produksi serta bagaimana cara memperbaiki hal tersebut agar semuanya dapat diproduksi sesuai dengan tenggat waktu yang telah ditentukan.

PERIOD		MEI 2017				PLAN PRODUKSI PT. KMI PERIODE MEI 2017																		PPIC			PPIC			PRODUKSI						
CUSTOMER		PT. KMI																																		
PART NAME		SHAFT KICK																																		
PART NUMBER		13066-0014																						EKO P			JANG S			S MARNO						
NO.	PROCESS	MACHINE	S/T	KPS	DATE	MEI 2017																														
						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	1		CUTTING OFF		FONG HO		28.8		977		PLAN																									
					ACTUAL																															
					PLAN																															
					ACCUM																															
					ACTUAL																															
	2		CUTTER DRILL		BL SD 25		30.5		921		PLAN																									
					ACCUM																															
					ACTUAL																															
					BALANCE																															
					PLAN																															
	3		SERATION		HOBING MC (2 Mesin)		184		153		ACCUM																									
					ACTUAL																															
					ACCUM																															
					BALANCE																															
					DELIVERY SCHEDULE TOTAL PCS																															

Gambar IV.6 Contoh Format Jadwal Produksi

Sumber: PT Galih Ayom Paramesti (2018)

Keterangan *field-field* pada Gambar IV.6:

- *Period*: Berisi waktu atau periode penggerjaan penjadwalan produksi.
- Pelanggan: Berisi nama perusahaan yang memesan produk.
- *Part Name*: Berisi nama produk yang dibuatkan jadwal produksinya.
- *Part Number*: Berisi nomor produk yang dibuatkan jadwal produksinya.
- No: Merupakan penomoran dalam penjadwalan produksi.
- Proses: Berisi nama-nama proses yang dijalankan untuk membuat suatu produk yang dibuat penjadwalan produksinya.
- *Machine*: Berisi nama mesin yang digunakan untuk menjalankan proses.
- S/T: Berisi waktu yang digunakan dalam memproduksi suatu produk.
- KPS: Berisi jumlah yang dapat dihasilkan oleh mesin dalam 1 (satu) *shift* kerja.
- *Date*: Berisi tanggal penggerjaan pada penjadwalan produksi.
- *Plan*: Berisi perencanaan produksi pada hari tertentu.
- *Actual*: Berisi hasil produksi yang sebenarnya pada hari tertentu.
- *Accum*: Berisi jumlah yang sudah dibuat (*optional*)
- *Balance*: Berisi sisa jumlah (*optional*)

4.9. Sistem Informasi Penjadwalan Produksi yang Sedang Berjalan

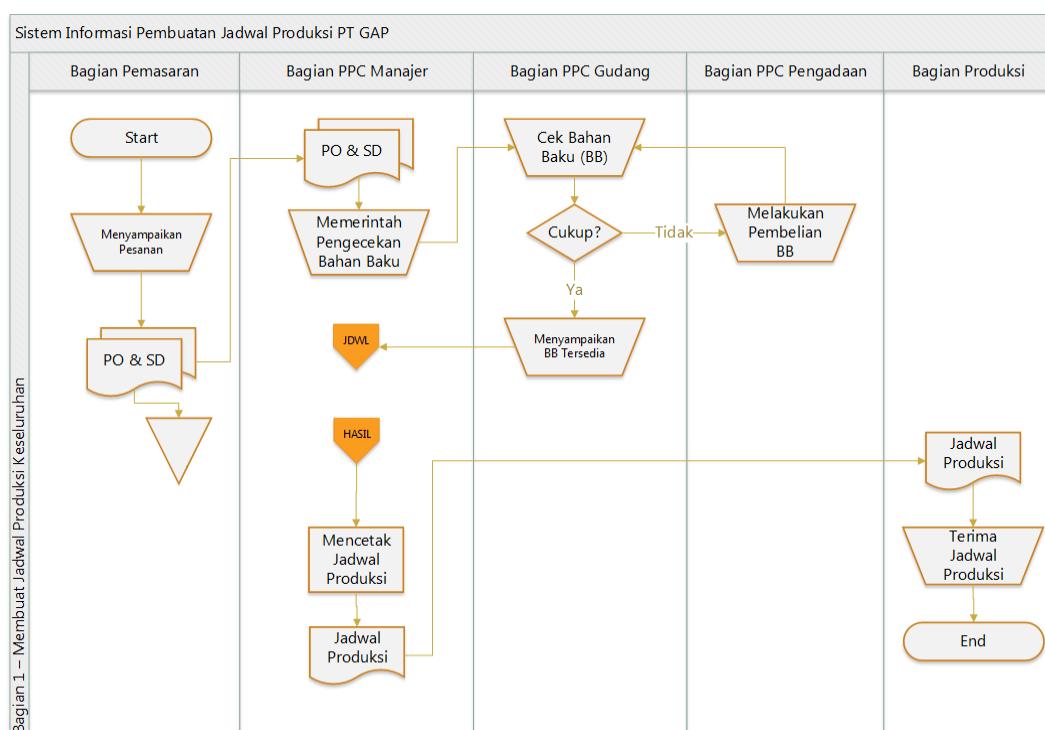
PT Galih Ayom Paramesti melibatkan beberapa bagian dalam sistem informasi penjadwalan produksinya. Bagian tersebut antara lain adalah Bagian Pemasaran, Bagian Produksi, Bagian Gudang, Bagian Pengadaan dan Bagian PPC Manajer. Sistem penjadwalan produksi (dapat dilihat pada Gambar IV.7 dan Gambar IV.8) di awali ketika Bagian Pemasaran memberikan pesanan kepada Bagian PPC Manajer.

Pesanan tersebut diberikan beserta dengan dokumen PO (*Purchase Order*) dan SD (*Schedule Delivery*) yang diberikan oleh pelanggan. Dokumen tersebut nantinya akan digunakan sebagai acuan ketika hendak melakukan penyusunan jadwal produksi. Pada proses penjadwalan produksi, Bagian PPC Manajer memulai dengan menentukan dan memperhitungkan jenis dan jumlah bahan baku yang dibutuhkan untuk menunjang produksi dari pesanan pelanggan.

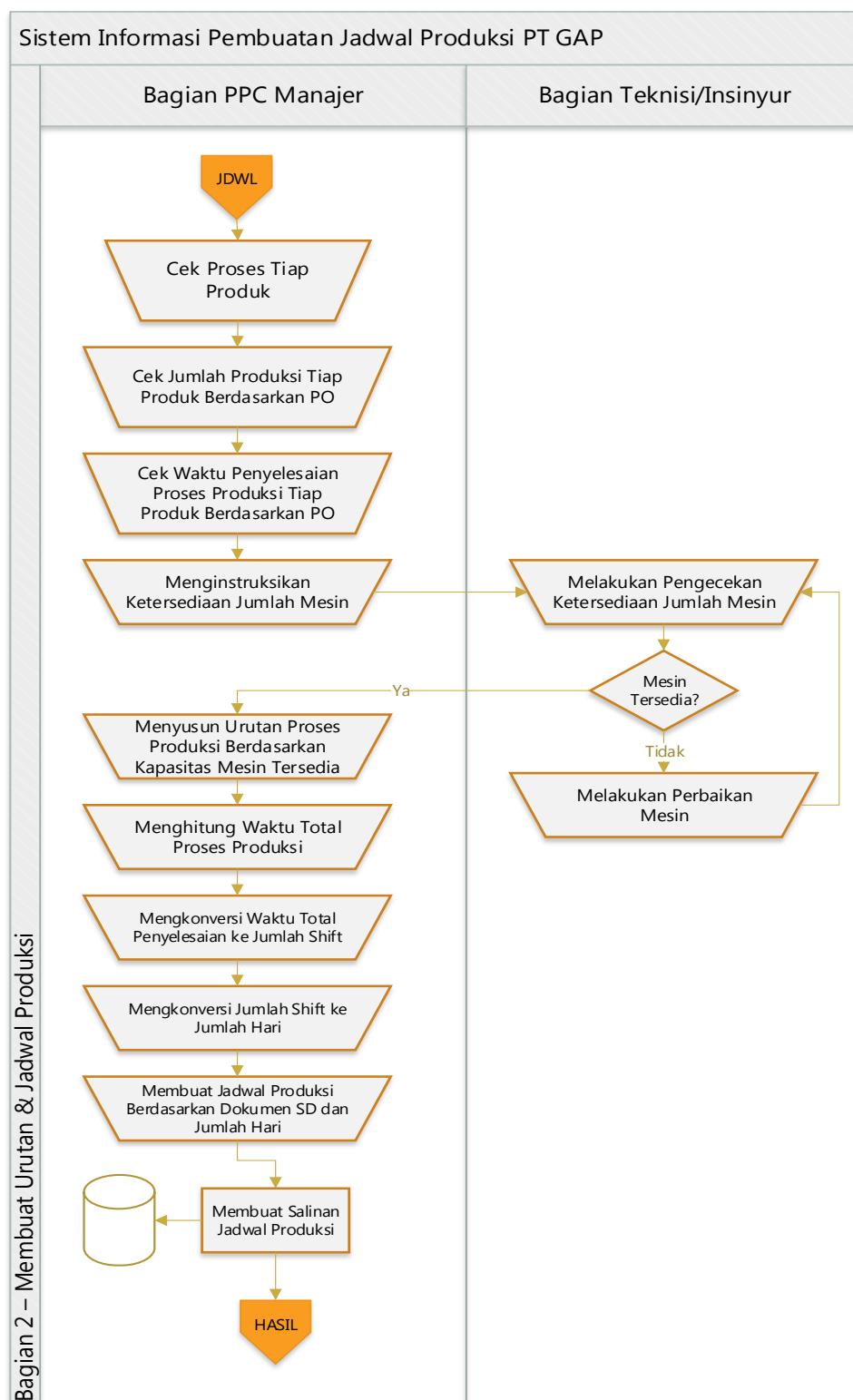
Bagian PPC Manajer kemudian akan menginstruksikan Bagian Gudang untuk melakukan proses selanjutnya yaitu pengecekan akan ketersediaan barang di gudang. Pada saat melakukan pengecekan akan dicari tahu apabila bahan baku untuk menjalankan proses produksi mencukupi atau tidak. Bagian Pengadaan kemudian akan melakukan pembelian apabila bahan baku tidak mencukupi.

Namun, apabila bahan baku sudah mencukupi maka Bagian PPC Manajer dapat melanjutkan proses penjadwalan produksi. Proses Penjadwalan produksi dilakukan oleh Bagian PPC Manajer dengan melakukan perhitungan jumlah produk yang dapat dihasilkan. Perhitungan tersebut dilakukan dengan mempertimbangkan waktu dalam satu sesi kerja (*shift*) yang kemudian dihubungkan dengan waktu yang dibutuhkan oleh mesin untuk menghasilkan produk.

Bagian PPC Manajer kemudian akan mengkonversi waktu tersebut untuk mencari tahu waktu yang dibutuhkan apabila hendak menyelesaikan pesanan pelanggan sebagai akhir dari proses penjadwalan produksi. Hasil akhir dokumen jadwal produksi tersebut akan diberikan kepada Bagian Produksi untuk kemudian dijadikan sebagai acuan dalam proses produksi.



Gambar IV.7 Bagan Alir Penjadwalan Produksi Keseluruhan
Sumber: (Hasil Analisis, 2019)



Gambar IV.8 Bagan Alir Pembuatan Urutan dan Jadwal Produksi
Sumber: Hasil Analisis (2019)

BAB V

HASIL ANALISIS DAN PEMBAHASAN

5.1. Analisis Kebutuhan Sistem

Terdapat beberapa kebutuhan yang diperlukan dalam pembuatan sistem informasi penjadwalan produksi. Berikut ini merupakan kebutuhan-kebutuhan pada sistem usulan yang diperoleh setelah dilakukannya analisis pada sistem sebelumnya yang dapat dilihat pada Tabel V.1, Tabel V.2 dan Tabel V.3.

Tabel V.1 Analisis Kebutuhan Sistem Usulan

No	Permasalahan	Kebutuhan <i>User</i>	Solusi
1	Proses penjadwalan produksi hanya menggunakan Microsoft Excel.	Sistem yang dapat mengelola data jadwal produksi serta menampilkan hasilnya.	Merancang dan membangun sistem penjadwalan produksi yang terintegrasi.
2	Informasi mesin yang digunakan disimpan di kertas serta data Microsoft Excel.	Sistem yang dapat mengelola informasi mesin ke dalam basis data.	Merancang dan membangun sistem yang dapat menampilkan dan menyimpan informasi mesin yang digunakan.
3	Informasi produk yang sedang atau telah diproduksi sebelumnya disimpan di kertas serta data Microsoft Excel.	Sistem yang dapat mengelola informasi produk dan detail produk ke dalam basis data.	Merancang dan membangun sistem yang dapat menampilkan dan menyimpan informasi detail produk yang dibuat.
4	Proses penjadwalan produksi dilakukan berdasarkan preferensi pegawai hal tersebut mengakibatkan efektivitas perhitungan kurang konsisten.	Sistem yang dapat menentukan perhitungan jadwal produksi berdasarkan metode Algoritma <i>Active Schedule</i> .	Merancang dan membangun sistem penjadwalan produksi dengan mengimplementasikan metode Algoritma <i>Active Schedule</i> .

Tabel V.1 Analisis Kebutuhan Sistem Usulan (Lanjutan)

No	Permasalahan	Kebutuhan <i>User</i>	Solusi
5	Belum terdapat sistem terkomputerisasi untuk mengelola data yang dibedakan berdasarkan jabatan	Sistem yang dapat mengelola fitur aplikasi yang ada berdasarkan hak akses.	Merancang dan membangun sistem yang memiliki fungsi khusus pada masing-masing hak akses pengguna.

Sumber: Hasil Analisis (2019)

Tabel V.2 Analisis Kebutuhan Sistem *Functional Requirement*

No	Kebutuhan Sistem (<i>Functional Requirement</i>)
1	Sistem dapat menyimpan penjadwalan produksi terintegrasi dengan bagian PPC Manajer.
2	Sistem menampilkan informasi mesin terintegrasi dengan bagian Teknisi/Insinyur.
3	Sistem menampilkan informasi produk terintegrasi dengan bagian PPC Manajer.
4	Sistem menampilkan informasi detail produk sesuai dengan data produk terintegrasi dengan bagian PPC Manajer
5	Sistem menampilkan hasil perhitungan yaitu berupa urutan-urutan penjadwalan produksi.

