

**PEMBANGUNAN DASHBOARD PRODUKSI PADA PTPN XII  
PERSERO KANTOR WILAYAH III GUNUNG GAMBIR BERBASIS  
WEB**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Sofyan Syahri Huzaini

145150201111009



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2018

## PENGESAHAN

PEMBANGUNAN DASHBOARD PRODUKSI PADA PTPN XII PERSERO KANTOR  
WILAYAH III GUNUNG GAMBIR BERBASIS WEB

SKRIPSI


Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer


Disusun Oleh :  
Sofyan Syahri Huzaini  
NIM: 145150201111009

Skripsi ini telah diuji dan dinyatakan lulus pada  
26 Desember 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1

Dosen Pembimbing 2


  
Bayu Priyambadha, S.Kom, M.Kom  
NIP: 19820909 200812 1 004

  
Denny Sagita Rusdianto, S.Kom, M.Kom  
NIP: 19851124 201504 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



  
Tri Astoro Kurniawan, S.T., M.T., Ph.D.  
NIP: 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiaris, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 26 Desember 2018



Sofyan Syahri Huzaini

NIM: 145150201111009



## PRAKATA

Puji syukur selalu penulis panjatkan kehadirat Allah SWT karena atas limpahan rahmat dan kasih sayang-Nya, penulisan skripsi ini dapat penulis selesaikan. Salawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW hingga di akhir masa.

Penulisan skripsi ini tidak dapat penulis selesaikan tepat waktu tanpa dukungan morel dan materiel dari orang-orang terdekat penulis. Pada kata pengantar ini, penulis ingin menyampaikan rasa terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Bayu Priyambadha, S.Kom, M.Kom selaku Dosen Pembimbing I yang telah banyak membantu dan mengarahkan dalam proses penulisan skripsi ini.
5. Bapak Denny Sagita Rusdianto, S.Kom, M.Kom selaku Dosen Pembimbing II yang telah meluangkan waktunya untuk memeriksa dan memberikan masukan berharga bagi penulisan skripsi ini.
6. Bapak dan ibu yang selalu memberikan dukungan morel serta materiel bagi penulis.
7. Dan teman-teman penulis yang tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari bahwa dalam proses penulisan skripsi ini masih terdapat kesalahan dan kekurangan di beberapa bagian. Karena itu kritikan dan saran yang membangun sangat penulis harapkan demi perbaikan dan pengembangan skripsi ini di masa mendatang. Semoga skripsi ini bermanfaat bagi siapapun yang membaca dan menerapkannya dalam kehidupan sehari-hari.

Malang, 26 Desember 2018

Penulis

sofyanshuzaini@gmail.com

## ABSTRAK

**Sofyan Syahri Huzaini, Pembangunan Dashboard Produksi Pada PTPN XII Persero Kantor Wilayah III Gunung Gambir Berbasis Web**

**Pembimbing: Bayu Priyambadha, S.Kom, M.Kom dan Denny Sagita Rusdianto, S.Kom, M.Kom**

PT Perkebunan Nusantara (PTPN) XII Gunung Gambir merupakan Badan Usaha Milik Negara (BUMN) yang bergerak di bidang perkebunan yang terletak di kabupaten Jember. Saat ini Gunung Gambir mengelola komoditas karet dan teh. Setiap hari hasil produksi yang diperoleh dilaporkan secara manual menggunakan Handi Talky serta dalam pembuatan laporan masih menggunakan aplikasi Microsoft Exel. Penyampaian data hasil produksi yang disampaikan antar satu tempat ke tempat lain sering mengalami ketidaksesuaian data dikarenakan salah mendengar informasi dari pengirim informasi. Data yang dikirim dibacakan secara manual dari laporan yang sebelumnya telah di rekap di dalam aplikasi Microsoft Exel. Hal ini menyebabkan masalah berupa data yang disampaikan tidak praktis dan tidak akurat serta proses penyampaian data yang lambat. Perekapan laporan produksi dengan menggunakan aplikasi Microsoft Exel juga dirasa sulit dalam membaca laporan, menganalisa data, dan memprediksi data sebab informasi yang ditampilkan sebagian besar berupa angka. Mengatasi masalah tersebut dibuatlah sistem dashboard produksi berbasis web yang diharapkan dapat memudahkan proses pengiriman data serta memudahkan dalam memonitoring semua proses produksi dengan tampilan yang mudah dan menarik sehingga dapat membantu dalam menentukan strategi untuk kedepannya. Penelitian ini dimulai dari proses analisis kebutuhan sistem yang dilakukan dengan melakukan wawancara dan observasi langsung terhadap kebun gunung gambir. Kemudian dilakukan tahap perancangan yang terbagi menjadi perancangan arsitektur, perancangan komponen, perancangan data dan perancangan antarmuka. kemudian tahap implementasi merealisasikan sistem yang akan dibangun sesuai dengan perancangan yang telah dibuat, keduanya dilakukan dengan pendekatan Object Oriented. Kemudian tahap pengujian dilakukan Pengujian Unit, Pengujian Intregrasi, dan Pengujian Validasi.

**Kata kunci:** monitoring, dashboard, produksi



## ABSTRACT

**Sofyan Syahri Huzaini, Development Dashboard Production at PTPN XII Persero Regional Office III Mount Gambir Based on the Web**

**Supervisor: Bayu Priyambadha, S.Kom M.Kom and Denny Sagita Rusdianto, S.Kom, M.Kom**

*PTPN XII Mount Gambir is a state-owned corporation that active in plantation industry located in Jember. Now Mount Gambir manage tea and latex comodities. Everyday the outcome of production report by manual use Handy Talky and to make the report still use Microsoft Excel. The reportation of the outcome data is reported between one place to another place that usually encounter unfitted data because of the miss heard of information from the sender. The data that have sent is repeated by manual from prior report have recaped in microsoft excel. It can causes problems such as the not accurate and not simple and then the process will take long time. The recapotulation of production report by Microsoft Excel also difficult to read, to analyze, and to predict the data because most of information that shown in the form of numbers. To solve that problems we make production6 dashboard system based on web that it will be expected to simplify the sending process of data then to simplify the monitoring all the production process by simple and interesting appearance. So it can help to determine the next strategy. This research start from analyzing process of the requirement system by interview and observation directly in Mount Gambir. The next step is planning that can be divide to architectur planning, component planning, data planning anf interface planning. The next step is implementation that relize the system that built before. Both steps are done by object oriented approach. The next step is unit testing, integration testing, and validation testing.*

**Keywords:** *monitoring, dashboard, production*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	3
1.5 Batasan masalah .....	3
1.6 Sistematika penulisan.....	3
BAB 2 TINJAUAN PUSTAKA.....	5
2.1 Kajian Pustaka.....	5
2.2 Monitoring .....	5
2.3 <i>Dashboard</i> .....	6
2.3.1 <i>Digital Dashboard</i> .....	7
2.3.2 Portal Perusahaan .....	10
2.3.3 Karakteristik <i>Dashboard</i> .....	11
2.3.4 Tujuan Penggunaan <i>Dashboard</i> .....	12
2.4 Konsep Dasar Website .....	12
2.5 PHP.....	13
2.6 Pendekatan <i>Object Oriented</i> .....	14
2.6.1 Pengertian Pendekatan <i>Object Oriented</i> .....	14
2.6.2 UML ( <i>Unified Modeling Language</i> ) .....	14
2.7 Pengujian .....	17
2.7.1 <i>White Box Testing</i> .....	17

2.7.2 Black Box Testing.....	17
2.8 System Development Life Cycle .....	18
2.8.1 Waterfall .....	18
BAB 3 METODOLOGI .....	20
3.1 Studi Literatur .....	21
3.2 Analisis Kebutuhan .....	21
3.3 Perancangan dan Implementasi .....	22
3.3.1 Perancangan Sistem .....	22
3.3.2 Implementasi Sistem.....	23
3.4 Pengujian .....	23
3.5 Kesimpulan.....	24
BAB 4 ANALISIS KEBUTUHAN .....	25
4.1 Elisitasi Kebutuhan Sistem.....	25
4.1.1 Gambaran Umum Sistem.....	27
4.2 Spesifikasi Kebutuhan.....	27
4.2.1 Identifikasi Aktor .....	28
4.2.2 Daftar Kebutuhan Fungsional .....	29
4.2.3 Daftar Kebutuhan <i>Non-Fungsional</i> .....	32
4.2.4 <i>Use Case Diagram</i> .....	33
4.2.5 <i>Use Case Scenario</i> .....	34
4.3 Validasi.....	51
4.3.1 Validasi Kebutuhan Fungsional .....	51
4.3.2 Validasi Kebutuhan <i>Non Fungsional</i> .....	52
4.4 Manajemen Kebutuhan .....	53
BAB 5 PERANCANGAN DAN IMPLEMENTASI .....	54
5.1 Perancangan .....	54
5.1.1 Perancangan Arsitektur.....	54
5.1.2 Perancangan Komponen .....	58
5.1.3 Perancangan Data .....	60
5.1.4 Perancangan Antarmuka.....	62
5.2 Implementasi .....	67
5.2.1 Spesifikasi Sistem .....	67



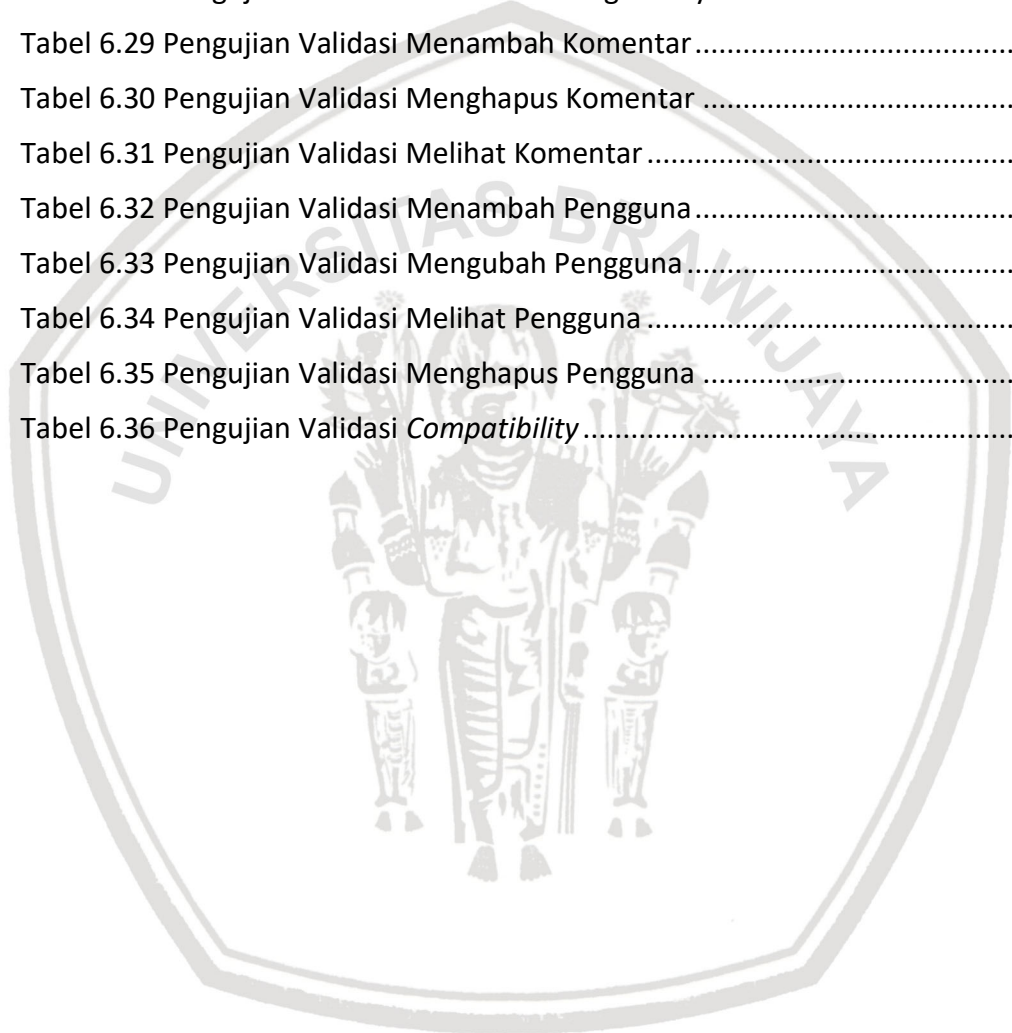
5.2.2 Batasan Implementasi.....	68
5.2.3 Implementasi Basis Data .....	69
5.2.4 Implementasi Komponen.....	69
5.2.5 Implementasi Antarmuka .....	72
<b>BAB 6 PENGUJIAN .....</b>	<b>76</b>
6.1 Pengujian Unit.....	76
6.1.1 Pengujian Unit Melihat Data Harian .....	76
6.1.2 Pengujian Unit Melihat Status Realisasi Anggaran .....	78
6.1.3 Pengujian Unit Menghitung Estimasi Biaya .....	82
6.2 Pengujian Integrasi .....	86
6.3 Pengujian Validasi.....	89
6.3.1 Kebutuhan Fungsional.....	89
6.3.2 Kebutuhan Non Fungsional.....	102
<b>BAB 7 PENUTUP .....</b>	<b>103</b>
7.1 Kesimpulan.....	103
7.2 Saran .....	104
<b>DAFTAR PUSTAKA.....</b>	<b>105</b>
<b>LAMPIRAN .....</b>	<b>107</b>
Lampiran A.....	107
Lampiran B .....	107
Lampiran C.....	108