Sumber: Hasil Analisis (2019)

Tabel V.3 Analisis Kebutuhan Sistem *Non Functional Requirement*

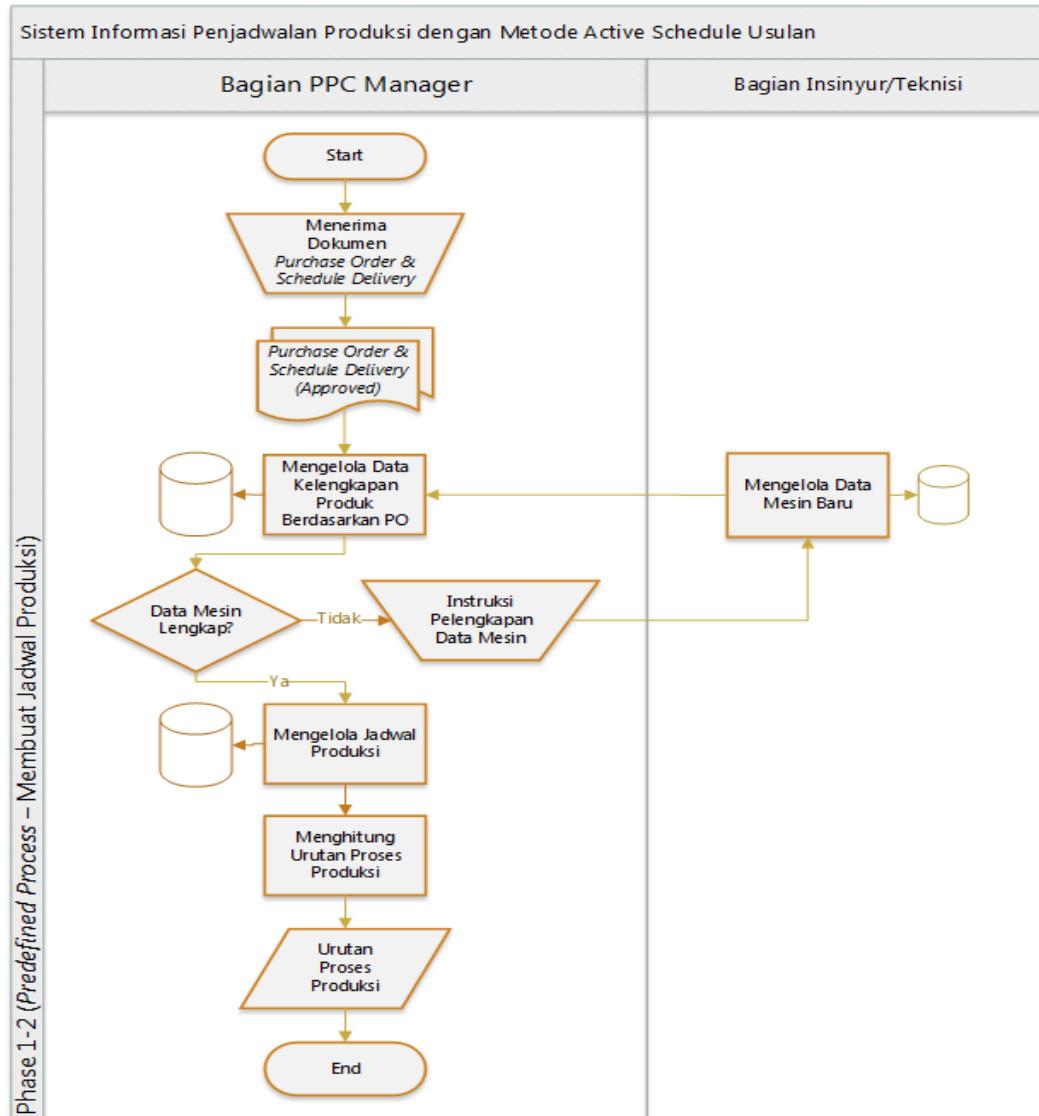
No	Kebutuhan Sistem (<i>Non Functional Requirement</i>)
1	Aplikasi harus dijalankan menggunakan koneksi internet.
2	Aplikasi beroperasi pada PC (<i>Personal Computer</i>).
3	Aplikasi dapat dijalankan oleh Bagian PPC Manajer, Bagian Insinyur/Teknisi, serta Admin selaku pengguna yang mengatur aplikasi secara keseluruhan.
4	Penjadwalan Produksi hanya dapat dibuat oleh Bagian PPC Manajer serta Admin.
5	Informasi Produk dan Detail Produk hanya dapat dibuat oleh Bagian PPC Manajer serta Admin.
6	Informasi Mesin hanya dapat dibuat oleh Bagian Teknisi serta Admin.
7	Informasi pengguna yang dapat menjalankan dan menggunakan fitur aplikasi hanya dapat dibuat dan diakses oleh Admin.

Sumber: Hasil Analisis (2019)

5.2. Sistem Informasi Penjadwalan Produksi Usulan

Sistem informasi penjadwalan produksi usulan dengan implementasi Algoritma *Active Schedule* tersusun atas beberapa tahapan. Berikut ini merupakan tahapan-tahapan serta gambaran yang dibuat menggunakan diagram alir yang dapat dilihat pada Gambar V.1:

1. Secara garis besar, sistem informasi penjadwalan produksi usulan hanya terpusat pada implementasi metode *active schedule* untuk menghasilkan urutan proses produksi ketika melakukan proses penjadwalan produksi. Alur proses penerimaan pesanan yang dilakukan sebelum dilakukannya proses implementasi dan alur pembuatan jadwal produksi yang dilakukan setelah dilakukannya proses implementasi sama seperti alur yang sudah berjalan. Tahapan proses bisnis usulan tersebut terdiri dari:
 - a. Bagian PPC Manajer menerima dokumen *Purchase Order (PO)* dan *Schedule Delivery (SD)* yang telah disetujui oleh Bagian Pemasaran.
 - b. Bagian PPC Manajer akan menambahkan data produk secara lengkap berdasarkan dokumen *Purchase Order* yang kemudian akan tersimpan ke dalam basis data.
 - c. Pada saat melakukan tambah data kelengkapan produk, apabila mesin yang hendak ditambahkan terkait data produk sebelumnya belum tersedia maka Bagian PPC Manajer akan memberikan instruksi kepada Bagian Teknisi untuk menambahkan data mesin baru ke dalam basis data mesin.
 - d. Bagian Teknisi akan menambahkan data mesin baru, kemudian tersimpan ke dalam basis data.
 - e. Jika semua data telah tersedia, Bagian PPC Manajer akan menambahkan data jadwal, kemudian tersimpan ke dalam basis data.
 - f. Bagian PPC Manajer kemudian akan melakukan hitung urutan proses produksi berdasarkan jadwal yang pernah ditambahkan sebelumnya.
 - g. Hasil akhir setelah hitung urutan proses produksi adalah urutan proses produksi yang kemudian dapat digunakan sebagai acuan dalam pembuatan jadwal produksi.



Gambar V.1 Flowmap Sistem Informasi Penjadwalan Produksi Usulan
Sumber: Hasil Analisis (2019)

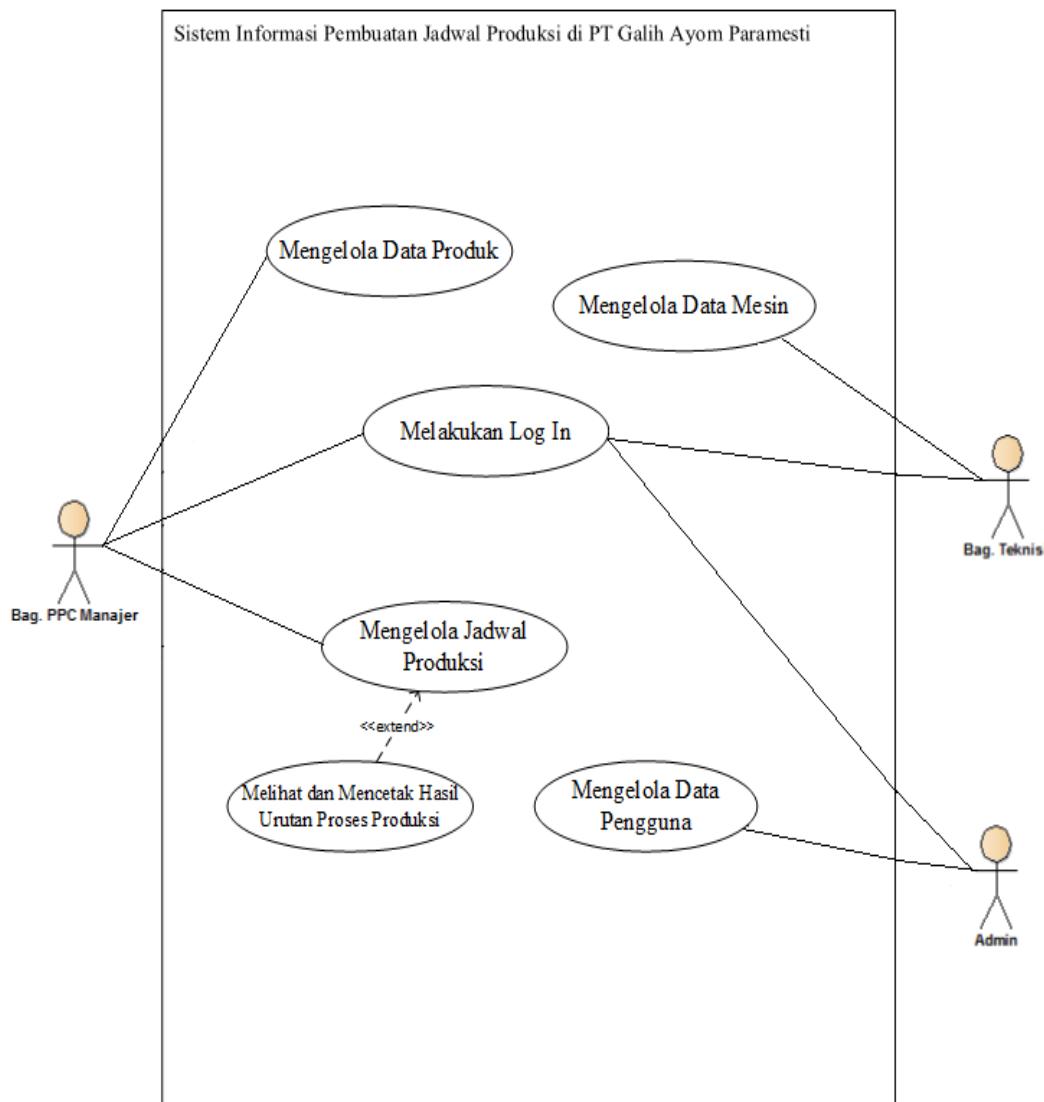
5.3. Analisis Sistem Usulan

Sistem penjadwalan produksi dibangun dengan menggunakan UML (*Unified Modeling Language*) sebagai model penyusunnya. Pemodelan sistem UML yang digunakan pada analisis sistem usulan berfungsi untuk menjelaskan gambaran dari sistem yang hendak dibangun. Untuk menjelaskan gambaran sistem tersebut digunakan model UML berupa *use case diagram*, *activity diagram*, dan *deployment diagram*. Penggunaan *entity relationship diagram* juga digunakan untuk memperjelas relasi pada tiap-tiap entitas. Di bawah ini merupakan penerapan dari jenis pemodelan sistem yang digunakan.

5.3.1. Use Case Diagram

Use case diagram atau yang sering disebut juga dengan *behaviour diagram* menjelaskan kumpulan dari kegiatan-kegiatan yang dilakukan oleh masing-masing aktor pada sebuah sistem. Pada sistem penjadwalan produksi terdapat beberapa aktor yang terlibat seperti Bagian PPC Manajer, Bagian Insinyur atau Teknisi serta Admin yang memiliki kendali dalam mengatur data pengguna di aplikasi *web*.

Berikut ini merupakan Gambar V.2 yang menampilkan *use case diagram* dari sistem yang diusulkan.



Gambar V.2 *Use Case Diagram* Sistem Usulan
Sumber: Hasil Analisis (2019)

Penjelasan dari *use case diagram* yang digunakan sebagai media pemodelan pada sistem informasi penjadwalan produksi adalah sebagai berikut:

1. Definisi Aktor

Pengertian dari aktor yang ada pada *use case diagram* sistem informasi penjadwalan produksi dapat dilihat pada Tabel V.4.

Tabel V.4 Definisi Aktor *Use Case Diagram* Usulan

No	Aktor	Definisi
1	Admin	Aktor yang mengelola data keseluruhan yang terdapat pada aplikasi seperti data pengguna, data produk serta data mesin. Aktor admin dapat digunakan oleh Direktur atau Wakil Direktur.
2	Bag. PPC Manajer	Aktor yang mengelola data produk serta melakukan penjadwalan produksi
3	Bag. Teknisi / <i>Engineer</i>	Aktor yang mengelola data mesin yang digunakan pada tiap-tiap pembuatan produk.

Sumber: Hasil Analisis (2019)

2. Definisi *Use Case*

Pengertian dari masing-masing *use case* yang terdapat pada sistem informasi penjadwalan produksi dapat dilihat pada Tabel V.5.

Tabel V.5 Definisi *Use Case* Usulan

No	Use Case	Definisi
1	Melakukan <i>log in</i>	Proses melakukan <i>log in</i> pada aplikasi.
2	Mengelola data produk	Proses mengelola data produk serta detailnya oleh Bagian PPC Manajer atau Admin.
3	Mengelola data pengguna	Proses mengelola data pengguna oleh Admin.
4	Mengelola data mesin	Proses mengelola data mesin oleh Bagian Teknisi atau Admin.
5	Mengelola jadwal produksi	Proses membuat dan menghapus jadwal produksi oleh Bagian PPC Manajer atau Admin.
6	Melihat hasil urutan proses produksi	Proses melihat hasil urutan proses produksi oleh Bagian PPC Manajer atau Admin.

Sumber: Hasil Analisis (2019)

3. Skenario *Use Case*

Alur jalannya masing-masing *use case* pada sistem informasi penjadwalan produksi dapat dilihat berdasarkan penjelasan dibawah ini:

- *Use Case* Melakukan *Log in*

Berikut merupakan scenario *use case* melakukan *log in* yang terdapat pada Tabel V.6.

Tabel V.6. Skenario *Use Case* Melakukan *Log in*

Nama Use Case	Melakukan Log in
<i>Primary Actor</i>	Admin, Bagian PPC Manajer, Bagian Teknisi
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan proses <i>log in</i> yang dibedakan berdasarkan hak akses berupa <i>level/jabatan</i> masing-masing pengguna.
<i>Relationship</i>	<i>Association:</i> Admin, Bagian PPC Manajer, Bagian Teknisi.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi 2. Sistem menampilkan halaman <i>log in</i> 3. Sistem menampilkan <i>form log in</i> 4. Pengguna mengisi <i>username</i> dan <i>password</i> pada <i>form login</i> 5. Sistem melakukan validasi <i>log in</i> 6. Jika <i>username</i> dan <i>password valid</i>, maka akan muncul halaman utama (<i>dashboard</i>) 7. Jika <i>username</i> dan <i>password</i> tidak <i>valid</i> maka akan menampilkan halaman <i>log in</i> beserta penjelasan <i>error</i>.

Sumber: Hasil Analisis (2019)

- *Use Case* Mengelola Data Pengguna

Berikut merupakan scenario *use case* mengelola data pengguna yang terdapat pada Tabel V.7.

Tabel V.7. Skenario *Use Case* Mengelola Data Pengguna

Nama Use Case	Mengelola Data Pengguna
<i>Primary Actor</i>	Admin
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan proses mengelola data pengguna yang terdiri dari menambah, menghapus, mengubah data yang dilakukan oleh Admin.
<i>Relationship</i>	<i>Association:</i> Admin.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Admin memilih <i>sidemenu User</i> 2. Sistem menampilkan Data Pengguna 3. Admin dapat melakukan ketuk tombol <i>create</i>, <i>edit</i> atau <i>delete</i>. 4. Jika pengguna memilih tombol <i>delete</i> maka sistem akan melakukan hapus data sesuai dengan pilihan pengguna. 5. Jika pengguna memilih tombol <i>create</i> maka sistem akan menampilkan <i>form add</i> (tambah data) 6. Pengguna dapat memasukan data kemudian mengetuk tombol <i>save</i>. Sistem akan menyimpan ke dalam database 7. Jika pengguna memilih tombol <i>edit</i> maka sistem akan menampilkan <i>form edit</i> 8. Pengguna dapat mengubah data yang diinginkan kemudian mengetuk tombol <i>save</i>. Sistem akan menyimpan perubahan ke dalam database.

Sumber: Hasil Analisis (2019)

- *Use Case* Mengelola Data Mesin

Berikut merupakan scenario *use case* untuk mengelola data mesin yang terdapat pada Tabel V.8.

Tabel V.8. Skenario *Use Case* Mengelola Data Mesin

Nama Use Case	Mengelola Data Mesin
<i>Primary Actor</i>	Bagian <i>Engineer</i> (Teknisi), Admin
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan proses mengelola data mesin yang terdiri dari menambah, menghapus, mengubah data yang dilakukan oleh Pengguna.
<i>Relationship</i>	<i>Association: Engineer, Admin</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Pengguna memilih <i>treeview menu Machining</i> kemudian pilih <i>sidemenu Machines</i>. 2. Sistem menampilkan Data Mesin 3. Pengguna dapat melakukan ketuk tombol <i>create, update</i> atau <i>delete</i>. 4. Jika pengguna memilih tombol <i>delete</i> maka sistem akan melakukan hapus data sesuai dengan pilihan pengguna. 5. Jika pengguna memilih tombol <i>create</i> maka sistem akan menampilkan <i>form add</i> (tambah data) 6. Pengguna dapat memasukan data kemudian mengetuk tombol <i>save</i>. Sistem akan meyimpan ke dalam database 7. Jika pengguna memilih tombol <i>edit</i> maka sistem akan menampilkan <i>form edit</i> 8. Pengguna dapat mengubah data yang diinginkan kemudian mengetuk tombol <i>save</i>. Sistem akan menyimpan perubahan ke dalam database.

Sumber: Hasil Analisis (2019)

- *Use Case* Mengelola Data Produk

Berikut merupakan scenario *use case* untuk mengelola data produk yang terdapat pada Tabel V.9.

Tabel V.9. Skenario *Use Case* Mengelola Data Produk

Nama Use Case	Mengelola Data Produk
<i>Primary Actor</i>	Bagian PPC Manajer, Admin
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan proses mengelola data produk dan detail produk yang terdiri dari menambah, menghapus, mengubah data yang dilakukan oleh Pengguna.
<i>Relationship</i>	<i>Association:</i> Manajer, Admin.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Pengguna memilih <i>treeview menu Product Managing</i> kemudian pilih <i>menu Products</i> atau <i>Product Detail</i>. 2. Sistem menampilkan Data Produk atau Data Detail Produk 3. Pengguna dapat melakukan ketuk tombol <i>create</i>, <i>edit</i> atau <i>delete</i> apabila memilih Data Produk. 4. Jika pengguna memilih tombol <i>delete</i> maka sistem akan melakukan hapus data sesuai dengan pilihan pengguna. 5. Jika pengguna memilih tombol <i>create</i> maka sistem akan menampilkan <i>form add</i> (tambah data) 6. Pengguna dapat memasukan data kemudian mengetuk tombol <i>save</i>. Sistem akan menyimpan ke dalam database. 7. Jika pengguna memilih tombol <i>edit</i> maka sistem akan menampilkan <i>form edit</i> berdasarkan baris yang dipilih pengguna. 8. Pengguna dapat mengubah data yang diinginkan kemudian mengetuk tombol <i>save</i>. Sistem akan menyimpan perubahan ke dalam database.

Sumber: Hasil Analisis (2019)

- *Use Case* Mengelola Jadwal Produksi

Berikut merupakan scenario *use case* untuk Mengelola Jadwal Produksi yang terdapat pada Tabel V.10.

Tabel V.10. Skenario *Use Case* Mengelola Jadwal Produksi

Nama Use Case	Mengelola Jadwal Produksi
<i>Primary Actor</i>	Bagian PPC Manajer, Admin
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan proses membuat atau menghapus jadwal produksi oleh Pengguna.
<i>Relationship</i>	<i>Association:</i> Manajer, Admin.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Pengguna memilih sidemenu <i>Schedulling</i> kemudian pilih <i>menu Active Schedule</i>. 2. Sistem menampilkan data jadwal. 3. Ketuk <i>delete</i> untuk menghapus data berdasarkan baris. 4. Ketuk <i>add</i> untuk menambah jadwal baru. 5. Sistem menampilkan <i>form</i> tambah jadwal jika mengetuk <i>add</i>. 6. Pengguna dapat mengisi <i>form</i> yang berisi tanggal, produk dan jumlah untuk kemudian dimasukkan ke dalam database.

Sumber: Hasil Analisis (2019)

- *Use Case* Melihat dan Mencetak Hasil Urutan Proses Produksi

Berikut merupakan scenario *use case* untuk melihat hasil perhitungan jadwal produksi yang terdapat pada Tabel V.11.

Tabel V.11. Skenario *Use Case* Melihat Hasil Urutan Proses Produksi

Nama Use Case	Melihat Hasil Urutan Proses Produksi
<i>Primary Actor</i>	Bagian PPC Manajer, Admin
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan proses melihat dan mencetak hasil urutan proses produksi menggunakan metode <i>active schedule</i> berdasarkan jadwal produksi oleh Pengguna.

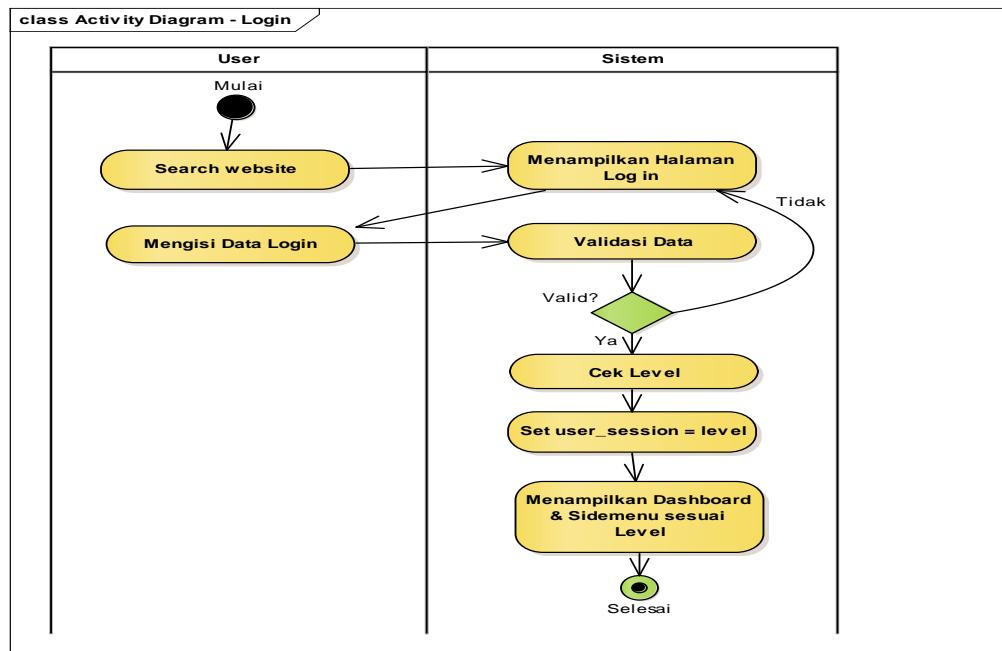
<i>Relationship</i>	<i>Association:</i> Manajer, Admin.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Pengguna memilih sidemenu <i>Schedulling</i> kemudian pilih menu <i>Active Schedule</i> 2. Sistem menampilkan data Jadwal 3. Ketuk <i>process</i> untuk memproses jadwal 4. Isi tanggal jadwal yang ingin di proses 5. Apabila data tersedia sistem akan menampilkan jadwal pada menu <i>Result Schedule</i>, namun apabila tidak ada maka muncul pesan kesalahan 6. Ketuk cetak untuk mencetak hasil jadwal

Sumber: Hasil Analisis (2019)

5.3.2. *Activity Diagram*

Activity diagram menjelaskan aliran kerja atau proses bisnis yang dikerjakan oleh sistem atau perangkat lunak. Meskipun begitu, *activity diagram* tetap menggunakan *use case* sebagai acuannya. Berikut ini merupakan *activity diagram* berdasarkan sistem yang diusulkan:

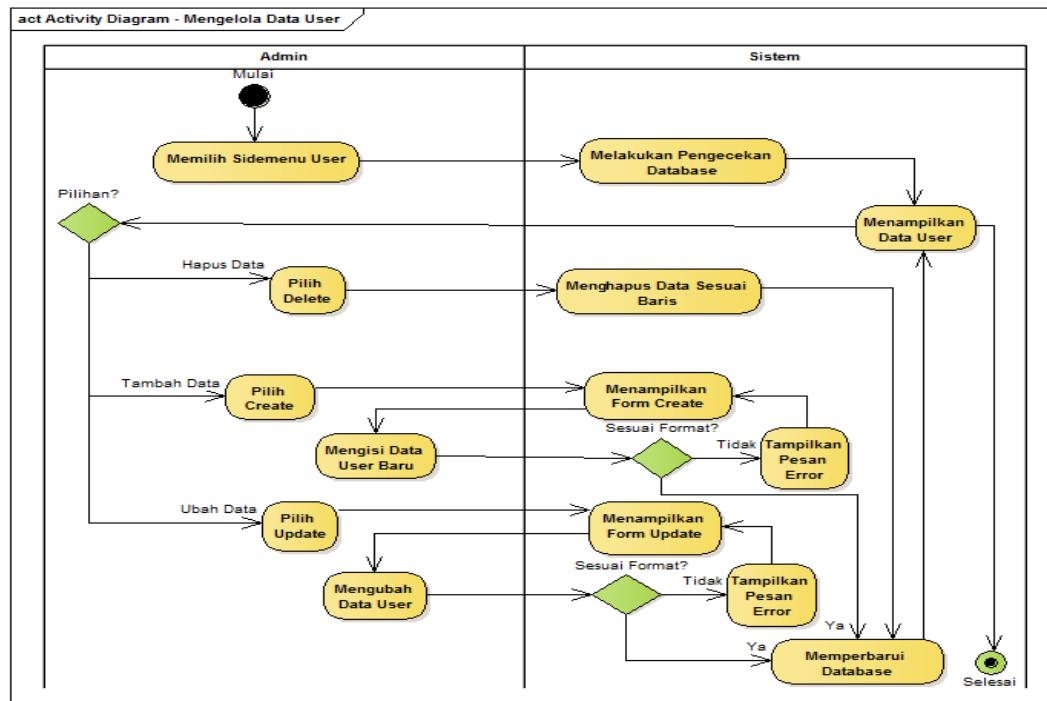
- *Activity Diagram* Melakukan *Log in*



Gambar V.3 *Activity Diagram Log In*

Sumber: Hasil Analisis (2019)

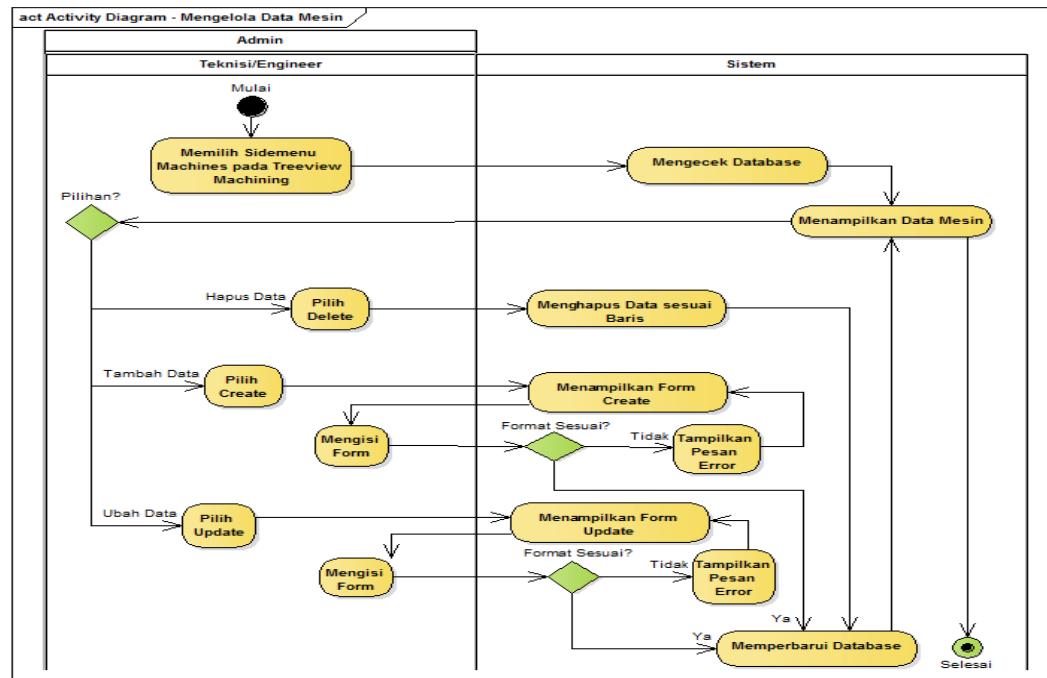
- *Activity Diagram Mengelola Data Pengguna*



Gambar V.4 *Activity Diagram Mengelola Data Pengguna*

Sumber: Hasil Analisis (2019)