## DAFTAR TABEL

Tabel 2.1 Elemen <i>Use Case</i> Diagram.....	15
Tabel 2.2 Komponen <i>Sequence</i> Diagram .....	16
Tabel 4.1 Identifikasi <i>Stakeholder</i> .....	25
Tabel 4.2 Identifikasi Aktor .....	28
Tabel 4.3 Kebutuhan Fungsional Sistem .....	29
Tabel 4.4 Spesifikasi Kebutuhan Non Fungsional Sistem.....	32
Tabel 4.5 <i>Use Case Scenario Login</i> .....	34
Tabel 4.6 <i>Use Case Scenario Logout</i> .....	34
Tabel 4.7 <i>Use Case Scenario</i> Menambah Data Produksi.....	35
Tabel 4.8 <i>Use Case Scenario</i> Mengubah Data Produksi.....	36
Tabel 4.9 <i>Use Case Scenario</i> Menghapus Data Produksi .....	36
Tabel 4.10 <i>Use Case Scenario</i> Menambah Data Harian .....	37
Tabel 4.11 <i>Use Case Scenario</i> Mengubah Data Harian .....	38
Tabel 4.12 <i>Use Case Scenario</i> Menghapus Data Harian .....	39
Tabel 4.13 <i>Use Case Scenario</i> Melihat Data Produksi.....	39
Tabel 4.14 <i>Use Case Scenario</i> Melihat Data Harian .....	40
Tabel 4.15 <i>Use Case Scenario</i> Mencetak Laporan Produksi .....	41
Tabel 4.16 <i>Use Case Scenario</i> Mencetak Laporan Harian.....	41
Tabel 4.17 <i>Use Case Scenario</i> Melihat Status Realisasi Produksi .....	42
Tabel 4.18 <i>Use Case Scenario</i> Total Produksi.....	43
Tabel 4.19 <i>Use Case Scenario</i> Menghitung Estimasi Biaya .....	43
Tabel 4.20 <i>Use Case Scenario</i> Melihat Status Biaya .....	44
Tabel 4.21 <i>Use Case Scenario</i> Menambah Target Produksi.....	45
Tabel 4.22 <i>Use Case Scenario</i> Menambah Target Biaya .....	45
Tabel 4.23 <i>Use Case Scenario</i> Menambah Komentar .....	46
Tabel 4.24 <i>Use Case Scenario</i> Mengubah Komentar .....	47
Tabel 4.25 <i>Use Case Scenario</i> Menghapus Komentar.....	48
Tabel 4.26 <i>Use Case Scenario</i> Melihat Komentar .....	49
Tabel 4.27 <i>Use Case Scenario</i> Menambah Pengguna .....	49

Tabel 4.28 <i>Use Case Scenario</i> Mengubah Pengguna .....	50
Tabel 4.29 <i>Use Case Scenario</i> Melihat Pengguna .....	50
Tabel 4.30 <i>Use Case Scenario</i> Menghapus Pengguna.....	51
Tabel 4.31 Validasi Kebutuhan Fungsional .....	52
Tabel 5.1 Algoritme Melihat Data Harian .....	58
Tabel 5.2 Algoritme Melihat Status Realisasi Produksi.....	58
Tabel 5.3 Algoritme Menghitung Estimasi Biaya .....	59
Tabel 5.4 Spesifikasi Perangkat Keras .....	68
Tabel 5.5 Implementasi Program Melihat Data Harian .....	70
Tabel 5.6 Implementasi Program Melihat Status Realisasi Anggaran .....	70
Tabel 5.7 Implementasi Program Menghitung Estimasi Biaya .....	71
Tabel 6.1 Algoritme Melihat Data Harian .....	76
Tabel 6.2 <i>Test Case</i> Melihat Data Harian .....	77
Tabel 6.3 Algoritme Melihat Status Realisasi Anggaran .....	78
Tabel 6.4 <i>Test Case</i> Melihat Status Realisasi Anggaran .....	81
Tabel 6.5 Algoritme Menghitung Estimasi Biaya .....	82
Tabel 6.6 <i>Test Case</i> Menghitung Estimasi Biaya .....	85
Tabel 6.7 Identifikasi dan Rencana Pengujian Integrasi .....	87
Tabel 6.8 Hasil Pengujian Integrasi Nomor 1 .....	87
Tabel 6.9 Hasil Pengujian Integrasi Nomor 2 .....	87
Tabel 6.10 Hasil Pengujian Integrasi Nomor 3 .....	88
Tabel 6.11 Pengujian Validasi <i>Login</i> .....	89
Tabel 6.12 Pengujian Validasi <i>Logout</i> .....	89
Tabel 6.13 Pengujian Validasi Menambah Data Produksi .....	90
Tabel 6.14 Pengujian Validasi Mengubah Data Produksi .....	90
Tabel 6.15 Pengujian Validasi Menghapus Data Produksi .....	91
Tabel 6.16 Pengujian Validasi Menambah Data Harian.....	91
Tabel 6.17 Pengujian Validasi Mengubah Data Harian .....	92
Tabel 6.18 Pengujian Validasi Menghapus Data Harian .....	93
Tabel 6.19 Pengujian Validasi Melihat Data Produksi.....	93
Tabel 6.20 Pengujian Validasi Melihat Data Harian .....	94
Tabel 6.21 Pengujian Validasi Mencetak Laporan Produksi .....	94

Tabel 6.22 Pengujian Validasi Mencetak Laporan Harian.....	95
Tabel 6.23 Pengujian Validasi Melihat Status Realisasi Produksi .....	95
Tabel 6.24 Pengujian Validasi Total Produksi .....	96
Tabel 6.25 Pengujian Validasi Menghitung Estimasi Biaya .....	96
Tabel 6.26 Pengujian Validasi Melihat Status Biaya .....	97
Tabel 6.27 Pengujian Validasi Menambah Target Produksi .....	97
Tabel 6.28 Pengujian Validasi Menambah Target Biaya .....	98
Tabel 6.29 Pengujian Validasi Menambah Komentar .....	98
Tabel 6.30 Pengujian Validasi Menghapus Komentar .....	99
Tabel 6.31 Pengujian Validasi Melihat Komentar .....	99
Tabel 6.32 Pengujian Validasi Menambah Pengguna .....	100
Tabel 6.33 Pengujian Validasi Mengubah Pengguna .....	100
Tabel 6.34 Pengujian Validasi Melihat Pengguna .....	101
Tabel 6.35 Pengujian Validasi Menghapus Pengguna .....	101
Tabel 6.36 Pengujian Validasi <i>Compatibility</i> .....	102



## DAFTAR GAMBAR

Gambar 2.1 <i>Column Chart</i> .....	8
Gambar 2.2 <i>Clustered Column Chart</i> .....	8
Gambar 2.3 <i>Stacked Column Chart</i> .....	9
Gambar 2.4 <i>Line Chart</i> .....	9
Gambar 2.5 <i>Pie Chart</i> .....	10
Gambar 2.6 <i>Gauge Meter</i> .....	10
Gambar 2.7 Model <i>Waterfall</i> (Pressman, 2010).....	18
Gambar 3.1 Diagram Blok Metodologi Penelitian .....	20
Gambar 4.1 <i>Use Case Diagram</i> .....	33
Gambar 5.1 <i>Sequence Diagram</i> Menambah Data Produksi .....	54
Gambar 5.2 <i>Sequence Diagram</i> Melihat Status Realisasi Produksi .....	55
Gambar 5.3 <i>Sequence Diagram</i> Menghitung Estimasi Biaya.....	56
Gambar 5.4 <i>Class Diagram</i> .....	57
Gambar 5.5 <i>Entity Relatonal Diagram</i> Dashboard Produksi PTPN XII Gunung Gambir.....	61
Gambar 5.6 <i>Physical Data Model</i> Dashboard Produksi PTPN XII Gunung Gambir.....	62
Gambar 5.7 Antarmuka Halaman Lihat Data Produksi .....	62
Gambar 5.8 Antarmuka Halaman Tambah Data Produksi.....	63
Gambar 5.9 Antarmuka Halaman Status Ralisasi Produksi.....	64
Gambar 5.10 Antarmuka Halaman Menghitung Estimasi Biaya.....	65
Gambar 5.11 Antarmuka Halaman Lihat Pengguna .....	66
Gambar 5.12 Antarmuka Halaman Home.....	67
Gambar 5.13 <i>Physical Data Model</i> Dashboard Produksi PTPN XII Gunung Gambir .....	69
Gambar 5.14 Implementasi Antarmuka Lihat Data Produksi .....	73
Gambar 5.15 Implementasi Antarmuka Tambah Data Harian .....	73
Gambar 5.16 Implementasi Antarmuka Status Realisasi Produksi.....	74
Gambar 5.17 Implementasi Antarmuka Menghitung Estimasi Biaya.....	74
Gambar 5.18 Implementasi Antarmuka Lihat Pengguna.....	75



Gambar 5.19 Implementasi Antarmuka Lihat Pengguna.....	75
Gambar 6.1 Flow Graph fungsi Melihat Data Harian.....	77
Gambar 6.2 Flow Graph fungsi Melihat Status Realisasi Anggaran.....	80
Gambar 6.3 Flow Graph fungsi Menghitung Estimasi Biaya.....	84
Gambar 7.1 Proses Bisnis Pembuatan Laporan Produksi.....	107
Gambar 7.2 Proses Bisnis Pembuatan Laporan Harian.....	107
Gambar 7.3 Proses Bisnis Permintaan Modal Kerja.....	108



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Teknologi informasi yang terus berkembang sangat cepat saat ini telah mempengaruhi kehidupan manusia dalam berbagai bidang. Salah satu dampak positif dari perkembangan teknologi informasi adalah penyebaran informasi. Dengan memanfaatkan jaringan komputer, penyebaran informasi dapat dilakukan dengan sangat mudah. Selain itu penyebaran informasi dapat dilakukan secara global dengan memanfaatkan jaringan komputer yang sangat besar, atau lebih dikenal dengan nama internet. Dengan menggunakan internet, penyebaran informasi yang dilakukan suatu perusahaan dengan perusahaan lainnya tidak terbatas jarak dan waktu (Pradipta, Prasetyo, dan Ambarsari, 2015), sehingga selain mudah dan cepat, dengan memanfaatkan internet dapat menekan biaya operasional yang harus dikeluarkan oleh suatu perusahaan. Hal ini tentunya juga akan berdampak terhadap keuntungan yang diperoleh suatu perusahaan.