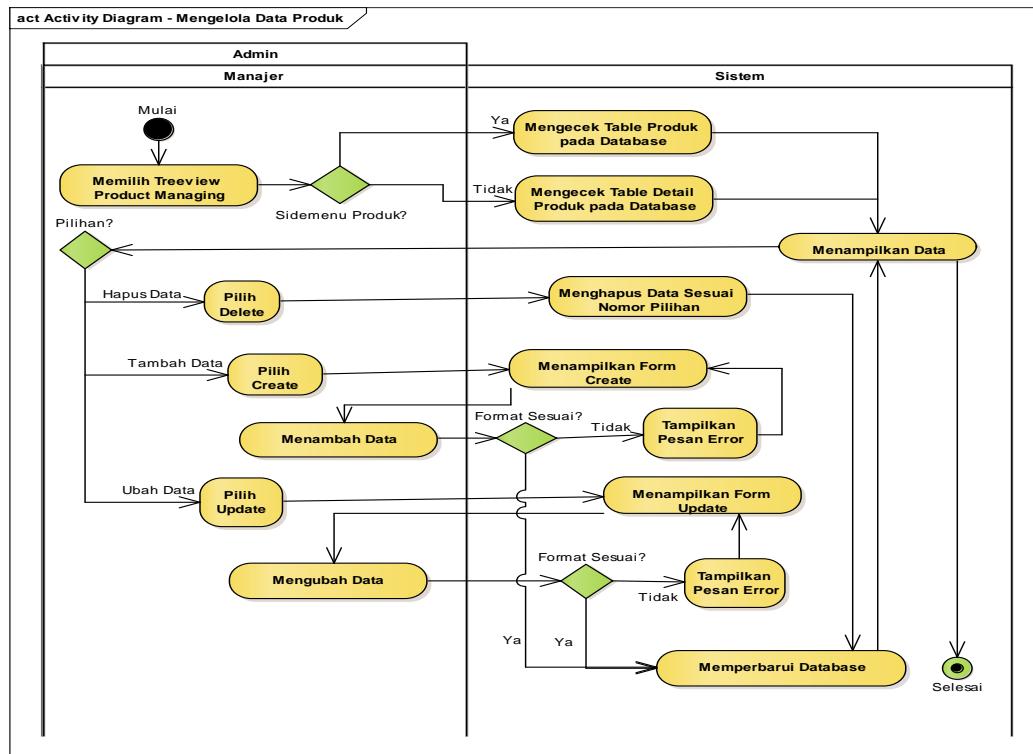
- *Activity Diagram Mengelola Data Mesin*



Gambar V.5 *Activity Diagram Mengelola Data Mesin*

Sumber: Hasil Analisis (2019)

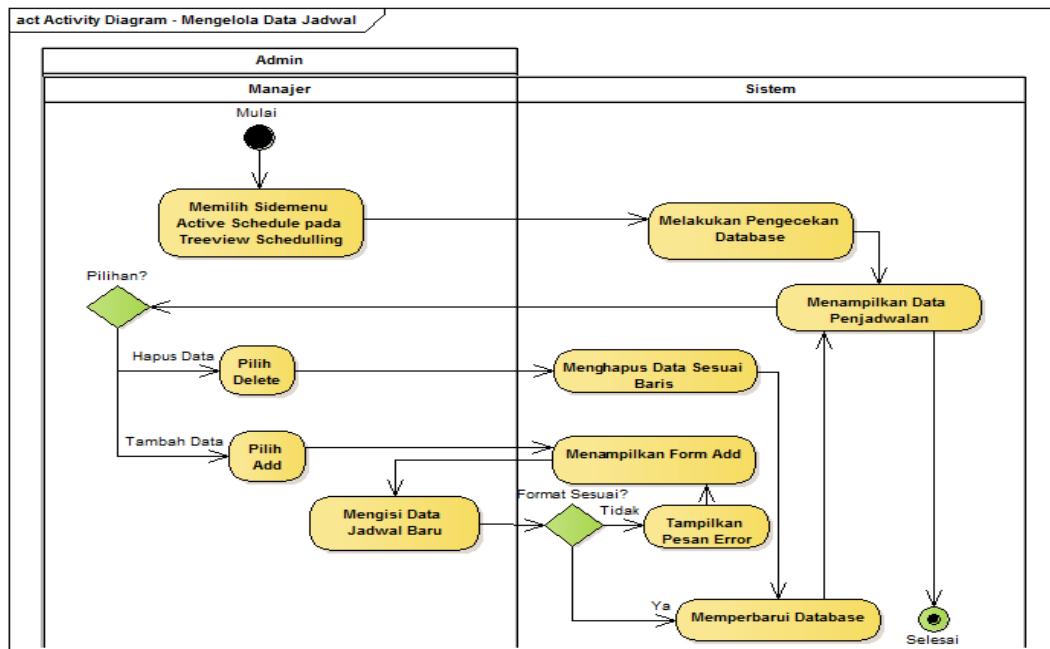
- *Activity Diagram Mengelola Data Produk*



Gambar V.6 *Activity Diagram Mengelola Data Produk*

Sumber: Hasil Analisis (2019)

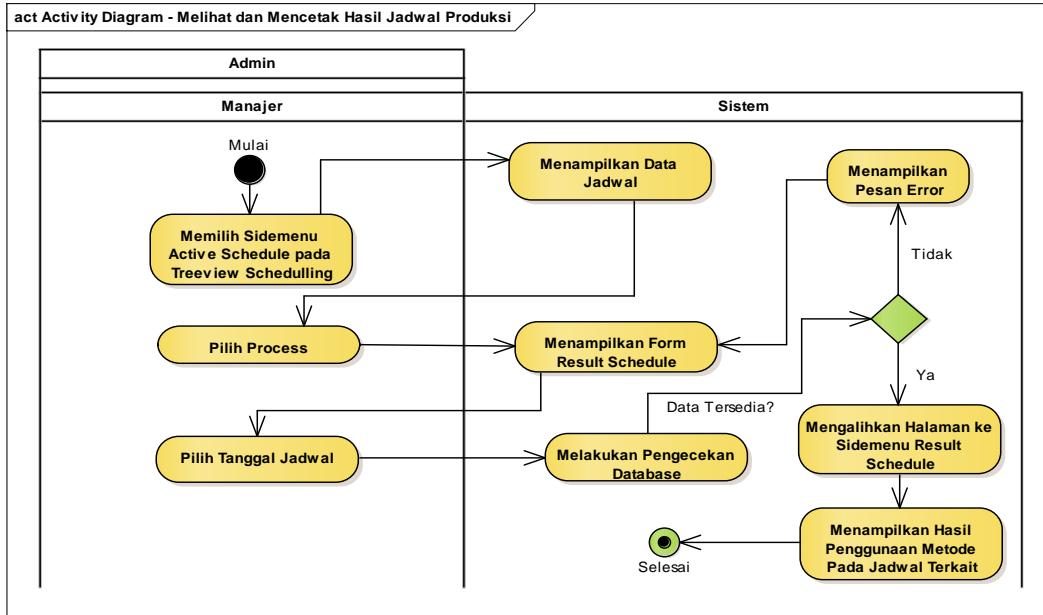
- *Activity Diagram Mengelola Jadwal Produksi*



Gambar V.7 *Activity Diagram Mengelola Jadwal Produksi*

Sumber: Hasil Analisis (2019)

- *Activity Diagram Melihat Hasil dan Mencetak Urutan Proses Produksi*

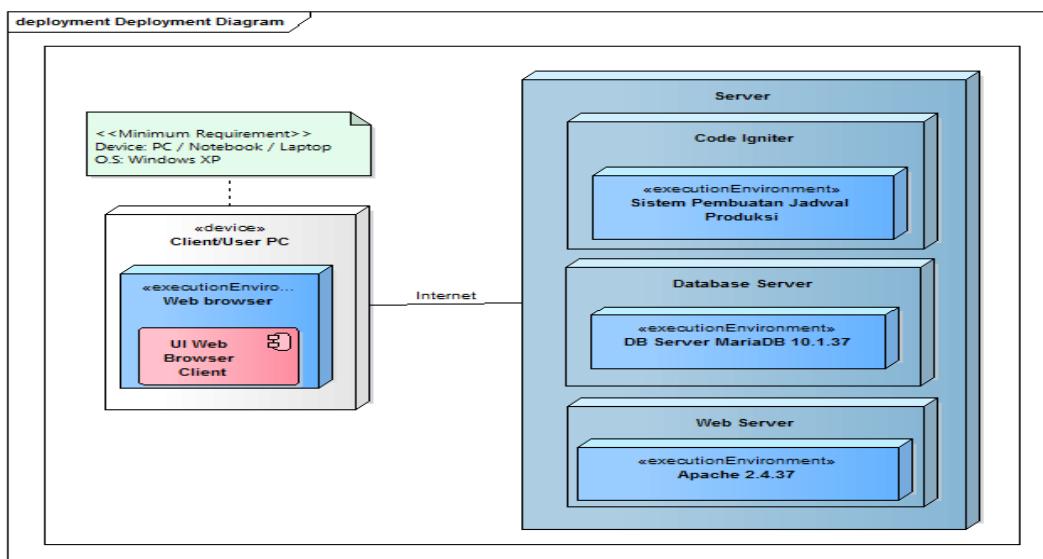


Gambar V.8 *Activity Diagram Melihat dan Mencetak Hasil Urutan Proses Produksi*

Sumber: Hasil Analisis (2019)

5.3.3. Deployment Diagram

Deployment diagram merupakan sebuah *diagram* yang berfungsi untuk menunjukkan komponen perangkat lunak yang digunakan ketika diimplementasikan ke dalam arsitektur fisik. Pada Gambar V.9 dibawah ini terdapat *deployment diagram* berdasarkan sistem yang diusulkan:



Gambar V.9 *Deployment Diagram* Sistem Usulan

Sumber: Hasil Analisis (2019)

Penjelasan dari *deployment diagram* pada Gambar V.9 yang menerangkan sistem informasi usulan penjadwalan produksi adalah sebagai berikut:

1. *Client* adalah komputer *user* atau pengguna yang sudah terinstal *web browser*. Contoh *browser* yang sering digunakan yaitu seperti Google Chrome, Mozilla Firefox, Safari atau Internet Explorer. *Browser* tersebut kemudian akan digunakan sebagai media untuk menjalankan aplikasi sistem informasi penjadwalan produksi.
2. *Server* aplikasi sistem informasi penjadwalan produksi terdiri dari *web server* menggunakan Apache 2.4.37, *application server* yang sudah tersedia dalam aplikasi serta *database server* untuk melakukan *store data* menggunakan MySQL Maria DB 10.1.37.

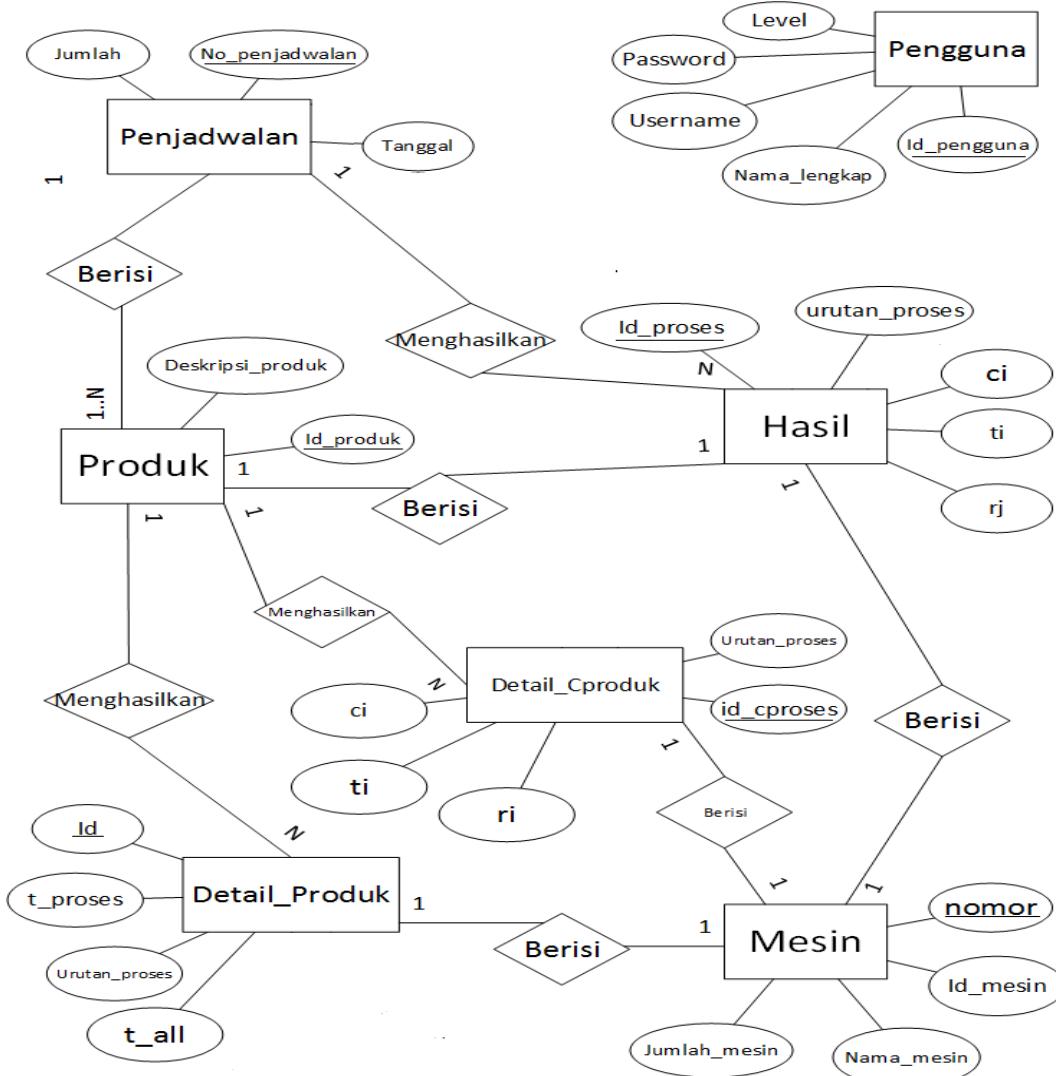
5.4. Pemodelan Data Sistem Usulan

Pemodelan data merupakan metode yang digunakan untuk memberikan analisis dari persyaratan data yang diperlukan agar dapat mendukung proses bisnis suatu organisasi. Tujuan utama dilakukan pemodelan dari sistem yang diusulkan adalah untuk menjelaskan secara rinci hubungan dan isi yang terdapat dalam basis data.

Pada sistem penjadwalan produksi usulan digunakan dua pemodelan data yaitu *Entity Relationship Diagram* (ERD) dan Kamus Data. Penggunaan *entity relationship diagram* dilakukan dengan tujuan untuk menjelaskan hubungan yang terdapat pada basis data sedangkan penggunaan kamus data bertujuan untuk menjelaskan isi dari basis data yang digunakan dalam sistem usulan.

5.4.1. Entity Relationship Diagram (ERD)

ERD merupakan model yang digunakan untuk menjelaskan hubungan antar data dalam penggunaan basis data. Untuk menggambarkan ERD diperlukan notasi-notasi beserta simbol-simbol yang memiliki arti masing-masing. Berikut ini merupakan ERD sistem informasi penjadwalan produksi usulan yang dapat dilihat pada Gambar V.10.



Gambar V.10 ERD Sistem Usulan
Sumber: Hasil Analisis (2019)

5.4.2. Kamus Data

Kamus data digunakan untuk mendefinisikan isi yang terkandung dalam basis data suatu sistem secara lengkap. Kamus data sistem informasi penjadwalan produksi yang diusulkan adalah sebagai berikut:

1. Spesifikasi Tabel Pengguna

Nama Tabel : pengguna

Akronim : Pengguna

Fungsi : Untuk menyimpan data pengguna yang digunakan sebagai hak akses ketika melakukan *log in*.