Pada era saat ini sebuah informasi menjadi kebutuhan yang sangat diperlukan. Sebuah perusahaan perlu melakukan *monitoring* dan pengukuran secara terus menerus terhadap kinerjanya untuk dapat mengetahui ketercapaian tujuan yang sebelumnya telah ditargetkan. Proses *monitoring* kerja memerlukan data dan informasi dari seluruh bagian organisasi. Yang nantinya hasil monitoring ini akan disampaikan kepada pihak-pihak yang berkepentingan secara efektif dan efisien. Efektif disini nilai yang terkandung dalam informasi dapat diresapi dengan benar, dan efisien berarti informasi dapat dipahami dengan mudah dan cepat. Tak terkecuali perusahaan dalam sektor perkebunan. Penyebaran informasi secara efektif dan efisien diperlukan guna menunjang kualitas hasil produksi yang mana nantinya secara tidak langsung berimbas terhadap perekonomian Indonesia.

Industri pada sektor perkebunan merupakan salah satu kekuatan dan penopang perekonomian di Indonesia. Pada 2016 saja sektor ini memberikan kontribusi terhadap Produk Domestik Bruto (PDB) nasional mencapai Rp 429 triliun. Pendapatan sektor perkebunan ini melampaui sektor minyak dan gas (migas) yang nilainya hanya Rp. 365 triliun. Berdasarkan 127 komoditas perkebunan, hanya 15 komoditas yang menghasilkan devisa (Sanusi, 2017). Meski dalam kondisi yang belum baik sektor ini memberi andil terbesar terhadap perekonomian Indonesia.

PT Perkebunan Nusantara XII atau yang disebut PTPN XII adalah Badan Usaha Milik Negara (BUMN) yang bergerak pada sektor perkebunan. PTPN XII mengelola areal perkebunan seluas 80.000 ha tersebar luas diseluruh Jawa Timur yang terdiri dari 34 kebun dan terbagi menjadi menjadi tiga wilayah (ptpn, 2015).

Kebun Gunung Gambir merupakan salah satu perkebunan milik PTPN XII yang berada pada wilayah III. Gunung Gambir saat ini mengelola komoditas karet, teh dan aneka kayu. Pengiriman data informasi yang masih manual yaitu menggunakan *handy talki* (HT) serta pembuatan laporan yang masih menggunakan aplikasi *Microsoft Excel* sering menyebabkan masalah dalam proses

pengelolaan produksi, yang nantinya juga akan berdampak terhadap hasil produksi. Penyampain data hasil produksi yang disampaikan antar satu tempat ke tempat lain sering mengalami ketidaksesuaian data dikarenakan salah mendengar informasi dari pengirim informasi. Data dikirim ke kantor induk melalui HT dengan cara dibacakan secara manual dari laporan yang sebelumnya telah di rekap oleh juru tulis di dalam aplikasi *microsoft excel*. Kemudian penerima data yaitu kantor induk merekap ulang data yang diterima dari juru tulis ke dalam *microsoft excel*. Hal ini menyebabkan masalah berupa data yang disampaikan tidak praktis dan tidak akurat serta proses penyampaian data yang lambat. Perekapan laporan produksi dengan menggunakan aplikasi *microsoft excel* juga dirasa sulit untuk membaca laporan, menganalisa data, dan memprediksi data sebab informasi yang ditampilkan sebagian besar berupa angka.

Masalah tersebut dapat diatasi dengan pengelolaan data yang baik dan didukung dengan teknologi informasi yang baik pula. Diperlukan sebuah pengolahan data yang lebih menarik dan akurat sehingga nantinya dapat membantu proses dalam menganalisa data. Oleh sebab itu dibutuhkan sebuah sistem *dashboard* untuk *me-monitoring* pengolahan data informasi produksi. Dengan adanya sistem tersebut maka proses mengolah dan *monitoring* akan lebih mudah karena dapat dilakukan dimana saja dan kapan saja.

Berdasarkan penjelasan pada latar diatas, maka peneliti mengambil judul "Pembangunan *Dashboard* Produksi PTPN Persero Kantor Wilayah III Gunung Gambir Berbasis *Web*".

## 1.2 Rumusan masalah

Mengacu pada penjelasan latar belakang diatas, rumusan masalah yang dapat dikaji berupa

1. Bagaimana menganalisis kebutuhan dalam membangun sistem *dashboard* produksi berbasis *web* yang sesuai dengan kebutuhan PTPN XII Gunung Gambir?
2. Bagaimana perancangan dalam membangun sistem *dashboard* produksi berbasis *web* yang sesuai dengan kebutuhan PTPN XII Gunung Gambir?
3. Bagaimana menerapkan konsep *dashboard* produksi kedalam sistem *dashboard* untuk PTPN XII Gunung Gambir sesuai dengan model yang telah dibuat?
4. Bagaimana menguji sistem hasil implementasi sebelumnya dengan pengujian *White Box* dan *Black Box*?

## 1.3 Tujuan

Penelitian yang dilakukan oleh penulis memiliki tujuan yaitu :

1. Mendapatkan kebutuhan yang diperlukan untuk pembuatan sistem *dashboard* produksi PTPN XII Gunung Gambir.
2. Memodelkan sistem *dashboard* produksi yang sesuai dengan hasil analisis kebutuhan pada PTPN XII Gunung Gambir.

3. Membangun sistem *dashboard* produksi untuk PTPN XII Gunung Gambir yang sesuai dengan model yang telah dibuat.
4. Untuk mengetahui apakah sistem telah berjalan baik sesuai dengan pemodelan yang telah dibuat.

#### 1.4 Manfaat

Adapun manfaat dari penelitian ini adalah diharapkan dapat mempercepat serta mempermudah dalam memberikan informasi mengenai hasil produksi serta membantu dalam pengambilan keputusan dalam menentukan strategi kedepan di PTPN XII Gunung Gambir sehingga dapat meningkatkan kualitas maupun kuantitas hasil produksi PTPN XII Gunung Gambir.

#### 1.5 Batasan masalah

Upaya penelitian ini tidak menyimpang dari rumusan masalah, maka perlu adanya batasan didalam variabelnya. Oleh karea itu, penulis membatasi diri bahwa penelitian ini memonitoring hasil produksi pada PTPN XII Gunung Gambir afdeling karang anom, dan sistem yang dibangun berbasis *web*.

#### 1.6 Sistematika penulisan

##### 1. BAB I PENDAHULUAN

Menjelaskan tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian dan batasan masalah yang terkait dengan sistem *dashboard* produksi pada PTPN XII Gunung Gambir.

##### 2. BAB II TINJAUAN PUSTAKA

Teori dasar pendukung sebagai rujukan untuk mengerjakan penelitian yang merupakan inti dasar dari sebuah teori yang akan diterapkan.

##### 3. BAB III METODOLOGI

Metode dan langkah kerja yang dilakukan dalam pembuatan sistem *dashboard* produksi PTPN XII Gunung Gambir.

##### 4. BAB IV ANALISIS KEBUTUHAN

Menganalisi kebutuhan apa saja yang diperlukan untuk pembuatan sistem *dashboard* produksi PTPN XII Gunung Gambir.

##### 5. BAB V PERANCANGAN DAN IMPLEMENTASI

Perancangan sistem dashboard produksi PTPN XII Gunung Gambir sesuai dengan kebutuhan yang telah dianalisis. Implementasi menjelaskan tentang implementasi pada sistem. Implementasi sistem sesuai dengan perancangan yang telah dibuat

## 6. BAB VII PENGUJIAN

Bab ini berisikan pengujian terhadap sistem untuk menguji kelayakan sistem agar terhindar dari kesalahan.

## 7. BAB VIII PENUTUP

Bab ini akan diuraikan tentang kesimpulan dan saran.





## BAB 2 TINJAUAN PUSTAKA

### 2.1 Kajian Pustaka

Tinjauan utama dalam penelitian ini merujuk pada penelitian yang telah dilakukan sebelumnya oleh Oktavia (2012) yang berjudul Sistem Pengawasan Kinerja Unit Produksi Dalam Pemenuhan *Marketing Order* Di PT Sanbe Farma Bandung Berbasis *Web*. Dalam penelitian tersebut membangun sebuah sistem pengawasan kinerja unit produksi dalam pemenuhan marketing order berbasis web dan bersifat *online*. Berdasarkan proses analisis, desain dan implementasi dari perancangan perangkat lunak yang telah dibangun, penelitian tersebut menghasilkan kesimpulan yaitu: (1) Sistem Pengawasan Kinerja Unit Produksi dalam Pemenuhan *Marketing Order* Di PT Sanbe Farma Bandung Berbasis *Web* dapat membantu manajer untuk mencari referensi dalam pengambilan keputusan yang berkaitan dengan pemenuhan *marketing order*. (2) Sistem Pengawasan Kinerja Unit Produksi dalam Pemenuhan Marketing Order Di PT Sanbe Farma Bandung Berbasis Web memudahkan direktur dan manajer dalam melakukan pemantauan kinerja masing-masing unit produksi atas pemenuhan MO, hutang MO dan stok barang.

Ilhamsyah (2017) yang berjudul Perancangan Model Dashboard Untuk Monitoring Evaluasi Mahasiswa. Dalam penelitian tersebut dashboard digunakan untuk monitoring mahasiswa di bidang pendidikan di Jurusan Sistem Informasi FMIPA Universitas Tanjungpura Pontianak sebagai sarana untuk mengukur kualitas mahasiswa serta mengambil keputusan dalam menentukan strategi kedepan dalam pembinaan mahasiswa dilingkungan Jurusan Sistem Informasi FMIPA Untan Pontianak. Berdasarkan proses analisis, desain dan implementasi dari perancangan perangkat lunak yang telah dibangun, penelitian tersebut menghasilkan kesimpulan yaitu: (1) model dashboard monitoring mahasiswa menyediakan dashboard monitoring mahasiswa berbasis web yang dapat diakses pihak jurusan Sistem Informasi sebagai indikator kinerja monitoring mahasiswa. (2) Hasil analisis menggunakan model dashboard tektikal menunjukkan bahwa model tersebut dapat dijadikan acuan sebagai parameter dalam menetapkan KPI monitoring mahasiswa Jurusan Sistem Informasi.

### 2.2 Monitoring

Monitoring merupakan proses pengumpulan informasi secara terus menerus dan teratur yang nantinya dapat digunakan untuk menjawab permasalahan mengenai suatu proyek atau kegiatan tertentu (Oktavia, 2012). Monitoring dapat membantu memberi peringatan ketika nantinya terjadi sesuatu yang salah dan membantu agar pekerjaan tetap berada pada jalurnya. Monitoring bertujuan meningkatkan efektifitas dan efisiensi dari sebuah proyek atau organisasi, yang didasarkan pada tujuan dan sasaran rencana kegiatan yang telah ditetapkan. Monitoring membantu merupakan kegiatan yang sangat penting untuk dapat

mengetahui kualitas, efektifitas, efisiensi, capaian serta dampak yang diterima setelah menjalankan pekerjaan yang telah direncanakan dan dilaksanakan (Sunaryanto, et al., 2012).

Beberapa pakar manajemen mengemukakan bahwa fungsi monitoring memiliki nilai bobot yang sama pentingnya dengan fungsi perencanaan. Keberhasilan dalam mencapai tujuan separuhnya ditentukan oleh rencana yang telah diterapkan dan setengahnya lagi oleh pengawasan atau monitoring (Conor, 1974).

Tujuan monitoring yaitu (Oktavia, 2012) :

1. Mengkaji apakah kegiatan yang telah dilaksanakan sesuai dengan rencana yang telah ditetapkan.
2. Melakukan penilaian apakah pola kerja dan manajemen yang diterapkan sudah tepat untuk mencapai tujuan kegiatan.
3. Mengidentifikasi masalah yang timbul agar dapat langsung diatasi.
4. Menyesuaikan kegiatan dengan lingkungan yang berubah, tanpa menyimpang dari tujuan.
5. Mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan.

Aspek monitoring adalah sebagai berikut :

1. Aspek masukan (*input*) kegiatan, antara lain mencakup : tenaga manusia, dana, bahan, peralatan, jam kerja, data, kebijakan, manajemen yang dibutuhkan untuk melaksanakan kegiatan.
2. Aspek proses / aktivitas, yaitu aspek dari kegiatan yang mencerminkan suatu proses kegiatan seperti penelitian, pelatihan, proses produksi, dan pemberian bantuan.
3. Aspek keluaran (*output*), yaitu aspek kegiatan yang mencakup hasil proses dari proses, terutama proses yang berkaitan dengan kuantitas (jumlah).

Tipe monitoring terbagi menjadi dua, yaitu :

1. Monitoring rutin, yaitu kegiatan mengkompilasi informasi secara regular berdasarkan sejumlah indikator kunci. Monitoring rutin dapat dipergunakan untuk mengidentifikasi penerapan program dengan atau tanpa perencanaan.
2. Monitoring jangka pendek, yaitu monitoring yang dilakukan untuk jangka waktu tertentu dan biasanya diperuntukkan bagi aktifitas yang spesifik. Monitoring jangka pendek dapat dipergunakan untuk mengidentifikasi penerapan program dengan suatu perencanaan.

### 2.3 Dashboard

*Dashboard* merupakan alat untuk menyajikan informasi secara sekilas, solusi bagi kebutuhan informasi organisasi (Few, 2006). Dengan *dashboard* kita dapat

dengan cepat menentukan letak dari suatu permasalahan, sehingga dapat dengan segera menentukan langkah yang efektif untuk pengambilan keputusan. *Dashboard* bekerja dengan cara mengumpulkan informasi yang relevan dari semua bagian organisasi, mengkonsolidasikan, dan menyampaikannya secara aman, cepat, dengan personalisasi sesuai dengan peran pengguna dalam organisasi.

*Dashboard* adalah alat bantu kontrol untuk mengidentifikasi jalannya proses bisnis secara mudah dengan menggunakan indikator-indikator tertentu. Misalnya untuk pengguna menggunakan indikator warna-warna khusus yaitu merah, hijau atau kuning yang melambangkan status atau kondisi dari suatu bisnis atau proyek, tanda peringatan, ringkasan-ringkasan, grafik-grafik seperti *bar*, *chart*, *pie chart*, dimana biasanya dibuat satu set dalam suatu portal, dan dapat diatur.

Ada 3 macam tipe dari *dashboard*, menurut Irawan (2015) yaitu:

1. *Dashboard* strategis, merupakan dashborad yang digunakan untuk mendukung manajemen level strategis untuk memberikan informasi dalam mengambil keputusan bisnis, memprediksi peluang, serta memberikan arahan tujuan strategis.
2. *Dashboard* taktis, tipe ini berfokus terhadap proses analisis untuk menentukan penyebab dari kondisi atau keadaan tertentu.
3. *Dasboard* operasional, berfungsi untuk membantu monitoring segala aktifitas bisnis, berfokus pada monitoring aktifitas dan kejadian yang tidak berubah secara drastis.

### 2.3.1 Digital *Dashboard*

Digital *dashboard* adalah antarmuka yang mengumpulkan dan memvisualisasikan data dari berbagai sumber, seperti basis data, file yang dihosting secara lokal, dan layanan web (Rasmussen, 2009). Digital *dashboard* biasanya berupa *chart* atau grafik yang memiliki tampilan interaktif dan menarik. Digital *dashborad* banyak digunakan karena dalam penyampaianya mudah dipahami karena berbasis grafis dan warna warna yang menarik. Digital dashboard digunakan karena dapat memberikan warning atau deteksi peringatan dini jika terdapat informasi yang sangat penting atau berbahaya.

Berikut adalah contoh digital *dashboard* dan perbedaan penggunaanya (Alexander, 2014) :

#### 1. *Column Chart*

*Chart* ini digunakan untuk menunjukkan perbandingan data dalam waktu atau periode tertentu.



**Gambar 2.1 Column Chart**

(sumber : [https://cdn-images-1.medium.com/max/1320/1\\*5RgXTZlivCi7MwVgJ-04tg.png](https://cdn-images-1.medium.com/max/1320/1*5RgXTZlivCi7MwVgJ-04tg.png))

2. *Clustered Column Chart*

Chart ini digunakan untuk membandingkan berbagai jenis elemen atau data yang berada dalam satu waktu atau periode yang sama.



**Gambar 2.2 Clustered Column Chart**

(sumber : [https://cdn-images-1.medium.com/max/2000/1\\*j3ZfYtoH7oMPCAzEgJtnuQ.png](https://cdn-images-1.medium.com/max/2000/1*j3ZfYtoH7oMPCAzEgJtnuQ.png))

3. *Stacked Column Chart*

Chart ini mirip dengan *clustered column chart*, namun semua data yang berbeda disajikan menjadi satu kolom, sehingga kita bisa melihat total semua data. Namun diberikan pembeda antar data satu dengan data lain, biasanya dengan menggunakan pembatas berupa garis atau warna.



**Gambar 2.3 Stacked Column Chart**

(sumber : [https://cdn-images-1.medium.com/max/1320/1\\*j3ZfYtoH7oMPCAzEgJtnuQ.png](https://cdn-images-1.medium.com/max/1320/1*j3ZfYtoH7oMPCAzEgJtnuQ.png))

4. *Line Chart*

*Chart* ini biasanya digunakan untuk menampilkan data yang berkelanjutan. *Chart* ini sangat cocok untuk mengidentifikasi suatu pola yang terjadi di dalam sebuah data.



**Gambar 2.4 Line Chart**

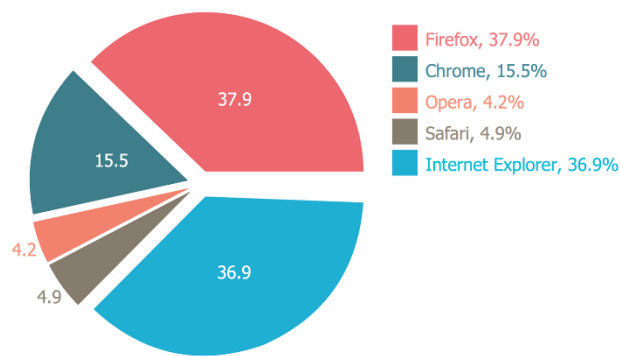
(sumber : [https://cdn-images-1.medium.com/max/2000/1\\*orDzqfwjqdDXb7YwOLxH0w.png](https://cdn-images-1.medium.com/max/2000/1*orDzqfwjqdDXb7YwOLxH0w.png))

5. *Pie Chart*

*Chart* ini digunakan ketika terdapat lebih dari satu data yang ingin dibandingkan. Biasanya data yang dibandingkan ditampilkan dalam bentuk presentase. *Chart* ini sangat efektif ketika membandingkan lebih dari 3 buah data yang berbeda.







**Gambar 2.5 Pie Chart**

(sumber : <https://1.bp.blogspot.com/-7GD5rHR593w/WX3yjtaC79I/AAAAAAAAAD4w/OhKZNhe9qtsfUFjhUnL8ac2SI9pAYVaEgCLcBGAs/s1600/cara%2Bmembuat%2Bpie%2Bchart%2Bdi%2Bexcel.png>)

**6. Gauge Meter**

*Gauge meter* atau biasa disebut diagram *speedometer* biasanya digunakan untuk menampilkan sebuah ketercapaian dalam suatu *progres*. Umumnya *gauge meter* dalam memberi informasi yang berbeda menggunakan pembeda warna. Misalnya warna merah untuk kondisi yang masuk area berbahaya, warna kuning untuk kondisi yang masuk area sedang atau diambang batas antara aman dan bahaya, dan warna hijau untuk kondisi sudah masuk area aman.



**Gambar 2.6 Gauge Meter**

(sumber : <http://sis.binus.ac.id/files/2017/01/Capturez.jpg>)

**2.3.2 Portal Perusahaan**

Portal perusahaan merupakan platform yang sangat populer untuk menerapkan dashboard secara masal. Portal perusahaan bisa juga disebut informasi perusahaan (*Enterprise Informan Portal / EIP*). Shilakes dan Tyleman, Meeril Lynch Inc mendefinisikan EIP sebagai aplikasi yang memungkinkan perusahaan untuk memperoleh informasi yang disimpan secara internal, serta menyediakan pengguna gateway tunggal untuk membuat informasi yang sesuai dengan kebutuhan masing-masing dan pengetahuan untuk membuat pemberitahuan keputusan bisnis (Collins, 2003).



Manfaat diterapkannya *dashboard* yang mendukung portal terhadap organisasi adalah sebagai berikut:

1. Memiliki kokasi tunggal dalam mengakses *dashboard* serta dokumen, meteri presentasi dan diskusi *online*.
2. Memungkinkan penggunaan password tunggal (lebih sedikit pengguna, lebih mudah untuk masuk ke berbagai aplikasi).
3. Meningkatkan efisiensi sebab pengguna hanya perlu mengakses satu sumber saja untuk mendapatkan informasi yang beragam.
4. Menetapkan titik pusat bagi organisasi untuk menjalankan beberapa, atau bahkan seluruh, aplikasi intelijen bisnis (IB).

Tidak semua aplikasi intelijen bisnis (IB) mendukung penerapan portal perusahaan, namun semua aplikasi yang berbasis *web* dapat diakses melalui *hyperlinks*, dan *link* tersebut dapat diakses hampir semua bagian yang relevan dari portal sebuah *dashboard*. Hal ini membantu dalam menyediakan akses terhadap informasi yang sesuai dengan dashboard. Misalnya *hyperlink* ke detail laporan *finance* dapat ditempatkan di sisi grafik *finance* yang menunjukkan angka faktual dan anggaran dari seluruh laporan laba rugi.

### 2.3.3 Karakteristik *Dashboard*

Menurut Malik (2005), karakteristik *dashboard* dibagi sebagai berikut :

1. *Synergetic*  
Memiliki tampilan visual yang simple dan mudah dipahami oleh pengguna, mensinergikan informasi dari beragam aspek yang berbeda dalam satu layar.
2. *Monitor*  
Menampilkan key performance indicator (KPI) yang dibutuhkan dalam pengambilan keputusan dalam keadaan tertentu, sesuai dengan tujuan yang telah ditetapkan.
3. *Accurate*  
Informasi yang ditampilkan harus akurat, sesuai dengan keadaan yang sebenarnya untuk mendapatkan kepercayaan pengguna.
4. *Responsive*  
Merespon masukan yang diterima dengan memberikan alert (seperti *alarm*, *blinker*, *email*) untuk memperingatkan pengguna jika terjadi hal yang kritis.
5. *Timely*  
Menampilkan informasi terkini yang diperlukan untuk pengambilan keputusan.
6. *Interactive*  
Menampilkan informasi yang sesuai dengan yang diinginkan oleh pengguna, dapat melakukan analisis sebab akibat dan sebagainya.

7. *More data history*  
Dapat menampilkan perbandingan data yang telah tersimpan sebelumnya untuk dapat mengetahui apakah kondisi sekarang lebih baik atau tidak
8. *Personalized*  
Dapat menyajikan informasi yang spesifik terkait jenis pengguna, hak akses dan batasan data.
9. *Analytical*  
Dapat melakukan analisis terkait masalah tertentu sehingga dapat menjadikan sebuah informasi untuk mengambil keputusan.
10. *Collaborative*  
Fasilitas pertukaran data antar pengguna sebagai sarana komunikasi dalam melakukan manajemen dan kontrol.
11. *Trackability*  
Memungkinkan setiap pengguna dapat menentukan nilai yang akan dilakukannya.

#### **2.3.4 Tujuan Penggunaan *Dashboard***

Menurut Malik (2005), tujuan dalam penggunaan dashboard adalah sebagai berikut :

1. Mengkomunikasikan strategi  
Dapat dengan mudah mendiskusikan strategi dan tujuan yang dibuat oleh manager, kepada semua pihak yang berkepentingan, sesuai dengan level yang telah ditentukan.
2. Memonitoring dan menyesuaikan pelaksanaan strategi  
Mengawasi pelaksanaan dari rencana dan strategi yang telah dibuat agar sejalan dengan tujuan yang telah ditentukan.
3. Menyampaikan dan menyesuaikan pelaksanaan strategi  
Menyajikan informasi dengan menampilkan grafik, bagan, simbol, dan warna yang menarik sehingga memudahkan pengguna dalam memahami dan merepresentasikan informasi secara benar.