Tipe : *Data Master*

Tabel V.12 Tabel Pengguna

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Pengguna	id_pengguna	int	11	<i>Primary key</i>
2	Nama Lengkap	nama_lengkap	varchar	25	
3	Username	username	varchar	10	
4	Password	password	varchar	100	
5	Jabatan	level	int	1	

Sumber: Hasil Analisis (2019)

2. Spesifikasi Tabel Detail Produk

Nama Tabel : detail_produk

Akronim : Detail Produk

Fungsi : Untuk menyimpan data produk secara detail.

Tipe : *File* data manajer

Tabel V.13 Tabel Detail Produk

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID	id	int	11	<i>Primary key</i>
2	ID Produk	id_produk	varchar	11	<i>Foreign key</i>
3	ID Mesin	id_mesin	char	4	<i>Foreign key</i>
4	Urutan Proses	urutan_proses	int	11	
5	Waktu Proses	t_proses	decimal	10,1	
6	Total Waktu Proses	t_all	decimal	10,1	

Sumber: Hasil Analisis (2019)

3. Spesifikasi Tabel Produk

Nama Tabel : produk

Akronim : Produk

Fungsi : Untuk menyimpan data produk.

Tipe : *File* data manajer

Tabel V.14 Tabel Produk

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Produk	id_produk	varchar	11	<i>Primary key</i>
2	Nama Produk	deskripsi_produk	varchar	25	

Sumber: Hasil Analisis (2019)

4. Spesifikasi Tabel Mesin

Nama Tabel : mesin

Akronim : Mesin

Fungsi : Untuk menyimpan data mesin yang digunakan sebagai acuan dalam tabel detail produk.

Tipe : *File data engineer*

Tabel V.15 Tabel Mesin

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID	nomor	int	11	<i>Primary key</i>
2	ID Mesin	id_mesin	char	4	
3	Nama Mesin	nama_mesin	varchar	40	
4	Jumlah Mesin	jumlah_mesin	int	11	

Sumber: Hasil Analisis (2019)

5. Spesifikasi Tabel Penjadwalan

Nama Tabel : penjadwalan

Akronim : Jadwal

Fungsi : Untuk menyimpan data penjadwalan yang telah dilakukan implementasi metode *active schedule*.

Tipe : *File data manajer*

Tabel V.16 Tabel Penjadwalan

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	No Jadwal	no_penjadwalan	int	11	<i>Primary key</i>
2	Tanggal	tanggal	Date		
3	ID Produk	id_produk	varchar	50	<i>Foreign key</i>
4	Jumlah	jumlah	int	11	

Sumber: Hasil Analisis (2019)

6. Spesifikasi Tabel Hasil

Nama Tabel : hasil

Akronim : Hasil

Fungsi : Untuk melihat data penjadwalan yang telah dilakukan penerapan metode *active schedule*.

Tipe : *File data manajer*

Tabel V.17 Tabel Hasil

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID	id_proses	int	11	<i>Primary key</i>
2	ID Produk	id_produk	varchar	11	<i>Foreign key</i>
3	Urutan Proses	urutan_proses	Int	11	
4	Waktu Awal Proses	ci	decimal	10,1	
5	Waktu Proses	ti	decimal	10,1	
6	Waktu Terpilih	rij	decimal	10,1	

Sumber: Hasil Analisis (2019)

7. Spesifikasi Tabel Detail CProduk (Metode)

Nama Tabel : detail_cproduk

Akronim : Detail CProduk (Metode)

Fungsi : Untuk menyimpan data sementara pada metode.

Tipe : *Data Dummy*

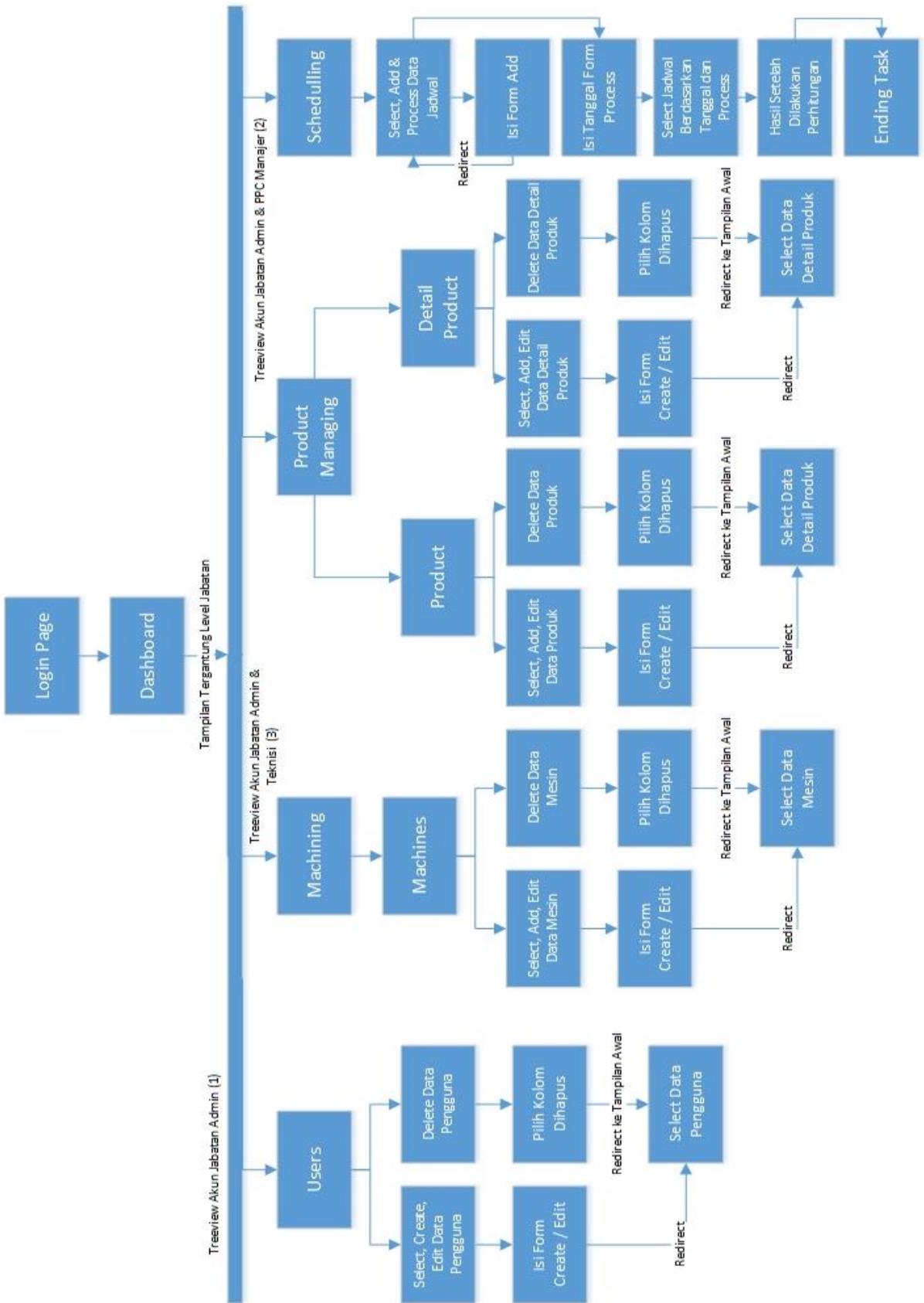
Tabel V.18 Tabel Detail CProduk

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID	Id_cproses	int	11	<i>Primary key</i>
2	ID Produk	id_produk	varchar	11	<i>Foreign key</i>
3	ID Mesin	id_mesin	char	4	<i>Foreign key</i>
4	Urutan Proses	urutan_proses	int	11	
5	Waktu Awal Proses	ci	decimal	10,1	
6	Waktu Proses	ti	decimal	10,1	
7	Total ci+ti	ri	decimal	10,1	

Sumber: Hasil Analisis (2019)

5.5. Windows Navigation Diagram (WND)

Penggunaan WND pada sistem usulan digunakan untuk menunjukkan bagaimana halaman dalam aplikasi berinteraksi antara satu dengan yang lain. Berikut merupakan WND dari sistem informasi penjadwalan produksi yang dapat dilihat pada Gambar V.11.



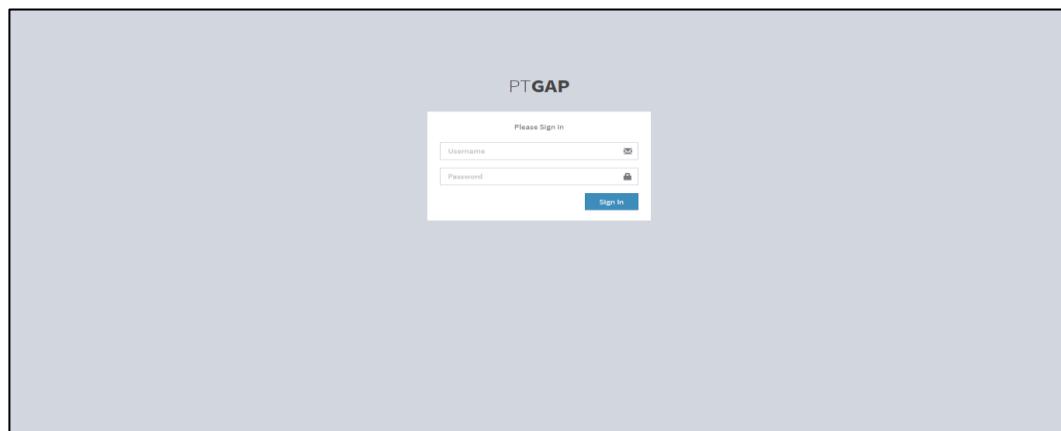
Gambar V.11 *Windows Navigation Diagram Sistem Usulan*

5.6. Perancangan Antarmuka Sistem

Perancangan antarmuka (*interface*) digunakan untuk menampilkan dan menjelaskan tampilan dari tiap-tiap bagian dari aplikasi yang sudah dibuat. Berikut merupakan antarmuka sistem informasi penjadwalan produksi yang diusulkan.

1. Form Log in

Halaman ini merupakan halaman awal sebelum masuk ke dalam program. Pengguna diwajibkan melakukan *log in* terlebih dahulu agar bisa menggunakan aplikasi. Antarmuka dari *form log in* dapat dilihat pada Gambar V.12 berikut.



Gambar V.12 UI Log in

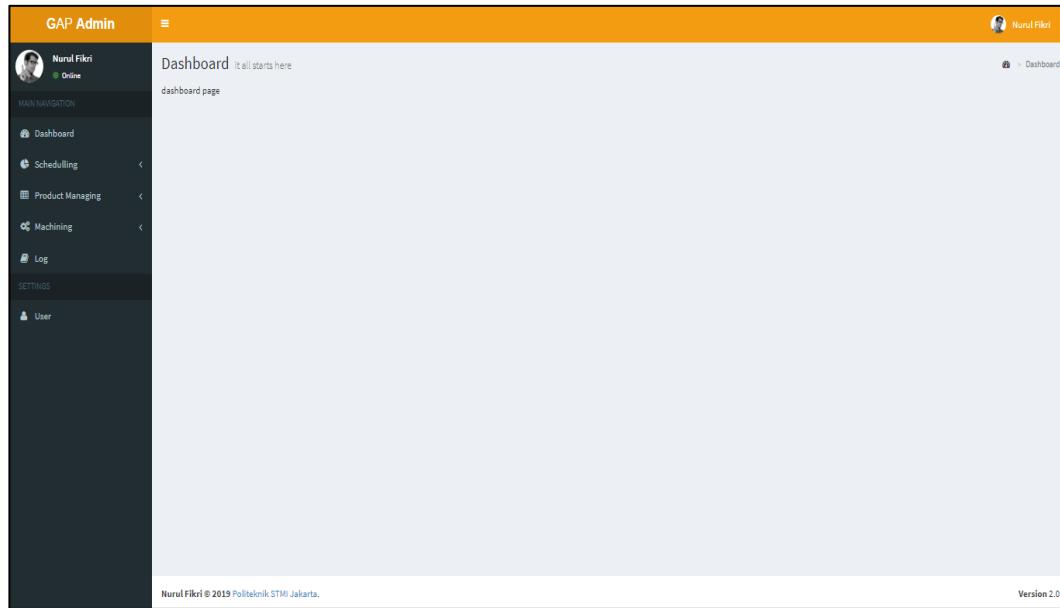
Sumber: Hasil Analisis (2019)

2. Halaman Utama (*Dashboard*)

Halaman utama merupakan halaman dimana pengguna dapat menggunakan fitur-fitur yang tersedia sesuai dengan jabatan dan kepentingan penggunanya masing-masing. Berikut merupakan antarmuka dari halaman utama pada sistem informasi penjadwalan produksi yang dibagi berdasarkan jabatan masing-masing pengguna:

a. Halaman utama admin

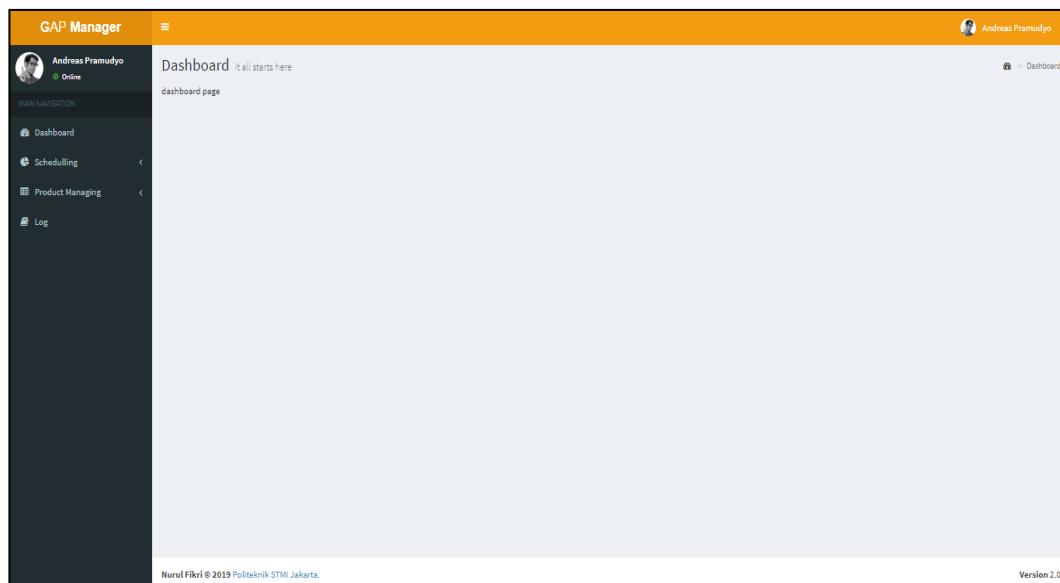
Admin memiliki wewenang penuh terhadap aplikasi sistem informasi penjadwalan produksi. Halaman utama admin dapat menggunakan segala fungsi-fungsi yang ada seperti mengelola data pengguna, mengelola data mesin, mengelola data produk serta membuat jadwal produksi (dapat dilihat pada Gambar V.13). Pengguna dari akun admin adalah direktur atau wakil direktur.