#### **2.4 Konsep Dasar Website**

*World wide web* (web) di zaman sekarang telah menjadi pilihan masyarakat dalam mendistribusikan informasi baik dalam lingkup individu seperti penggunaan blog, organisasi seperti penggunaan *web* untuk organisasi tertentu dan juga digunakan untuk keperluan komunitas, tim, lembaga pemerintahan, perusahaan dan masih banyak lagi.

Aplikasi *web* atau lebih dikenal dengan istilah *WebApps* merupakan salah satu domain dari tujuh domain perangkat lunak komputer dan didefinisikan sebagai

satu atau kumpulan beberapa file hypertext yang saling terhubung satu sama lain dimana file hypertext tersebut merepresentasikan informasi menggunakan teks, grafis dan lain sebagainya (Yuhefizar, 2008).

Jika ditinjau dari definisi web dan *website*, kedua ini merupakan hal yang berbeda. Web merupakan kumpulan file hypertext sedangkan website merupakan kumpulan dari seluruh web.

Berdasarkan konten atau isinya, web dapat dibagi menjadi 2 jenis yaitu sebagai berikut (Hidayat, 2010):

1. Web Statis didefinisikan sebagai web yang memiliki isi yang tidak berubah-ubah. Yang dimaksud dengan tidak berubah-ubah adalah halaman web tersebut tidak dapat diubah secara langsung. Contoh web seperti ini adalah web profil perusahaan, lembaga, komunitas dan lain sebagainya. Web profil pada umumnya memuat informasi yang tidak perlu diubah setiap hari sehingga web profil perusahaan tergolong web statis.
2. Web Dinamis merupakan web yang isi dapat berubah-ubah setiap saat. Untuk melakukan perubahan tersebut pengguna cukup mengubahnya langsung secara online. Contoh web seperti ini adalah web yang berisi daftar hadir anggota perusahaan dan web media sosial karena konten web bisa berubah kapan saja.

## 2.5 PHP

PHP mulai dikembangkan melalui penelitian yang dilakukan oleh Rasmus Lerdorf pada tahun 1994 untuk menyelesaikan resume nya. Awalnya PHP digunakan untuk mencatat pengunjung *homepage*-nya. Pada saat itu Rasmus membuat PHP 2.0. Kemudian penelitian dilanjutkan oleh Zeev Suraski dan Andi Gutmans pada tahun 1997 yang menghasilkan PHP 3.0 (Aziz, 2017). PHP dikenal sebagai bahasa pemrograman open source yang berbasis web, sekarang pengembangan PHP sudah sampai pada versi 7.

Menurut Anhar (2010), PHP merupakan akronim dari PHP *Hypertext Processor*. PHP adalah bahasa pemrograman web *server-side* yang bersifat *open source*. File yang berisi sintaks bahasa pemrograman ini memiliki ekstensi `[dot]php`. PHP mendukung kerja web dari aspek pemrograman termasuk operasi aritmatik dan implementasi dari algoritme tertentu. Penggunaan bahasa pemrograman PHP memungkinkan pengguna untuk membuat *variabel*, *array*, percabangan, perulangan dan berbagai macam fungsi yang sama seperti bahasa pemrograman pada umumnya. Semua scrip php dieksekusi di dalam sebuah server dimana *script* tersebut dijalankan. PHP juga mendukung pembuatan sistem berbasis web yang berorientasi objek maupun pembuatan sistem yang bersifat struktural.



## 2.6 Pendekatan *Object Oriented*

### 2.6.1 Pengertian Pendekatan *Object Oriented*

Pendekatan dengan berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Pengertian “berorientasi objek” berarti bahwa kita mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Object Oriented Analisis mempelajari permasalahan dengan menspesifikasikannya atau mengobservasi permasalahan tersebut dengan menggunakan metode berorientasi objek. Biasanya analisa sistem dimulai dengan adanya dokumen permintaan (requirement) yang diperoleh dari semua pihak yang berkepentingan. Hasil analisis berorientasi objek adalah deskripsi dari apa sistem secara fungsional diperlukan untuk melakukan, dalam bentuk sebuah model konseptual. Itu biasanya akan disajikan sebagai seperangkat menggunakan kasus, satu atau lebih UML diagram kelas, dan sejumlah diagram interaksi. Tujuan dari analisis berorientasi objek adalah untuk mengembangkan model yang menggambarkan perangkat lunak komputer karena bekerja untuk memenuhi seperangkat persyaratan yang ditentukan pelanggan (Hidayat, 2010).

### 2.6.2 UML (*Unified Modeling Language*)




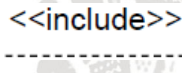

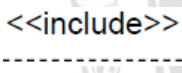
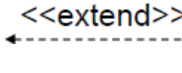

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik / gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software.

Menurut Nugroho (2009), dalam melakukan pemodelan menggunakan pendekatan *object oriented*, ada beberapa jenis diagram UML, yaitu :

1. *Use Case Diagram*, yaitu diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng- create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Komponen yang terdapat pada *use case diagram* adalah sebagai berikut (Nugroho, 2010) :



Tabel 2.1 Elemen *Use Case Diagram*

Nama Komponen	Simbol	Pengertian
Actor		Mempresentasikan atau menggambarkan seseorang atau sesuatu (seperti mesin, perangkat lain) yang berinteraksi dengan sistem atau melakukan pekerjaan lain. <i>Actor</i> hanya berinteraksi dengan use case namun tidak memiliki kontrol terhadap use case. Secara umum actor diklasifikasikan menjadi 3 jenis, yaitu pengguna sistem, sistem lain yang berhubungan dengan sistem yang akan dibangun, dan waktu.
Use case		Merupakan gambaran fungsional dari sistem. <i>Use case</i> menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem, sehingga pengguna sistem akan paham dan mengerti mengenai kegunaan dari sistem.
Asosiasi		Sebagai penghubung antara <i>actor</i> dan <i>use case</i> .
Include		Memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas lain yang disediakan oleh <i>use case</i> lain.
Asosiasi		Sebagai penghubung antara <i>actor</i> dan <i>use case</i> .
Include		Memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas lain yang disediakan oleh <i>use case</i> lain.
Extend		Menspesifikasikan bahwa <i>use case</i> tertentu memperluas perilaku dari <i>use case</i> pada suatu titik yang diberikan.
Generalisasi		Hubungan dimana objek anak berbagi perilaku dan struktur data dari objek yang ada di atasnya.

2. *Use Case Scenario*, adalah use case yang menjelaskan secara detail aliran event yang terjadi pada setiap use case (Nugroho, 2010). Aliran ini menceritakan keseluruhan kejadian pada setiap use case, mulai dari awal hingga akhir secara sistematis dan urut. Use case ini menceritakan apa yang akan sistem kerjakan kedalam aliran event. Aliran event ini bertujuan untuk mendokumentasikan aliran-aliran logika dalam setiap use case.




3. *Class Diagram*, *class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek (Nugroho, 2010). *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* dapat merupakan implementasi dari sebuah interface, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian interface mendukung resolusi metoda pada saat run-time.





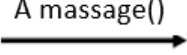
Struktur class secara umum terdiri dari 2 bagian (Nugroho, 2010) :

1. *Atrribut / field / properti*  
Menggambarkan tampilan, status, kualitas objek dan berlaku sebagai data, didefinisiakn oleh *class*, individu berbeda satu dengan lainnya. Didefinisikan dalam *class* menggunakan variable.
2. *Perilaku / behaviour / operations / method*  
Menggambarkan bagaimana sebuah intance dari class beroperasi misal bagaimana reaksi dari class jika diminta untuk melakukan sesuau. Didefinisikan dengan menggunakan fungsi
4. *Sequence Diagram*, yaitu diagram yang menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu (Nugroho, 2010). *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men- trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Komponen yang terdapat pada *sequence diagram* adalah sebagai berikut (Nugroho, 2010) :

**Tabel 2.2 Komponen *Sequence Diagram***

Nama Komponen	Simbol	Pengertian
Actor		Mempresentasikan atau menggambarkan seseorang atau sesuatu (seperti mesin, perangkat lain) yang berinteraksi dengan sistem atau melakukan pekerjaan lain.

Tabel 2.2 Komponen *Sequence Diagram*

Nama Komponen	Simbol	Pengertian
Boundary		Mewakili semua antarmuka dari sistem.
Controler		Mewakili semua objek di dalam sistem yang memiliki tugas sebagai pengaturproses logika dari sistem
Entity		Mewakili semua objek di dalam sistem yang merepresentasikan konsep bisnis dan penyimpanan data.
Life Line		Menggambarkan tempat mulai dan berakhirnya sebuah proses.
Line Message		Menggambarkan pengiriman pesan.

## 2.7 Pengujian

### 2.7.1 White Box Testing

*White-box testing* merupakan sebuah langkah dalam pengujian dimana yang diuji adalah bagian dalamnya koding program yang telah dibuat dengan mengikuti scenario atau test-case yang telah dibuat sebelumnya (Al fatta, 2007).

Menurut Black (2009), Structural test (atau yang biasa disebut *white-box tests* dan *glass-box tests*) menemukan *bug* dalam elemen struktur tingkat rendah seperti yang terjadi di tingkatan *code*, database *scemas*, *chips*, *subassemblies* dan *interfaces*. Pengujian struktural ini didasarkan pada bagaimana suatu sistem beroperasi.

Pengujian struktural melibatkan pengetahuan teknis terperinci dari sistem. Untuk menguji *software*, tester membuat pengujian yang paling struktural dengan melihat kode dan struktur data itu sendiri. Untuk pengujian *hardware*, tester membuat pengujian struktural untuk membandingkan spesifikasi *chip* untuk pembacaan *oscilloscopes* atau meter tegangan.

### 2.7.2 Black Box Testing

*Black-box testing* merupakan langkah dalam pengujian dimana yang diuji adalah bagian luar program seperti antarmuka aplikasi, kecepatan akses, ketersediaan (Al fatta, 2007).

Menurut Shalahuddin dan Rosa (2011) pengujian ini tidak menguji *source code* program. Data di uji dibangkitkan, dieksekusi pada perangkat lunak

kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.

Pengujian black box berfokus pada pengujian persyaratan fungsional perangkat lunak, untuk mendapatkan serangkaian kondisi input yang sesuai dengan persyaratan fungsional suatu program. Pengujian black box berusaha menemukan kesalahan dalam kategori (Al fatta, 2007) :

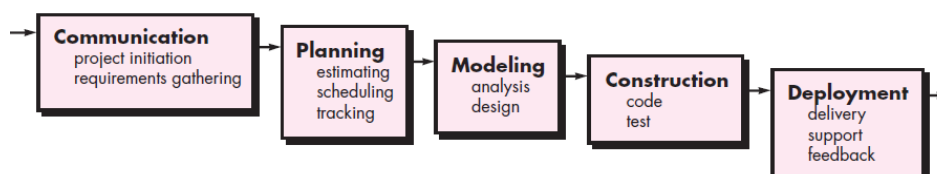
- Fungsi-fungsi yang tidak benar atau hilang dalam aplikasi.
- Kesalahan interface.
- Kesalahan dalam struktur data.
- Kesalahan kerja inisialisasi dan kesalahan terminasi.

## 2.8 System Development Life Cycle

*System development life cycle* (SDLC) yaitu siklus hidup pengembangan sistem yang dalam rekayasa perangkat lunak merupakan proses pembuatan dan pengembangan sistem serta model dan metodologi yang digunakan untuk mengembangkan sebuah sistem (Susanto, 2005). SDLC meliputi proses rencana (*planing*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) serta pengelolaan (*maintenance*). Konsep SDLC mendasari berbagai jenis metodologi pengembangan dalam rekayasa perangkat lunak. Metode-metode ini nantinya akan membentuk suatu kerangka kerja dalam perencanaan dan pengendalian dalam pengembangan perangkat lunak. Terdapat tiga jenis metodologi dalam siklus hidup pengembangan perangkat lunak, yaitu siklus hidup sistem tradisional (*traditional system life cycle*), siklus hidup menggunakan prototyping (*life cycle using prototyping*), dan siklus hidup sistem orientasi objek (*object-oriented system life cycle*). Ada berapa jenis model pengembangan perangkat lunak, salah satunya adalah model pengembangan perangkat lunak *waterfall*.

### 2.8.1 Waterfall

Model *waterfall* merupakan model pengembangan perangkat lunak yang sangat sering digunakan. Model pengembangan ini memiliki sifat linear mulai dari tahap awal pengembangan sistem yaitu perancangan hingga sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Proses perancangan dimulai dengan terlebih dahulu merencanakan semua proses kegiatan dan membuat jadwal sebelum mengerjakan (Pressman, 2010). Tahapan dari pengembangan pada model *waterfall* dapat dilihat dari gambar 2.1 berikut.



Gambar 2.7 Model *Waterfall* (Pressman, 2010)



Menurut Pressman (2017) model *waterfall* memiliki lima tahapan yaitu :

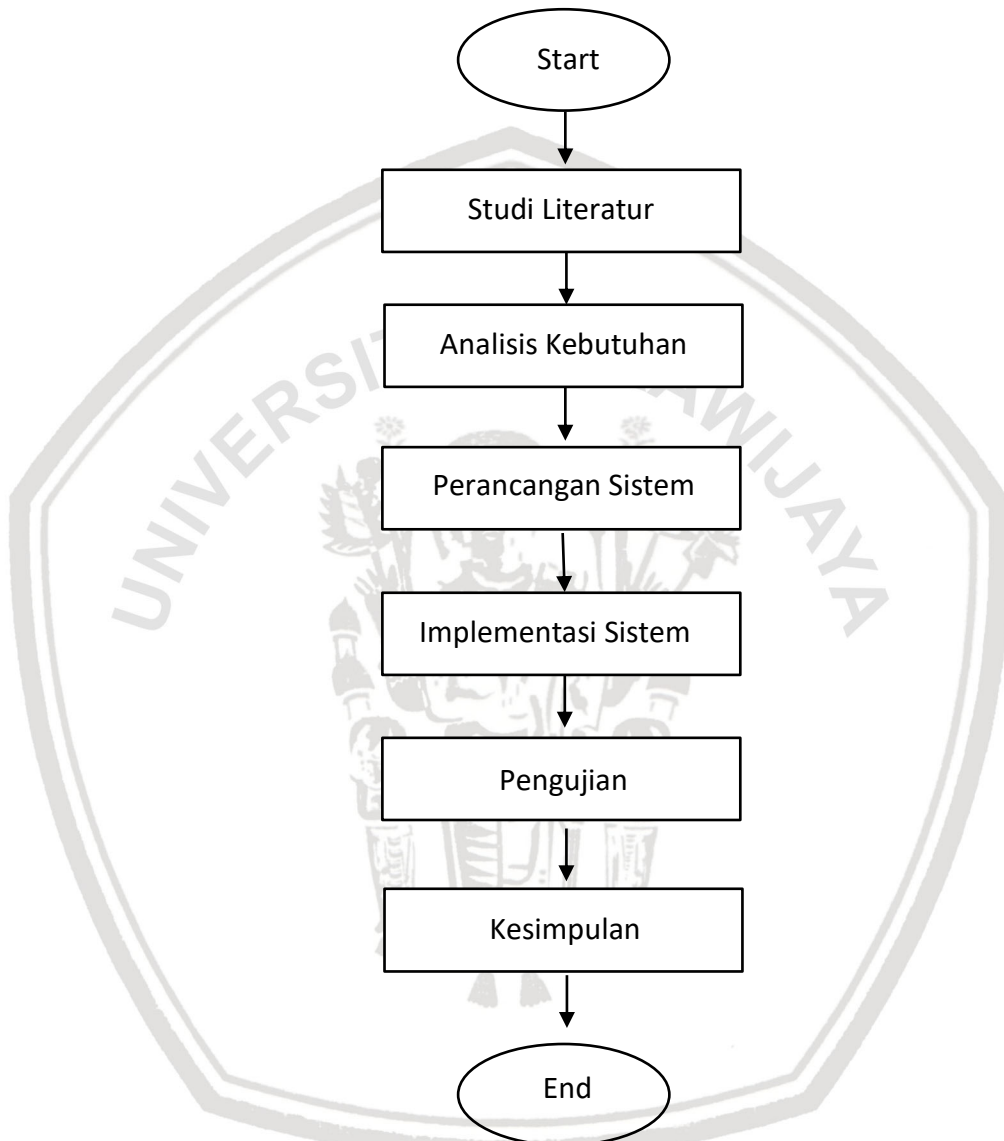
1. *Comunication*  
Pada tahap ini dilakukan analisis masalah dan tujuan yang akan dicapai. Kemudian dilakukan requirements gathering yang mana dilakukan pengumpulan kebutuhan dari pengguna.
2. *Planning*  
Tahap ini merupakan tahap perencanaan mengenai kebutuhan apa saja yang diperlukan dalam pembangunan sebuah sistem. Selain itu penjadwalan proses pengerjaan juga ditentukan dalam tahapan ini.
3. *Modeling*  
Pada tahap ini dilakukan perancangan kebutuhan sistem kedalam representasi untuk mengetahui kualitas tahapan selanjutnya yang akan dikerjakan. Tahap ini lebih memfokuskan struktur data, arsitektur perangkat lunak dan detail prosedur.
4. *Construction*  
Dalam tahap ini dilakukan pembuatan kode. *Coding* atau pengkodean merupakan proses implementasi dari hasil pemodelan yang telah dibuat sebelumnya. Setelah pengkodean selesai dibuat akan dilakukan *testing* terhadap sistem yang telah dikerjakan untuk mengetahui kesalahan-kesalahan terhadap sistem tersebut yang kemudian akan diperbaiki.
5. *Development*  
Merupakan tahap akhir dari pembuatan *software*. Setelah melakukan proses analisis, desain dan pengkodean maka sistem telah siap digunakan oleh pengguna. Lalu *software* yang telah dibuat nantinya harus dilakukan proses pemeliharaan secara berkala.

Pengembang perangkat lunak dituntut untuk menjaga siklus tersebut tetap terlaksana tanpa terjadinya *overlap* antara satu siklus dengan siklus lainnya sehingga proses pengembangan perangkat lunak dapat berjalan efisien.



## BAB 3 METODOLOGI

Pada bab ini membahas metode-metode yang menjadi acuan dalam pembangunan sistem *dashboard* produksi untuk PTPN XII Gunung Gambir. Tahapan metodologi penelitian memiliki struktur yang dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Blok Metodologi Penelitian

### 3.1 Studi Literatur

Literature yang digunakan dalam merancang system diperoleh melalui berbagai sumber informasi seperti *e-book*, berita *online*, jurnal, maupun informasi yang ada di internet. Hal-hal yang dilakukan untuk mendukung penelitian ini, diantaranya adalah pemahaman tentang :

1. *Dashboard*
2. *Monitoring*
3. *Webside*
4. *Object Oriented Programming*
5. *Unified Modeling language (UML)*
  - a. *Use case diagram*
  - b. *Sequence diagram*
  - c. *Class diagram*
  - d. *Use case scenario*

### 3.2 Analisis Kebutuhan

Kebutuhan-kebutuhan dalam membangun sistem dijabarkan pada tahapan analisis kebutuhan ini. Hal-hal yang membantu proses pembangunan sistem *dashboard* produksi untuk PTPN XII Gunung Gambir. Tahapan ini dilakukan untuk mengetahui hal apa saja yang dibutuhkan untuk membangun sistem. Mulai dari baik dari segi software, kebutuhan data, kebutuhan fungsional hingga kebutuhan *non-fungsional*. Proses analisis pembangunan dashboard produksi pada PTPN XII persero kantor wilayah III gunung gambir berbasis *website* adalah sebagai berikut:

#### 1. Proses Elisitasi Kebutuhan

Proses elisitasi kebutuhan dalam pembangunan dashboard produksi pada PTPN XII Gunung Gambir berbasis *website* dilakukan berdasarkan analisis terhadap hasil wawancara dan observasi terhadap karyawan yaitu juru tulis serta pimpinan perusahaan PTPN XII kebun Gunung Gambir. Metode wawancara dilakukan kepada juru tulis dan wakil kepala kantor untuk mengetahui secara jelas proses produksi perusahaan. Dikarenakan letak perusahaan yang mudah dijangkau oleh penulis, observasi dipilih guna mengetahui permasalahan yang terjadi untuk melengkapi data-data yang diperlukan penulis.

#### 2. Proses Spesifikasi Kebutuhan

Proses spesifikasi kebutuhan mendefinisikan kebutuhan dan menspesifikasi hasil dari proses elisitasi yang dikategorikan menjadi beberapa kebutuhan seperti kebutuhan fungsional dan kebutuhan *non-fungsional* agar dapat mengetahui aktifitas apa saja yang dapat dikerjakan oleh sistem serta siapa saja yang terlibat dengan mengidentifikasi aktor beserta batasan-batasannya. Selain itu dilakukan pemodelan kebutuhan dalam bentuk *use case diagram* serta *use case scenario*.

### 3. Proses Validasi Kebutuhan

Pada proses ini dilakukan validasi atau pengecekan kesesuaian kebutuhan yang telah didefinisikan terhadap keinginan pengguna yang telah disepakati sebelumnya.

### 4. Proses Manajemen Kebutuhan

Dalam proses manajemen kebutuhan dilakukan proses kontrol dengan memberikan nama kode pada masing-masing kebutuhan yang telah dispesifikasi sebelumnya.

## 3.3 Perancangan dan Implementasi

### 3.3.1 Perancangan Sistem

Perancangan sistem merupakan tahap yang dilakukan untuk merancang sistem sesuai dengan hasil dari analisis kebutuhan fungsional sistem. Perancangan dilakukan untuk merencanakan gambaran awal sistem agar nantinya sistem yang dibuat tidak melenceng atau melebar dari rencana awal.

Pada penelitian ini, pengerjaan website yang dilakukan menggunakan perancangan dengan menggunakan pendekatan object-oriented, dimana dalam pengerjaannya akan dibuat sebagai berikut:

#### 1. Perancangan Arsitektur

Dalam perancangan arsitektur ini akan dibuat pemodelan *sequence* diagram berdasarkan hasil pemodelan *use case scenario* yang telah disdefiniskan sebelumnya. Pada bagian ini akan dijelaskan tiga sampel *sequence* diagram yang merupakan tiga fitur utama. Pada tahap ini akan melakukan pengelompokan kelas didalam pemodelan *class* diagram termasuk juga atribut kelas, pengoperasian serta hubungan antar kelas lain.

#### 2. Perancangan Komponen

Dalam perancangan komponen akan dibuat 3 algoritme utama yang terdapat dalam sistem yang akan dibangun. Algoritme tersebut nantinya kan dipaparkan dalam bentuk *pseudocode*.

#### 3. Perancangan Data

Selanjutnya akan dibuat perancangan data yang nantinya terdapat tabel basis data yang akan dibuat dalam sistem dalam bentuk *entity relationship diagram* (ERD) dan *Physical Data Model* (PDM) yang nantinya akan menjadi acuan dalam pembangunan basis data yang akan digunakan didalam sistem yang akan dibuat.

#### 4. Perancangan Antarmuka

Kemudian dibuat rancangan antarmuka yang merepresentasikan rancangan yang telah dibuat sebelumnya yang nantinya akan menjadi acuan dalam implementasi antarmuka sistem dalam bentuk tata letak komponen yang ada dalam sistem.