Gambar V.13 UI Dashboard Admin
Sumber: Hasil Analisis (2019)

b. Halaman utama karyawan (Bagian PPC Manajer)

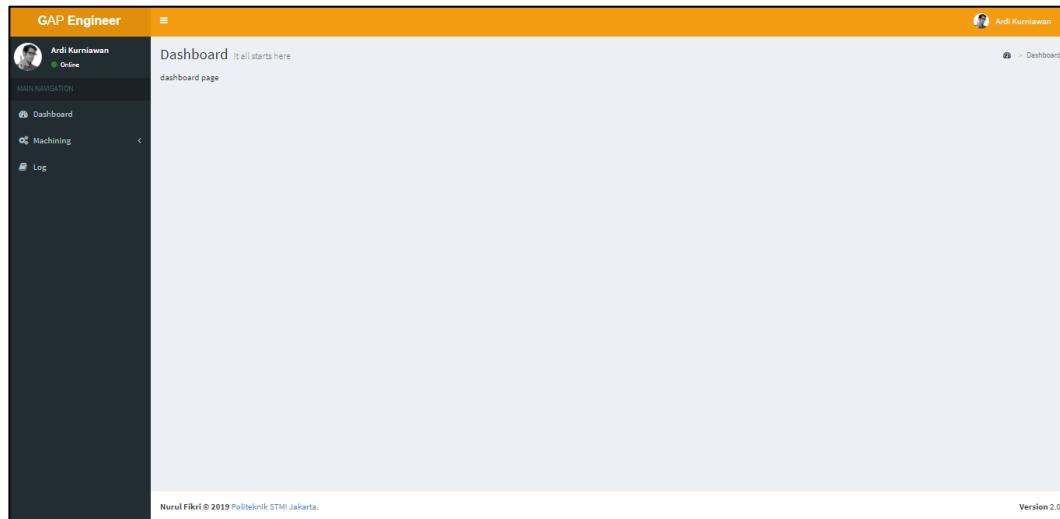
Bagian PPC Manajer memiliki wewenang dalam mengatur kegiatan yang berkaitan dengan penjadwalan dan produksi. Fungsi seperti mengelola data produk, detail dari data produk, hingga penjadwalan produksi dapat (dapat dilihat pada Gambar V.14) dilakukan oleh pengguna yang memiliki tingkat jabatan PPC Manajer.



Gambar V.14 UI Dashboard PPC Manager
Sumber: Hasil Analisis (2019)

c. Halaman utama karyawan (Bagian Teknisi)

Fungsi dan fitur pada aplikasi yang dapat digunakan oleh Bagian Teknisi (dapat dilihat pada Gambar V.15) antara lain menambah data mesin, mengubah data mesin dan menghapus data mesin.



Gambar V.15 *UI Dashboard Engineer*

Sumber: Hasil Analisis (2019)

3. *Form Data Pengguna*

Form ini digunakan untuk menampilkan data pengguna yang tersedia pada basis data (dapat dilihat pada Gambar V.16).

#	Username	Name	Level	Action
1	fikri	Nurul Fikri	Admin	Edit Delete
2	andreas	Andreas Pramudyo	Manager	Edit Delete
3	aardi	Ardi Kurniawan	Engineer	Edit Delete

Gambar V.16 *UI Form Data Pengguna*

Sumber: Hasil Analisis (2019)

4. Form Tambah dan Ubah Data Pengguna

Form ini digunakan untuk menambah dan mengubah data pengguna yang terdapat pada basis data (dapat dilihat pada Gambar V.17).

Add Users	
Name *	<input type="text"/>
Username *	<input type="text"/>
Password *	<input type="password"/>
Password Confirmation *	<input type="password"/>
Level *	<input type="text" value="Pilih"/>
<input type="button" value="Save"/> <input type="button" value="Reset"/>	

Edit Users	
Name *	<input type="text" value="Nurul Fikri"/>
Username *	<input type="text" value="fikri"/>
Password *	<input type="password"/>
Password Confirmation	<input type="password"/>
Level	<input type="text" value="Manajer"/>
<input type="button" value="Save"/> <input type="button" value="Reset"/>	

* Harus diberi isi.

Gambar V.17 UI Form Tambah dan Ubah Data Pengguna

Sumber: Hasil Analisis (2019)

5. Form Data Mesin

Form ini digunakan untuk menampilkan data mesin yang tersedia pada basis data (dapat dilihat pada Gambar V.18).

#	No Mesin	Nama Mesin	Jumlah	Action
1	M001	FONG HO	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	M002	BENCH LATE SD 32	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	M003	HOBING MC	5	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	M004	CNC FOCUS 120	3	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
5	M005	BENCH LATE SD 25	6	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
6	M006	AMADA	1	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
7	M007	CNC VICTOR	1	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
8	M008	WEST LATES	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
9	M009	ROLLING MC	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
10	M010	CNC FOCUS 140	2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Showing 1 to 10 of 11 entries

Previous 1 2 Next

Nurul Fikri © 2019 Politeknik STMI Jakarta.

Version 2.0

Gambar V.18 UI Form Data Mesin

Sumber: Hasil Analisis (2019)

6. Form Tambah dan Ubah Data Mesin

Form ini digunakan untuk menambah dan mengubah data mesin yang terdapat pada basis data (dapat dilihat pada Gambar V.19).

The screenshot shows the 'Machine' section of the GAP Admin interface. It displays two forms side-by-side:

- Add Machine:** A form for creating new machine entries. It has three input fields: 'Machine Code *' (empty), 'Machine Name *' (empty), and 'Machine Quantity *' (empty). Below the fields are 'Save' and 'Reset' buttons.
- Edit Machine:** A form for modifying existing machine entries. It also has three input fields: 'Machine Code *' (M002), 'Machine Name *' (FONG HO), and 'Machine Quantity *' (2). Below the fields are 'Save' and 'Reset' buttons. A note at the bottom states: '*Harus diisi'.

Gambar V.19 UI Form Tambah dan Ubah Data Mesin

Sumber: Hasil Analisis (2019)

7. Form Data Produk

Form ini digunakan untuk menampilkan data produk yang tersedia pada basis data (dapat dilihat pada Gambar V.20).

The screenshot shows the 'Product' section of the GAP Admin interface. It displays a table of products:

#	Kode Produk	Nama Produk	
1	0014	SHAFT KICK	Edit Delete
2	0068	SLEEVE	Edit Delete
3	1910	WASHER	Edit Delete
4	2561	COLLAR	Edit Delete

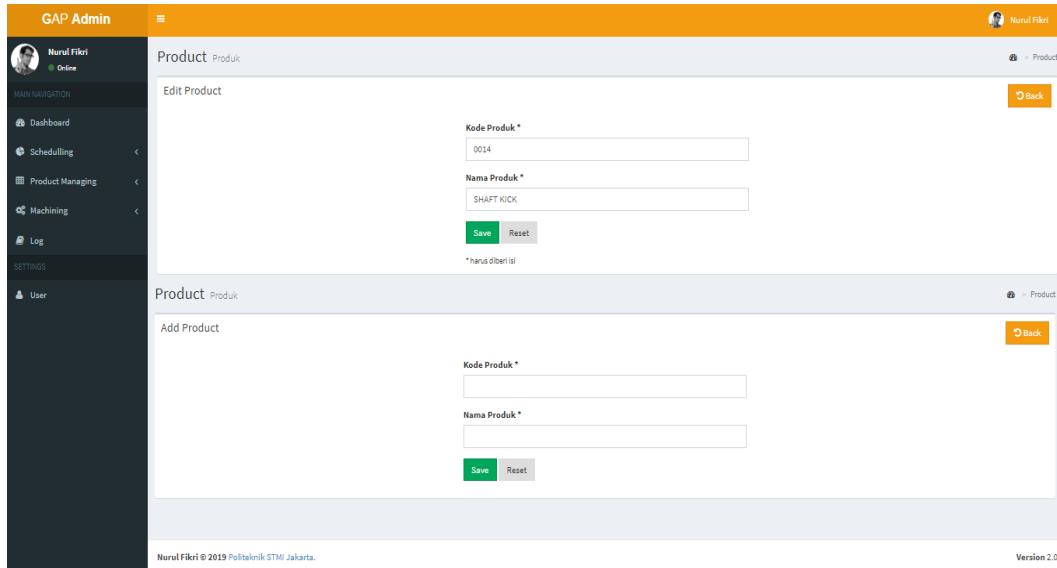
At the top right of the table, there is a 'Create' button. At the bottom left, it says 'Nurul Fikri © 2019 Politeknik STMI Jakarta.', and at the bottom right, it says 'Versi 2.0'.

Gambar V.20 UI Form Data Produk

Sumber: Hasil Analisis (2019)

8. Form Tambah dan Ubah Data Produk

Form ini digunakan untuk menambah dan mengubah data produk yang terdapat pada basis data (dapat dilihat pada Gambar V.21).



Gambar V.21 UI Form Tambah dan Ubah Data Produk
Sumber: Hasil Analisis (2019)

9. Form Data Detail Produk

Form ini digunakan untuk menampilkan data detail produk yang tersedia pada basis data (dapat dilihat pada Gambar V.22). Berbeda dengan *form* sebelumnya *form* disini menggabungkan data yang terdapat pada tabel mesin dan produk kemudian menampilkannya sesuai dengan jenis produk dan mesin yang sama.

#	Tipe Produk	Nama Produk	Nama Mesin	Urutan Proses	Waktu Proses	Action
1	0068	SLEEVE	BENCH LATE SD 32	1	28.0	Edit Delete
2	0068	SLEEVE	CF 12A	2	26.4	Edit Delete
3	2561	COLLAR	FONG HO	1	12.1	Edit Delete
4	2561	COLLAR	BENCH LATE SD 25	2	33.0	Edit Delete
5	2561	COLLAR	CNC FOCUS 120	3	19.2	Edit Delete
6	1910	WASHER	AMADA	1	22.9	Edit Delete
7	0014	SHAFT KICK	FONG HO	1	12.8	Edit Delete

Gambar V.22 UI Form Data Detail Produk
Sumber: Hasil Analisis (2019)

10. Form Tambah Detail Produk

Form ini digunakan untuk menambah data produk yang terdapat pada basis data (dapat dilihat pada Gambar V.23).

Product Detail

Add Detail Product

Kode Produk *

Kode Mesin *

Urutan Proses *

Waktu Proses *

Save Reset

Gambar V.23 UI Form Tambah Detail Produk

Sumber: Hasil Analisis (2019)

11. Form Data Jadwal

Form ini digunakan untuk menampilkan data penjadwalan yang telah ditambahkan sebelumnya (dapat dilihat pada Gambar V.24). Pada form ini ditampilkan kolom lain dengan cara menggabungkan id produk yang terdapat pada data produk.

GAP Admin

Schedule Jadwal

Data Schedule

#	Tanggal	Tipe Produk	Nama Produk	Nama Mesin	Urutan Proses	Waktu Proses	Jumlah	Action
1	2019-07-30	0014	SHAFT KICK	FONG HO	1	12.8	100	
2	2019-07-30	0068	SLEEVE	BENCH LATE SD 32	1	28.0	100	
3	2019-07-30	0068	SLEEVE	CF 12A	2	28.4	100	
4	2019-07-31	0014	SHAFT KICK	FONG HO	1	12.8	250	
5	2019-07-31	0014	SHAFT KICK	FONG HO	1	12.8	100	
6	2019-07-31	0068	SLEEVE	BENCH LATE SD 32	1	28.0	100	
7	2019-07-31	0068	SLEEVE	CF 12A	2	28.4	100	
8	2019-08-01	2561	COLLAR	FONG HO	1	12.1	100	
9	2019-08-01	2561	COLLAR	CNC FOCUS 120	3	19.2	100	
10	2019-08-01	2561	COLLAR	CNC FOCUS 120	4	19.2	100	

NurulFikri © 2019 Politeknik STMI Jakarta. Version 2.0

Gambar V.24 UI Form Data Jadwal

Sumber: Hasil Analisis (2019)

12. Form Tambah Jadwal

Form ini digunakan untuk menambah data penjadwalan yang terdapat pada basis data (dapat dilihat pada Gambar V.25).

The screenshot shows the 'Schedule' section of the GAP Admin interface. On the left is a sidebar with navigation links: Dashboard, Scheduling, Product Managing, Machining, Log, and User. The main area is titled 'Schedule Jadwal' and contains a form for adding a new schedule. The form includes fields for 'Tanggal *' (date), 'Kode Produk *' (product code), and 'Jumlah *' (quantity). At the bottom are two buttons: 'Add' and 'Process'.

Gambar V.25 UI Form Tambah Jadwal
Sumber: Hasil Analisis (2019)

13. Form Data Hasil

Form ini digunakan untuk memproses hasil penjadwalan berdasarkan tanggal (dapat dilihat pada Gambar V.26).

The screenshot shows the 'Result' section of the GAP Admin interface. The sidebar includes links for Dashboard, Scheduling, Product Managing, Machining, Log, and User. The main area is titled 'Result Hasil' and displays a table titled 'Schedule Result'. The table has columns: No, Nama Produk (Product Name), Urutan Proses (Processing Order), Nama Mesin (Machine Name), Waktu Awal (Start Time), Waktu Proses (Processing Time), and Total Waktu Proses (Total Processing Time). The table shows 10 entries out of 17 total. At the bottom right are buttons for 'Previous', '1', '2', and 'Next'.