### 3.3.2 Implementasi Sistem

Implementasi merupakan tahap yang dilakukan untuk merealisasikan sistem yang akan dibangun berdasarkan pada tahap perancangan sistem beserta kebutuhan-kebutuhan yang telah diperoleh dari user sehingga nantinya seluruh hasil dari implementasi tidak melenceng dari perancangan yang telah dibuat dan dapat ditelusuri. Proses yang berada pada tahap implementasi adalah sebagai berikut :

#### 1. Implementasi Basis Data

Pada tahapan implementasi basis data, penerapan dilakukan dengan mengacu pada tahap perancangan basis data yang telah dibuat pada tahap perancangan sistem. Basis data yang diterapkan menggunakan basis data *Database Management System* (DBMS) MySQL. Untuk pengelolaannya digunakan aplikasi web phpMyAdmin untuk memudahkan dalam memanipulasi tabel atau basis data yang akan dibuat.

#### 2. Implementasi Komponen

Dalam implemen komponen dilakukan implementasi logika program yang akan dibangun dengan mengacu pada perancangan komponen dengan memperhatikan algoritme-algoritme utama yang berada didalam sistem. Logika tersebut diterapkan menggunakan bahasa PHP dengan menggunakan *framework codeigniter* (CI).

#### 3. Implementasi Antarmuka

Implementasi antar muka sistem mengacu kepada perancangan antar muka yang telah dirancang pada tahap perancangan. Dalam implementasi antarmuka menggunakan bahasa pemrograman HTML *javascript*, CSS dan *bootstrap*.

### 3.4 Pengujian

Pengujian merupakan tahap pengujian sistem dengan menjalankan sistem untuk melihat bug, kesalahan, ataupun kekurangan yang ada di dalam sistem yang dibangun. Hal ini harus dilakukan untuk meminimalisir kesalahan sebisa mungkin sebelum nantinya sistem akan digunakan oleh penggunanya. Dalam tahap pengujian ini terdapat dua langkah yang harus dilakukan yaitu *white-box testing* dan *black-box testing*. Berikut adalah penjelasan dari pengujian yang akan dilakukan :

#### 1. Pengujian Unit

Merupakan pengujian yang dilakukan terhadap setiap operasi didalam tiap kelas. Dalam pengujian ini menggunakan *basis path testing*, dimana merupakan salah satu jenis pengujian *white-box*. Pengujian ini dilakukan terhadap tiga operasi saja yang akan dijadikan sampel uji.

## 2. Pengujian Integrasi

Pengujian ini melibatkan kelas-kelas yang terdapat dalam sistem, dimana pengujian ini berfokus kepada kelas yang memiliki hubungan dengan kelas lain. Sama seperti pengujian unit, pengujian integrasi juga menggunakan metode *basis path testing*. Hanya tiga operasi pada satu kelas yang berhubungan dengan kelas lain yang akan diuji untuk dijadikan sampel.

## 3. Pengujian Validasi

Pengujian sistem dilakukan untuk mengetahui apakah sistem telah berjalan sesuai dengan kebutuhan yang telah ditentukan pada tahap analisis, baik kebutuhan fungsional maupun kebutuhan non fungsional. Dalam pengujian ini menggunakan metode pengujian *black-box*.

## 3.5 Kesimpulan

Penarikan kesimpulan dilakukan setelah semua tahapan selesai mulai dari tahap analisis hingga tahap pengujian sistem. Berdasarkan penjabaran mengenai latar belakang masalah sistem *dashboard*, landasan pustaka yang membahas metode yang akan di gunakan, juga metodologi penelitian, penulis akan dapat mengambil kesimpulan yang nantinya akan menjawab rumusan masalah yang sebelumnya telah dirumuskan.

Selain penarikan kesimpulan, nantinya juga akan dilakukan evaluasi. Evaluasi dilakukan untuk mengetahui keasalahan maupun kekurangan yang terjadi, sehingga nantinya jika ada penelitian selanjutnya yang dilakukan untuk mengembangkan ataupun melakukan perbaikan harapannya akan menghasilkan sistem yang lebih baik dari segi bahasa pemrograman, rancangan sistem, maupun tampilan *web*.



## BAB 4 ANALISIS KEBUTUHAN

Analisis kebutuhan merupakan tahap awal yang dilakukan dalam pembangunan sebuah sistem. Didalam tahap ini akan menentukan kebutuhan apa saja yang harus terdapat dalam dashboard produksi PTPN XII Gunung Gmbir. Persoalan yang didapat kurangnya efektif dan efisien dalam pengolahan informasi data produksi pada PTPN XII ini, baik dari segi pembuaan laporan, pengiriman laporan hingga monitoring biaya. Sehingga dibutuhkan sebuah sistem untuk dapat menangani masalah tersebut. Didalam tahapana ini juga akan dijelaskansiapa saja yang dapat mengakses sistem. Kebutuhan-kebutuhan serta siapa saja yang dapat mengakses sistem selanjutnya akan dibahas pada sub bab berikut.

### 4.1 Elisitasi Kebutuhan Sistem

Penggalian kebutuhan dalam penelitian ini dilakukan dengan cara observasi serta berkomunikasi langsung dengan pengguna sistem dan pihak lain yang memiliki kepentingan dalam pembangunan sistem. Observasi dilakukan dengan cara melihat langsung proses pembuatan laporan pruduksi dari mandor yang berada langsung dilapangan, hingga ketika mandor menyerahkan langsung ke juru tulis untuk direkap dan kemudian dikirim ke kantor induk. Hal itu dilakukan untuk mengetahui proses bisnis yang terjadi dalam proses pembuatan laporan produksi. Sedangkan wawancara dilakukan dengan mencari informasi proses perekapan laporan produksi serta kendala yang sering didapatkan ketika memonitoring produksi.

Proses wawancara dilakukan kepada juru tulis dan wakil kepala kantor tentang poses pembuatan laporan produksi, laporan harian, pengajuan modal kerja, siapa saja yang terlibat didalamnya, serta kendala yang sering dirasakan. Dari hasil wawancara tersebut didapatkan *stakeholder* siapa saja yang terlibat dengan sistem yang akan dibangun beserta peranannya. Identifikasi *stakeholder* dapat dilihat didalam tabel 4.1 berikut.

**Tabel 4.1 Identifikasi Stakeholder**

<b>Stakeholder</b>	<b>Deskripsi</b>
Pimpinan	Merupakan orang yang bertanggung jawab terhadap hasil produksi di PTPN XII Gunung Gambir kepada direksi.
Kantor Induk	Merupakan orang yang bertanggung jawab mengumpulkan informasi hasil produksi dari afdeling.
Sinder	Merupakan orang yang bertanggung jawab didalam afdeling.
Wakil Kepala Kantor	Merupakan orang yang bertanggung jawab untuk mengelola pengguna yang terdaftar didalam sistem

	yang akan dibangun. Wakil kepala kantor nantinya akan berperan sebagai admin.
Juru tulis	Orang yang bekerja di afdeling yang bertugas mencatat hasil produksi yang diperoleh kemudian mengirim data ke kantor induk.
Mandor	Orang yang bekerja di lapangan yang bertugas mencatat hasil produksi yang diperoleh

Terdapat lima *stakeholder* yang berperan dalam pengolahan produksi pada PTPN XII Gunung Gambir yaitu pimpinan, kantor induk, sinder, mandor, dan juru tulis. Pimpinan disini merupakan orang yang bertanggung jawab terhadap hasil produksi didalam PTPN XII, yang nantinya akan dipertanggung jawabkan kepada direksi. Pimpinan disini adalah Manajer, wakil manajer, serta kepala kantor induk. Kantor induk merupakan orang yang bertugas menghimpun hasil produksi dari masing-masing juru tulis afdeling yang ada di kebun Gunung Gambir. Wakil kepala kantor orang yang mempunyai hak untuk mengelola siapa saja yang terdaftar didalam sistem yang akan dibangun. Juru tulis memiliki peran menghimpun laporan dari masing-masing mandor kemudian melaporkan ke kantor induk.

Dari hasil observasi diperoleh proses atau tahapan pengolahan laporan produksi mulai dari mandor hingga tersampaikan ke kantor induk serta proses permintaan modal kerja. Pengepul melakukan penyadapan karet secara langsung. Kemudian dibawa ke penimbang untuk di hitung berapa berat total yang diperoleh. Kemudian mandor mencatat total semua getah karet yang dihasilkan pada hari itu. Mandor mencatatnya dalam buku secara manual. Kemudian dilaporkan kepada juru tulis. Juru tulis merekap semua laporan yang diterima dari mandor kedalam komputer melalui aplikasi *microsoft excel*. Kemudian juru tulis mengirim laporan tersebut kepada kantor induk dengan cara membacakan satu persatu hasil produksi pada hari itu melalui *handy talky*. Pada lampiran A akan dijelaskan proses bisnis yang terjadi dalam pembuatan laporan produksi. Dalam proses pembuatan laporan harian mandor mencatat pekerjaan yang dilakukan pada hari itu kemudian melaporkan kepada juru tulis. Juru tulis merekap laporan yang telah diterima kedalam komputer melalui aplikasi *microsoft excel*. Kemudian mengirimkan laporan harian kepada kantor induk melalui pesuratan (orang yang mengirim surat atau dokumen penting). Pada lampiran B akan dijelaskan proses bisnis yang terjadi dalam pembuatan laporan harian. Dalam proses ini diperoleh juga proses permintaan modal kerja. Dimana sinder mengajukan laporan modal kerja dalam bentuk laporan PPAP berdasarkan laporan harian yang sebelumnya telah direkap oleh juru tulis. Kemudian mengajukannya dalam rapat tahunan yang diadakan di kantor pusat. Pada lampiran C akan dijelaskan proses bisnis yang terjadi dalam proses pengajuan modal kerja.

Untuk pengelolaan target produksi, Pimpinan menentukan target produksi berdasarkan laporan produksi yang sebelumnya telah dihimpun didalam aplikasi *microsoft excel*. Kendala yang sering dihadapi yaitu merasa kesulitan ketika akan



membaca ataupun menganalisis data serta memprediksi data yang akan datang karena data yang disajikan dalam bentuk angka dan tabel yang sangat rumit.

#### 4.1.1 Gambaran Umum Sistem

Berdasarkan permasalahan yang telah dirumuskan dalam latar belakang sebelumnya, yang sebelumnya merupakan hasil dari observasi dan wawancara terhadap calon pengguna, didapatkan sebuah kebutuhan sistem dimana sistem harus dapat digunakan oleh juru tulis dan pimpinan PTPN XII kebun Gunung Gambir untuk melakukan *monitoring* terhadap hasil produksi dan mengawasi pelaksanaan dari rencana dan strategi yang telah dibuat agar tetap sejalan dengan tujuan yang telah ditentukan. Dimana juru tulis dapat membuat laporan produksi sesuai dengan hasil produksi yang diperoleh. Juru tulis juga mendapatkan informasi terkait perbandingan antara realisasi produksi dengan target yang akan dicapai guna membantu dalam menentukan strategi yang akan digunakan, juru tulis juga akan mendapatkan informasi terkait status produksi terhadap target produksi yang nantinya akan memberi peringatan terkait kondisi yang terjadi guna meminimalisir ketidaktercapainnya target produksi secara dini. Juru tulis juga mendapatkan informasi mengenai estimasi upah sebagai dasar untuk menetapkan biaya yang nantinya akan dikeluarkan. Juru tulis juga mendapatkan informasi terkait status biaya yang telah dikeluarkan terhadap target yang telah direncanakan guna membantu dalam menentukan strategi yang akan digunakan, serta juru tulis dapat memperoleh informasi estimasi biaya yang dikeluarkan berdasarkan laporan harian. Admin hanya dapat mengelola pengguna yang terdaftar didalam system. Sedangkan pimpinan disini terbagi menjadi empat yaitu manager, wakil manager, kepala kantor serta kantor induk. Pimpinan dapat secara langsung memonitoring tercapainya realisasi produksi terhadap target, serta memonitoring perbandingan biaya yang telah dikeluarkan terhadap estimasi biaya yang direncanakan. Pimpinan dapat memberikan komentar terhadap juru tulis yang mewakili kantor afdeling jika terjadi ketidaksesuaian anantara rencana dan strategi dengan yang terjadi dilapangan. Pimpinan juga dapat menentukan target produksi dan anggaran perusahaan. Semua kebutuhan itu nantinya akan dibangun kedalam sistem *dashboard* yang berbasis web. Sehingga nantinya pengiriman data dapat dengan mudah dilakukan karena berbasis web dan tampilan akan lebih menarik dan mudah dipahami karena menggunakan *dashboard* sehingga memudahkan pimpinan dalam melakukan monitoring dan penentuan strategi kedepannya.