No	Nama Produk	Urutan Proses	Nama Mesin	Waktu Awal	Waktu Proses	Total Waktu Proses
1	COLLAR-2561	1	FONG HO	0.00	1812.00	1812.00
2	SLEEVE-0068	1	BENCH LATE SD 32	0.00	2800.00	2800.00
3	COLLAR-2561	2	BENCH LATE SD 25	1812.00	1850.00	3462.00
4	SLEEVE-0068	2	CF 12A	2800.00	1056.00	3856.00
5	SHAFT KICK-0014	1	FONG HO	1812.00	2160.00	3972.00
6	SHAFT KICK-0014	2	BENCH LATE SD 25	3972.00	762.50	4734.50
7	COLLAR-2561	3	CNC FOCUS 120	3462.00	1920.00	5382.00
8	COLLAR-2561	4	CNC FOCUS 120	5382.00	1920.00	7302.00
9	SHAFT KICK-0014	3	HOBING MC	4734.50	5520.00	10254.50
10	COLLAR-2561	5	AMADA	7302.00	4680.00	11982.00

Gambar V.26 UI Form Data Hasil
Sumber: Hasil Analisis (2019)

5.7. Implementasi Metode *Active Schedule* Pada Sistem

Sistem penjadwalan produksi yang menggunakan metode *Active Schedule* pada bahasa pemrograman PHP dengan *framework* CodeIgniter dapat diimplementasikan dengan baris-baris kode pemrograman di bawah ini yang diletakkan pada *controller* CodeIgniter:

```
public function active_schedule()
{
    $this->_rules();
    //hapus seluruh hasil
    $del = $this->Result_m->delete_all_data();
    //update tabel by data tanggal
    //panggil tabel terkait (jadwal, detail_produk, mesin)
    //mesin
    $tgl = $this->input->post('tanggal');
    //penjadwalan
    $param2 = $this->Jadwal_m->get($tgl);
    if ($param2->num_rows()<1) {
        echo "<script>alert('Jadwal tidak ada');</script>";
        echo "<script>window.location='".site_url('jadwal/hasil')."';</script>";
    } else {
        foreach ($param2->result() as $b){
            $id_produk = $b->id_produk,
            $jml = $b->jumlah;
            $param3 = $this->Dproduk_m->getp($id_produk);
            foreach ($param3->result() as $b){
                $id_produk = $b->id_produk;
                $id_mesin = $b->id_mesin;
                $t_proses = $b->t_proses;
                $urutan = $b->urutan_proses;
                $param4 = $this->Mesin_m->getm($id_mesin);
                foreach ($param4->result() as $b){
                    $jumlah_mesin = $b->jumlah_mesin;
                }
            }
        }
    }
}
```

```

$t_all = ($t_proses * $jml) / $jumlah_mesin;

$params = $this->db->query("UPDATE detail_produk SET t_all
= '$t_all' WHERE id_produk = '$id_produk' and urutan_proses = '$urutan'");
}

}

}

//inisialisasi
$proses = $this->Metode_m->get_by_awal_tanggal($tgl);
"Inisialisasi Awal Tabel <br>";
foreach ($proses as $row){
    $row->id ."," .
    $row->id_produk ."," .
    $row->id_mesin ."," .
    $row->urutan_proses ."," .
    $row->t_proses ."," .
    $row->t_all ."<br>";
    //set nilai ci, cek apakah proses sebelumnya sudah dijalankan pada tabel
    t_hasil dan ambil nilai $rj
    // $prosesb4      =      $this->T_hasil_model->get_by_prosesb4($row-
    >job,$row->no_proses, $row->mesin);
    $waktu_awal = 0;
    //if ($prosesb4){
        // $ci=$prosesb4->rj;
    //}
    $total_waktu = $row->t_all;
    $waktu_akhir = $waktu_awal+$total_waktu;
    $data = array(
        'id_produk' => $row->id_produk,
        'urutan_proses' => $row->urutan_proses,
}

```

```

'id_mesin' => $row->id_mesin,
'ci' => $waktu_awal,
'ti' => $total_waktu,
'ri' => $waktu_akhir,
);
$this->Dummy_m->insert($data);
}

$c_id=0;
$c_job = 0;
$c_proses = 0;
//end of inisialisasi
$i=0;
$n = $this->Metode_m->total_rows();
while ($i<$n){

    //tampilkan isi dari c_proses
    "<br>Looping ke-".($i+1)."<br>";
    $cp = $this->Dummy_m->get_all();
    foreach ($cp as $row){

        $row->id_cproses ."," .
        $row->id_produk ."," .
        $row->id_mesin ."," .
        $row->urutan_proses ."," .
        $row->ci ."," .
        $row->ti ."," .
        $row->ri ."<br>";}

    //ambil ri paling kecil
    $min_ri = $this->Dummy_m->get_by_min_ri();
    if ($min_ri){
}

```

```

$min_ri->id_cproses .",".

$min_ri->id_produk .",".

$min_ri->urutan_proses .",".

$min_ri->id_mesin .",".

$min_ri->ri ."<br>";

//tambahkan min ri ke dalam insert

$data = array(
    'id_produk' => $min_ri->id_produk,
    'urutan_proses' => $min_ri->urutan_proses,
    'id_mesin' => $min_ri->id_mesin,
    'ci' => $min_ri->ci,
    'ti' => $min_ri->ti,
    'rj' => $min_ri->ri,
);

$this->Result_m->insert($data);

//simpan nilai id_proses yang terpilih
$c_id=$min_ri->id_cproses;
$c_job=$min_ri->id_produk;
$c_proses=$min_ri->urutan_proses;
$c_mesin=$min_ri->id_mesin;
$c_ri = $min_ri->ri;

//update tabel c_proses menggantikan nilai yang tadi
//delete job dan proses yang sudah terpilih
$this->Dummy_m->delete($c_id);
//mengupdate ci dan ri dari proses yang telah ada di cproses
$cp2 = $this->Dummy_m->get_all();
foreach ($cp2 as $row){
    if ($row->id_mesin == $c_mesin)
        if ($row->ci <$c_ri){


```

```

//update ci dan ri dari baris tersebut
$data = array(
'id_produk' => $row->id_produk,
'urutan_proses' => $row->urutan_proses,
'id_mesin' => $row->id_mesin,
'ci' => $c_ri,
'ti' => $row->ti,
'ri' => $row->ti+$c_ri, );
$this->Dummy_m->update($row->id_cproses,$data); }

//mencari proses selanjutnya dari job yang terpilih
$next_proses = $this->Metode_m->get_next_process($c_job,$c_proses+1);
if ($next_proses){ //cari ri dari next_proses
$prosesb4      =      $this->Result_m->get_by_process_before($next_proses->id_produk, $next_proses->urutan_proses, $next_proses->id_mesin);
$ci = 0;
if ($prosesb4){
    $ci=$prosesb4->rj; }
$ti= $next_proses->t_all;
$ri = $ci+$ti;
$data = array(
'id_produk' => $next_proses->id_produk,
'urutan_proses' => $next_proses->urutan_proses,
'id_mesin' => $next_proses->id_mesin,
'ci' => $ci,
'ti' => $ti,
'ri' => $ri, );
$this->Dummy_m->insert($data); } }

$i=$i+1; }

echo "<script>alert('Jadwal berhasil diproses');</script>";
echo "<script>window.location='".site_url('hasil')."';</script>"; }

}

```

5.9. Implementasi Sistem

Untuk mendukung kebutuhan implementasi sistem diperlukan suatu spesifikasi perangkat lunak (*software*) maupun perangkat keras (*hardware*) yang sesuai. Adapun spesifikasinya adalah sebagai berikut:

1. Analisis Kebutuhan *Software*
 - a. Sistem Operasi : Microsoft Windows 7 Home Premium
 - b. *Database Server* : MariaDB 10.1.37
 - c. *Web Server* : PHP version 7.3.2
 - d. Bahasa Pemrograman : PHP
 - e. *Framework* : CodeIgniter 3.1.10
2. Analisis Kebutuhan *Hardware*
 - a. *Processor* : Processor Intel® Core i3
 - b. RAM : RAM 2 GB
 - c. *Harddisk* : Harddisk 214 GB
 - d. Peralatan : Mouse, Keyboard, Monitor.

5.8. Pengujian *Black Box Testing*

Black box testing merupakan suatu metode pengujian pada perangkat lunak yang dilakukan untuk menguji perangkat lunak akan kesiapan fungsi-fungsi yang telah dibuat. Pengujian *black box testing* dilaksanakan menggunakan pedoman *use case* yang telah dibuat sebelumnya. Berikut ini merupakan pengujian yang dilakukan terhadap aplikasi sistem penjadwalan produksi.

- a. *Functional Test Use Case Melakukan Log in*
Berikut merupakan *functional test* dari *use case* melakukan *log in* yang dapat dilihat pada Tabel V.19.
- | | |
|------------------------|--|
| <i>Test Case ID</i> | : Login001 |
| <i>Function</i> | : Melakukan validasi saat pengguna <i>log in</i> |
| <i>Data Assumption</i> | : Fungsi validasi saat pengguna melakukan login berfungsi dengan baik. |
| <i>Description</i> | : Melakukan <i>log in</i> ke dalam sistem dengan menguji kesalahan dalam <i>input username</i> dan <i>password</i> . |

Tabel V.19. Test Case Melakukan Log in

Test ID	Test Case Name	Description	Expected Result	Actual Result
001	Validasi saat log in	Melakukan <i>log in</i> dengan melakukan <i>input username</i> dan <i>password</i> menggunakan kombinasi huruf, angka, atau simbol sebanyak 1 karakter	Berhasil <i>log in</i>	Sesuai
002	Validasi saat log in	Melakukan <i>log in</i> dengan memasukkan <i>input username</i> dan <i>password</i> menggunakan kombinasi huruf, angka, atau simbol sebanyak 8 karakter	Berhasil <i>log in</i>	Sesuai
003	Validasi saat log in	Melakukan <i>log in</i> dengan memasukkan <i>input username</i> yang benar dan <i>password</i> yang salah	Muncul pesan kesalahan dan mengalihkan ke halaman <i>log in</i>	Sesuai
004	Validasi saat log in	Melakukan <i>log in</i> dengan memasukkan <i>input username</i> yang salah dan <i>password</i> yang benar	Muncul pesan kesalahan dan mengalihkan ke halaman <i>log in</i>	Sesuai
005	Validasi saat log in	Melakukan <i>log in</i> dengan tidak mengisi kolom <i>username</i> atau <i>password</i>	Muncul pesan kolom belum diisi	Sesuai
006	Validasi saat log in	Melakukan <i>log in</i> dengan <i>input username</i> dan <i>password</i> yang benar	Berhasil <i>log in</i>	Sesuai

Tabel V.19. *Test Case* Melakukan *Log in* (Lanjutan)

Test ID	Test Case Name	Description	Expected Result	Actual Result
007	Validasi saat <i>log in</i>	Melakukan <i>log in</i> dengan <i>input username</i> dan <i>password</i> yang salah	Muncul pesan kesalahan dan kembali ke halaman <i>log in</i>	Sesuai

Sumber: Hasil Analisis (2019)

b. *Functional Test Use Case* Mengelola Data Pengguna

Pada *use case* mengelola data pengguna terdiri dari melakukan tambah data, ubah data dan hapus data. Berikut merupakan *test case* mengelola data pengguna (dapat dilihat pada Tabel V.20).

- Test Case ID* : Mengelola Data Pengguna001
Function : Melakukan kelola data pengguna
Data Assumption : Fungsi operasi kelola data pengguna berjalan dengan baik dan data masuk ke dalam basis data
Description : Melakukan proses tambah data, ubah data, hapus data serta menampilkan data sesuai basis data.

Tabel V.20. *Test Case* Mengelola Data Pengguna

Test ID	Test Case Name	Description	Expected Result	Actual Result
001	Menampilkan fitur <i>user</i>	Menampilkan fitur <i>user</i> ketika pengguna menggunakan akun dengan <i>level admin</i>	Menu ditampilkan	Sesuai
002	Menampilkan fitur <i>user</i>	Menampilkan fitur <i>user</i> ketika pengguna tidak menggunakan akun dengan <i>level admin</i>	Menu disembunyikan	Sesuai
003	Menampilkan data <i>user</i>	Menampilkan data akun pengguna sesuai dengan basis data	Data ditampilkan	Sesuai

Tabel V.20. *Test Case* Mengelola Data Pengguna (Lanjutan)

<i>Test ID</i>	<i>Test Case Name</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
004	Tambah data <i>user</i>	Melakukan tambah data <i>user</i> tanpa mengisi semua <i>form</i> tambah data	Muncul pesan kesalahan kolom belum diisi	Sesuai
005	Tambah data <i>user</i>	Melakukan tambah data <i>user</i> tanpa mengisi salah satu kolom pada <i>form</i> tambah data	Muncul pesan kesalahan kolom belum diisi	Sesuai
006	Tambah data <i>user</i>	Melakukan tambah data <i>user</i> dengan melakukan <i>input password</i> dan <i>password confirmation</i> yang berbeda	Muncul pesan kesalahan <i>password</i> dan <i>passconf</i> harus sama	Sesuai
007	Tambah data <i>user</i>	Melakukan tambah data dengan <i>input</i> gabungan huruf, angka dan simbol sebanyak 6 karakter	Data berhasil ditambah	Sesuai
011	Ubah data <i>user</i>	Mengklik tombol ubah pada salah satu data <i>user</i> , dan menampilkan <i>form</i> ubah berisi data <i>user</i> tersebut	<i>Form</i> tampil	Sesuai
012	Ubah data <i>user</i>	Melakukan ubah data dengan mengkosongkan kolom <i>username</i> dan nama lengkap pada <i>form</i> ubah data	Muncul pesan kesalahan kolom harus diisi	Sesuai
013	Ubah data <i>user</i>	Melakukan ubah data dengan mengkosongkan kolom <i>password</i> dan <i>password confirmation</i>	Data berhasil diubah	Sesuai

Tabel V.20. *Test Case* Mengelola Data Pengguna (Lanjutan)

Test ID	Test Case Name	Description	Expected Result	Actual Result
014	Ubah data <i>user</i>	Melakukan ubah data dengan mengisi <i>password</i> dan <i>password confirmation</i> dengan data berbeda	Muncul pesan kesalahan <i>password</i> dan <i>passconf</i> harus sama	Sesuai
015	Ubah data <i>user</i>	Melakukan ubah data dengan <i>input</i> gabungan angka, huruf atau simbol sebanyak 5 karakter	Data berhasil diubah	Sesuai
016	Ubah data <i>user</i>	Melakukan ubah data dengan <i>input</i> gabungan angka, huruf atau simbol dibawah 5 karakter	Muncul pesan kesalahan <i>username</i> dan <i>password</i> minimal 5 karakter	Sesuai

Sumber: Hasil Analisis (2019)

c. *Functional Test Use Case* Mengelola Data Mesin

Pada *use case* mengelola data mesin terdiri dari melakukan tambah data, ubah data dan hapus data. Berikut merupakan *test case* mengelola data mesin (dapat dilihat pada Tabel V.21).

- Test Case ID* : Mengelola Data Mesin 001
Function : Melakukan kelola data mesin
Data Assumption : Fungsi mengelola data mesin dan form validasi kesalahan berjalan dengan baik dan data dapat masuk ke dalam basis data.
Description : Melakukan proses tambah data, ubah data, hapus data serta menampilkan data sesuai basis data.

Tabel V.21. Test Case Mengelola Data Mesin

<i>Test ID</i>	<i>Test Case Name</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Menampilkan menu data mesin	Menampilkan fitur data mesin apabila menggunakan akun dengan <i>level</i> jabatan admin dan bagian teknik mesin (<i>engineer</i>)	Menu ditampilkan	Sesuai
002	Menghilangkan menu data mesin	Menghilangkan fitur data mesin apabila menggunakan akun dengan <i>level</i> jabatan selain admin dan bagian teknik mesin (<i>engineer</i>)	Menu disembunyikan	Sesuai
003	Menampilkan data mesin	Menampilkan data mesin sesuai dengan basis data ketika pengguna mengklik menu mesin	Data ditampilkan	Sesuai
004	Hapus data mesin	Menghapus data mesin sesuai baris yang diinginkan dengan mengklik tombol <i>delete</i> pada baris tersebut	Data berhasil dihapus	Sesuai
005	Tambah data mesin	Menampilkan <i>form</i> tambah data dengan mengklik tombol <i>create</i>	<i>Form</i> berhasil muncul	Sesuai
006	Tambah data mesin	Menambah data dengan mengkosongkan kolom pada <i>form</i> tambah data	Muncul pesan kesalahan kolom harus diisi	Sesuai
007	Tambah data mesin	Menambah data dengan mengisi kolom nama mesin dengan data duplikat	Muncul pesan kesalahan data nama mesin tidak boleh sama	Sesuai

Tabel V.21. *Test Case* Mengelola Data Mesin (Lanjutan)

<i>Test ID</i>	<i>Test Case Name</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
008	Tambah data mesin	Menambah data dengan mengisi kolom kode mesin dengan data duplikasi	Muncul pesan kesalahan data kode mesin tidak boleh sama	Sesuai
009	Tambah data mesin	Menambah data mesin dengan mengisi kolom kode mesin dan nama mesin menggunakan gabungan huruf, angka atau simbol dibawah 4 karakter.	Muncul pesan kesalahan jumlah data <i>input</i> minimal 4 karakter	Sesuai
010	Tambah data mesin	Menambah data mesin dengan mengisi kolom kode mesin dan nama mesin menggunakan gabungan huruf, angka atau simbol diatas 3 karakter	Data berhasil ditambahkan	Sesuai
011	Tambah data mesin	Menambah data mesin dengan mengisi semua kolom	Data berhasil ditambahkan	Sesuai
012	Ubah data mesin	Menampilkan <i>form</i> ubah data beserta data di dalamnya dengan mengklik tombol <i>edit</i> pada baris yang hendak diubah	<i>Form</i> berhasil muncul	Sesuai
013	Ubah data mesin	Mengubah data dengan mengkosongkan kolom pada <i>form</i> ubah data	Muncul pesan kesalahan kolom harus diisi	Sesuai

Sumber: Hasil Analisis (2019)

d. *Functional Test Use Case Mengelola Data Produk*

Pada *use case* mengelola data produk terdiri dari melakukan tambah data, ubah data dan hapus data. Berikut merupakan *test case* mengelola data produk (dapat dilihat pada Tabel V.22).

<i>Test Case ID</i>	: Mengelola Data Produk 001
<i>Function</i>	: Melakukan kelola data mesin
<i>Data Assumption</i>	: Fungsi mengelola data produk berjalan dengan baik dan data dapat masuk ke dalam basis data.
<i>Description</i>	: Melakukan proses tambah data, ubah data, hapus data serta menampilkan data sesuai basis data.

Tabel V.22. *Test Case Mengelola Data Produk*

<i>Test ID</i>	<i>Test Case Name</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Menampilkan menu data produk	Menampilkan fitur data produk apabila pengguna menggunakan akun dengan <i>level</i> jabatan admin dan bagian PPC Manajer	Menu ditampilkan	Sesuai
002	Menghilangkan menu data produk	Menghilangkan fitur data produk apabila pengguna menggunakan akun dengan <i>level</i> jabatan selain admin dan bagian PPC Manajer	Menu disembunyikan	Sesuai
003	Menampilkan data produk	Menampilkan data produk sesuai dengan basis data ketika pengguna mengklik menu produk	Data ditampilkan	Sesuai
004	Hapus data produk	Menghapus data produk sesuai baris yang diinginkan dengan mengklik tombol <i>delete</i> pada baris tersebut	Data berhasil dihapus	Sesuai

Tabel V.22. *Test Case* Mengelola Data Produk (Lanjutan)

<i>Test ID</i>	<i>Test Case Name</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
005	Tambah data produk	Menampilkan <i>form</i> tambah data dengan mengklik tombol <i>create</i>	<i>Form</i> berhasil muncul	Sesuai
006	Tambah data produk	Menambah data dengan mengkosongkan kolom pada <i>form</i> tambah data	Muncul pesan kesalahan kolom harus diisi	Sesuai
007	Tambah data produk	Menambah data dengan mengisi kolom kode produk dengan data duplikat	Muncul pesan kesalahan data kode produk tidak boleh sama	Sesuai
008	Tambah data produk	Menambah data dengan mengisi <i>form</i> tambah data dengan angka, huruf atau simbol sebanyak 1 karakter	Data berhasil ditambah	Sesuai
009	Tambah data produk	Menambah data dengan mengisi <i>form</i> tambah data dengan angka, huruf atau simbol diatas 5 karakter	Data berhasil ditambah	Sesuai
010	Ubah data produk	Menampilkan <i>form</i> ubah data beserta data di dalamnya ketika mengklik <i>edit</i> pada baris yang hendak diubah	<i>Form</i> berhasil muncul	Sesuai
011	Ubah data produk	Mengubah data dengan mengkosongkan kolom pada <i>form</i> ubah data	Muncul pesan kesalahan kolom harus diisi	Sesuai

Tabel V.22. *Test Case* Mengelola Data Produk (Lanjutan)

Test ID	Test Case Name	Description	Expected Result	Actual Result
012	Ubah data produk	Mengubah data dengan mengisi <i>form</i> ubah data dengan angka, huruf atau simbol sebanyak 1 karakter	Data berhasil diubah	Sesuai
013	Ubah data produk	Mengubah data dengan mengisi <i>form</i> ubah data dengan angka, huruf atau simbol diatas 5 karakter	Data berhasil diubah	Sesuai

Sumber: Hasil Analisis (2019)

e. *Functional Test Use Case* Mengelola Data Detail Produk

Pada *use case* mengelola data detail produk selain terdiri dari melakukan tambah data, ubah data dan hapus data juga terdapat fungsi tambahan yaitu menampilkan data berdasarkan data dalam tabel produk. Berikut merupakan *test case* mengelola data detail produk (dapat dilihat pada Tabel V.23).

Test Case ID : Mengelola Data Detail Produk 001

Function : Melakukan kelola data mesin

Data Assumption : Fungsi mengelola data detail produk berjalan dengan baik dan data dapat disimpan di basis data.

Description : Melakukan proses tambah data, ubah data, hapus data serta menampilkan data sesuai basis data.

Tabel V.23. *Test Case* Mengelola Data Detail Produk

Test ID	Test Case Name	Description	Expected Result	Actual Result
001	Menampilkan menu data detail produk	Menampilkan fitur data detail produk apabila pengguna memakai akun dengan <i>level</i> jabatan admin dan bagian PPC Manajer	Menu ditampilkan	Sesuai

Tabel V.23. Test Case Mengelola Data Detail Produk (Lanjutan)

<i>Test ID</i>	<i>Test Case Name</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
002	Meng-hilangkan menu data detail produk	Menghilangkan fitur data detail produk apabila pengguna menggunakan akun dengan <i>level</i> jabatan selain admin dan bagian PPC Manajer	Menu disembunyikan	Sesuai
003	Menampilkan data detail produk	Menampilkan data detail produk sesuai dengan data produk dan pada data detail produk di dalam basis data ketika pengguna mengklik menu detail produk	Data ditampilkan	Sesuai
004	Hapus data detail produk	Menghapus data detail produk sesuai baris yang diinginkan dengan mengklik tombol <i>delete</i> pada baris tersebut	Data berhasil dihapus	Sesuai
005	Tambah data detail produk	Menampilkan <i>form</i> tambah data dengan mengklik tombol <i>create</i>	<i>Form</i> berhasil muncul	Sesuai
006	Tambah data produk	Menambah data detail produk dengan mengkosongkan kolom pada <i>form</i> tambah data	Muncul pesan kesalahan kolom harus diisi	Sesuai
007	Tambah data produk	Menambah data detail produk dengan menggunakan huruf, angka atau simbol sebanyak 1 karakter pada <i>form</i> tambah data	Data berhasil ditambahkan	Sesuai

Tabel V.23. *Test Case* Mengelola Data Detail Produk (Lanjutan)

<i>Test ID</i>	<i>Test Case Name</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
008	Tambah data produk	Menambah data detail produk dengan menggunakan huruf, angka atau simbol sebanyak 5 karakter pada <i>form</i> tambah data	Data berhasil ditambahkan	Sesuai
009	Ubah data produk	Menampilkan <i>form</i> ubah data detail produk sesuai baris yang diinginkan dengan mengklik tombol <i>edit</i> pada baris tersebut	<i>Form</i> berhasil muncul	Sesuai

Sumber: Hasil Analisis (2019)

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil analisis yang telah dilakukan dalam penelitian Tugas Akhir ini dapat diambil beberapa kesimpulan antara lain sebagai berikut:

1. Sistem informasi penjadwalan produksi sudah diimplementasikan secara terkomputerisasi dan berbasis *web* sehingga sudah tidak lagi menghitung secara manual.
2. Pada sistem informasi penjadwalan produksi yang baru sudah diimplementasikan metode *active schedule* sehingga sudah tidak menggunakan intuisi karyawan dalam membuat urutan proses produksi yang mengakibatkan pembuatan jadwal produksi menggunakan pendekatan yang sama di setiap produk.
3. Berdasarkan hasil analisis, perhitungan urutan proses produksi menggunakan metode *active schedule* mengakibatkan pengurangan waktu dalam menghitung urutan proses produksi karena menggunakan pendekatan yang sama pada tiap produk. Meskipun begitu, penggunaan metode *active schedule* menghasilkan total waktu produksi yang lebih lama jika dibandingkan dengan penggunaan yang menggunakan pendekatan berbeda-beda pada tiap produk yang telah diterapkan perusahaan sebelumnya.

6.2. Saran

Saran yang diberikan untuk pengembangan sistem informasi pengiriman barang jadi ini selanjutnya adalah sebagai berikut:

1. Mengintegrasikan penjadwalan produksi saat ini dengan Bagian Pengadaan sehingga dapat dengan segera mengetahui stok tersedia dan membuat pembelian bahan baku.
2. Membandingkan metode *active schedule* dengan metode lain untuk penjadwalan produksi seperti metode *non-delay* atau metode *heuristic*.

DAFTAR PUSTAKA

- Anhar. (2010). *PHP & MySql Secara Otodidak*. Jakarta: PT TransMedia.
- Aryono, C. (2011). *Penjadwalan N Job M Mesin Dalam Proses Pembuatan Produk Ass Mesin Jahit Dan Baut Bearing Di PT. Sinar Sakti Matra Nusantara Bandung*. Bandung: Unikom.
- Darmawan, D., & Fauzi, N. K. (2013). *Sistem Informasi Manajemen*. Bandung: PT Remaja Rosdakarya.
- Dennis, A., Wixom, B. H., & Roth, R. M. (2013). *System Analysis and Design* (5th ed.). New Jersey: John Wiley & Sons, Inc.
- Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *System Analysis & Design An Object-Oriented Approach with UML*. Danvers: John Wiley & Sons, Inc.
- Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *System Analysis and Design an Object Oriented with UML*. WILEY.
- Fajriyah, Josi, A., & Fisika, T. (Jurnal SISFOKOM). *Rancang Bangun Sistem Informasi Tender Karet Desa Jungai Menggunakan Metode Waterfall*. 2017.
- Firdianti, A. (2018). *Implementasi Manajemen Berbasis Sekolah*. Yogyakarta: CV Gre Publishing.
- Gata, W., & Gata, G. (2013). *Sukses Membangung Aplikasi Penjualan dengan Java*. Jakarta: Elex Media.
- Gelinas, U., & Dull, B. R. (2012). *Accounting Information Systems*, 9th ed. USA: South Western Cengage Learning.
- Hakim, L. (2010). *Membangun Web Berbasis PHP dengan Framework CodeIgniter*. Yogyakarta: Lokomedia.
- Hidayat, R. (2010). *Cara Praktis Membuat Web*. Elex Media Komputindo.
- Indrajani. (2011). *Perancangan Basis Data dalam All in I*. Jakarta: PT Elex Media Komputindo.
- Indrajani. (2015). *Database Design (Case Study All in One)*. Jakarta: PT Elex Media Komputindo.

- Krismiaji. (2010). *Sistem Informasi Akuntansi*. Yogyakarta: Unit Penerbit Sekolah Tinggi Ilmu Manajemen YKPN.
- Mardi. (2014). *Sistem Informasi Akuntansi*. Bogor: Ghalia Indonesia.
- Mulyadi. (2016). *Sistem Informasi Akuntansi*. Jakarta: Salemba Empat.
- Nidhra, S., & Dondeti, J. (2012). *Black Box and White Box Testing Technique - A Literature Review*. International Journal of Embedded System and Application (IJESA).
- Pinedo, M. L. (2016). *Planning and Scheduling in Manufacturing and Services*.
- Purbadian, Y. (2016). *Trik Cepat Membangun Aplikasi Berbasis Web Dengan Framework CodeIngiter*. Yogyakarta: Andi Offset.
- Romney, M. B., & Steinbart, P. J. (2014). *Sistem Informasi Akuntansi: Accounting Information System (Edisi 13)*. Prentice Hall.
- Rosa, A. S., & Salahuddin, M. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- Rosa, A. S., & Salahuddin, M. (2013). *Analisa dan Desain Sistem Informasi*. Yogyakarta: Andi Offset.
- Rosa, A. S., & Salahuddin, M. (2013). *Rekayasa Perangkat Lunak Struktur dan Berorientasi Objek*. Bandung: Informatika.
- Rosa, A. S., & Salahuddin, M. (2014). *Rekayasa Perangkat Lunak .* Bandung: Informatika.
- Rosa, A. S., & Salahuddin, M. (2014). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Rusdiana, A., & Irfan, M. (2014). *Sistem Informasi Manajemen*. Bandung: Pustaka Setia.
- Saputra, A. (2011). *Trik dan Solusi Jitu Pemrograman PHP*. Jakarta: PT. Elex Media Komputindo.
- Sidik, B. F. (2017, April 17). *Apa Itu MariaDB Dan Apa Bedanya Dengan MySQL*. Retrieved Juli 28, 2019, from <http://www.kursuswebsite.org: http://www.kursuswebsite.org/apa-itu-mariadb-dan-apa-bedanya-dengan-mysql/>

- Stair, M. R., & Reynolds, G. W. (2010). *Principles of Information System : A Managerial Approach (9th ed.)*. Australia: Thomson Course Technology.
- Susanto, A. (2013). *Sistem Informasi Akuntansi*. Bandung: Lingga Jaya.
- Suseno, & Indrakusuma, B. (2014). Job Scheduling Menggunakan Metode Algoritma Active, Algoritma Non Delay Dan Heuristic Schedule Generation.
- Sutabri, T. (2012). *Analisis Sistem Informasi*. Yogyakarta: Andi.
- Sutarman. (2012). *Pengantar Teknologi Informasi*. Jakarta: PT Bumi Aksara.
- Turban, E., & Volonino, L. (2010). *Information Technology for Management*. John Wiley & Sons, Inc.
- Widharma, I. S., & Sajayasa, I. (2014). Penjadwalan Waktu beban Kerja dengan Metode Algoritma Active Schedule dan Heuristic Schedule untuk Efisiensi Daya Listrik.
- Yakub. (2012). *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Yaqin, M. A., Munawaroh, Q. N., Cahyanti, F. D., & Syaifullah, M. B. (2014). Optimasi Penjadwalan Produksi pada Game Hay Day dengan Kriteria Minimalisasi Waktu, Minimalisasi Biaya dan Experience Menggunakan Metode Algoritma Active Schedule Generation.