#### 4.2 Spesifikasi Kebutuhan

Spesifikasi kebutuhan dilakukan untuk mengidentifikasi aktor beserta batasannya serta menspesifikasi hasil dari proses elisitasi yang dikategorikan menjadi beberapa kebutuhan seperti kebutuhan fungsional dan kebutuhan *non-fungsional*. Kemudian dilakukan pemodelan kebutuhan dalam bentuk *use case* diagram serta *use case scenario*.

#### 4.2.1 Identifikasi Aktor

Identifikasi aktor bertujuan untuk mengetahui aktor-aktor yang nantinya akan terlibat didalam sistem beserta dengan wewenang dari masing-masing aktor tersebut terhadap sistem. Berdasarkan permasalahan yang telah dirumuskan didalam latar belakang, aktor yang terdapat didalam sistem yang akan dibangun nantinya adalah pimpinan, kantor induk dan juru tulis. Identifikasi aktor didapat dari hasil observasi dan wawancara. Pada tabel 4.2 akan dijelaskan aktor-aktor yang terlibat dalam sistem beserta dengan deskripsi peran dari masing-masing aktor tersebut.

**Tabel 4.2 Identifikasi Aktor**

Aktor	Deskripsi
User	User merupakan aktor yang dapat menggunakan sistem. User adalah aktor yang belum melakukan login. User hanya dapat melihat halaman awal sistem saja dan memilih aksi <i>login</i> .
Admin	Admin bertugas untuk mengatur pengguna lain yaitu kantor induk, pimpinan dan juru tulis seperti tambah kantor induk, tambah pimpinan, tambah juru tulis, hapus kantor induk, hapus pimpinan, hapus juru tulis, ubah kantor induk, ubah ubah, dan ubah juru tulis.
Kantor Induk	Kantor induk merupakan aktor yang dapat menggunakan sistem. Kantor induk dapat memantau sistem seperti melihat laporan produksi, melihat laporan harian, mencetak laporan produksi, mencetak laporan harian, melihat status realisasi produksi terhadap target, dan melihat status biaya produksi saat ini terhadap estimasi biaya yang telah direncanakan. Kantor induk juga dapat melihat komentar yang diberikan oleh pimpinan kepada juru tulis dan menambahkan target produksi serta target biaya yang harus dicapai.
Pimpinan	Pimpinan merupakan aktor yang dapat menggunakan sistem. Kantor induk dapat memantau sistem seperti melihat laporan produksi, melihat laporan harian, mencetak laporan produksi, mencetak laporan harian, melihat status realisasi produksi terhadap target, dan melihat status biaya produksi saat ini terhadap estimasi biaya yang telah direncanakan. Pimpinan juga dapat menambahkan target produksi serta target biaya yang harus dicapai. Pimpinan juga dapat menambahkan komentar, menghapus komentar serta mengubah komentar.
Juru Tulis	Juru tulis adalah aktor pengguna sistem yang dapat melihat laporan produksi, melihat laporan harian, menghapus laporan produksi, menghapus laporan harian, mengubah



**Tabel 4.2 Identifikasi Aktor**

Aktor	Deskripsi
	laporan produksi, mengubah laporan harian, mencetak laporan produksi, mencetak laporan harian, melihat status realisasi produksi terhadap target, dan melihat status biaya produksi saat ini terhadap estimasi biaya yang telah direncanakan, melihat komentar.
Mandor	Mandor adalah aktor pengguna sistem yang dapat mengelola laporan produksi, mengelola laporan harian, mencetak laporan produksi, mencetak laporan harian, melihat status realisasi produksi terhadap target, dan melihat status biaya produksi saat ini terhadap dan dapat melihat komentar.

#### 4.2.2 Daftar Kebutuhan Fungsional

Kebutuhan fungsional merupakan suatu layanan yang harus dapat disediakan oleh sistem, bagaimana reaksi sistem ketika menerima masukan tertentu dan bagaimana reaksi sistem pada situasi tertentu. Kebutuhan fungsional adalah jenis kebutuhan yang berisikan proses-proses apa saja yang dapat diberikan oleh sistem kepada pengguna. Didalam penelitian ini kebutuhan fungsional didapat dari hasil observasi serta wawancara kepada beberapa karyawan dan pimpinan PTPN XII Gunung Gambir yang merupakan calon pengguna sistem. Masing-masing kebutuhan akan diberikan kode DPF\_YYZZ. Dimana YY menunjukkan nomor kebutuhan utama sedangkan ZZ menunjukkan nomor spesifikasi dari kebutuhan utama. Kebutuhan fungsional dapat dilihat dari tabel 4.5 berikut.

**Tabel 4.3 Kebutuhan Fungsional Sistem**

No	Kode	Spesifikasi
1.	DPF_0100	Sistem harus dapat <i>login</i> untuk mengizinkan pengguna mengakses sistem serta dapat membedakan pengguna sesuai dengan <i>privilege</i> masing-masing.
1.1.	DPF_0101	Formulir <i>login</i> yang harus diisi adalah <i>username</i> berupa NIK dan <i>password</i> .
2.	DPF_0200	Sistem harus dapat <i>logout</i> agar admin, kantor induk, juru tulis dan pimpinan dapat keluar dari sistem.
3.	DPF_0300	Sistem harus dapat membuat laporan produksi.
3.1.	DPF_0301	Laporan produksi yang harus diisi adalah tanggal, nama mandor, tahun tanam, realisasi OHK, realisasi produksi.
4.	DPF_0400	Sistem harus dapat mengubah laporan produksi oleh juru tulis.



Tabel 4.3 Kebutuhan Fungsional Sistem

No	Kode	Spesifikasi
4.1.	DPF_0401	Laporan produksi yang dapat diubah adalah tanggal, nama mandor, tahun tanam, realisasi OHK, realisasi produksi.
5.	DPF_0500	Sistem harus dapat menghapus laporan produksi oleh juru tulis.
6.	DPF_0600	Sistem harus dapat membuat laporan harian.
6.1.	DPF_0601	Laporan harian yang harus diisi adalah tanggal, macam pekerjaan, lokasi (nomor) kebun/ tahun tanam, jumlah karyawan Gol. IA, jumlah karyawan lepas/bor, hasil pekerjaan tetap, hasil pekerjaan lepas, satuan upah IA, satuan upah KHL, satuan upah BOR dan estimasi upah.
7.	DPF_0700	Sistem harus dapat mengubah laporan harian oleh juru tulis.
7.11	DPF_0701	Laporan harian yang dapat diubah adalah macam pekerjaan, lokasi (nomor) kebun/ tahun tanam, jumlah karyawan Gol. IA, jumlah karyawan lepas/bor, hasil pekerjaan tetap, hasil pekerjaan lepas, satuan upah IA, satuan upah KHL, satuan upah BOR dan estimasi upah.
8.	DPF_0800	Sistem harus dapat menghapus laporan harian oleh juru tulis.
9.	DPF_0900	Sistem harus dapat menampilkan laporan produksi. Laporan produksi yang ditampilkan adalah tanggal, nama mandor, tahun tanam, realisasi OHK, realisasi produksi.
10.	DPF_1000	Sistem harus dapat menampilkan laporan harian. Laporan harian yang ditampilkan adalah tanggal, macam pekerjaan, lokasi (nomor) kebun/ tahun tanam, jumlah karyawan Gol. IA, jumlah karyawan lepas/bor, hasil pekerjaan tetap, hasil pekerjaan lepas, satuan upah IA, satuan upah KHL, satuan upah BOR dan estimasi upah.
11.	DPF_1100	Sistem harus dapat mencetak laporan produksi.
11.1.	DPF_1101	Laporan yang dicetak berisi tanggal, nama mandor, tahun tanam, target OHK, realisasi OHK, target produksi, realisasi produksi.
12.	DPF_1200	Sistem harus dapat mencetak laporan harian.

Tabel 4.3 Kebutuhan Fungsional Sistem

No	Kode	Spesifikasi
12.1.	DPF_1201	Laporan yang dicetak berisi tanggal, macam pekerjaan, lokasi (nomor) kebun/ tahun tanam, jumlah karyawan Gol. IA, jumlah karyawan lepas/bor, hasil pekerjaan tetap, hasil pekerjaan lepas, satuan upah IA, satuan upah KHL, satuan upah BOR, estimasi upah.
13.	DPF_1300	Sistem harus dapat memberikan informasi status ketercapaian realisasi produksi saat ini terhadap target.
13.1	DPF_1301	Status realisasi yang diberikan berupa <i>chart / speedometer</i> yang berwarna merah jika realisasi kurang dari 50%. Akan menampilkan <i>chart / speedometer</i> yang berwarna kuning jika realisasi lebih dari 50% dan kurang dari 100% dan akan menampilkan <i>chart / speedometer</i> yang berwarna hijau jika target lebih dari 100%.
14.	DPF_1400	Sistem harus dapat menampilkan total realisasi produksi selama satu tahun dalam bentuk grafik.
14.1.	DPF_1401	Sistem harus dapat menampilkan perbandingan total realisasi produksi tahun ini dengan tahun sebelumnya dalam bentuk grafik.
15.	DPF_1500	Sistem harus dapat menghitung estimasi biaya produksi yang akan dikeluarkan.
16.	DPF_1600	Sistem harus dapat memberikan informasi status biaya yang dikeluarkan saat ini terhadap realisasi sebagai <i>warning</i> agar biaya yang dikeluarkan sesuai dengan realisasi yang direncanakan.
16.1.	DPF_1601	Status biaya yang ditampilkan jika biaya yang dikeluarkan kurang dari 50% akan menampilkan <i>chart / speedometer</i> berwarna hijau, jika biaya yang dikeluarkan kurang dari 80% akan menampilkan <i>chart / speedometer</i> , jika biaya yang dikeluarkan lebih dari 80% akan menampilkan <i>chart / speedometer</i> berwarna merah.
17.	DPF_1700	Sistem harus dapat menambah target produksi yang harus dicapai.
18.	DPF_1800	Sistem harus dapat menambah target biaya yang akan dikeluarkan.
19.	DPF_1900	Sistem harus dapat menambahkan komentar.

Tabel 4.3 Kebutuhan Fungsional Sistem

No	Kode	Spesifikasi
20.	DPF_2000	Sistem harus dapat mengubah komentar.
21.	DPF_2100	Sistem harus dapat menghapus komentar.
22.	DPF_2200	Sistem harus dapat menampilkan komentar yang telah ditambahkan. Komentar yang ditampilkan adalah nama pimpinan dan isi komentar.
23.	DPF_2300	Sistem harus dapat menambah pengguna oleh admin. Data pengguna yang harus diisi adalah NIP/NIP sebagai username, nama, jabatan dan <i>password</i> .
24.	DPF_2400	Sistem harus dapat mengubah data pengguna.
24.1.	DPF_2401	Data pengguna yang dapat diubah adalah NIP/NIP, nama, jabatan dan <i>password</i> .
25.	DPF_2500	Sistem harus dapat menampilkan data pengguna. Data pengguna yang ditampilkan berupa NIK/NIP dan nama.
25.1.	DPF_2501	Sistem harus dapat menampilkan data pengguna secara detail. Data pengguna yang ditampilkan adalah NIP/NIP, nama, jabatan dan <i>password</i> .
26.	DPF_2600	Sistem harus dapat menghapus data pengguna yang dapat mengakses sistem.

#### 4.2.3 Daftar Kebutuhan *Non-Fungsional*

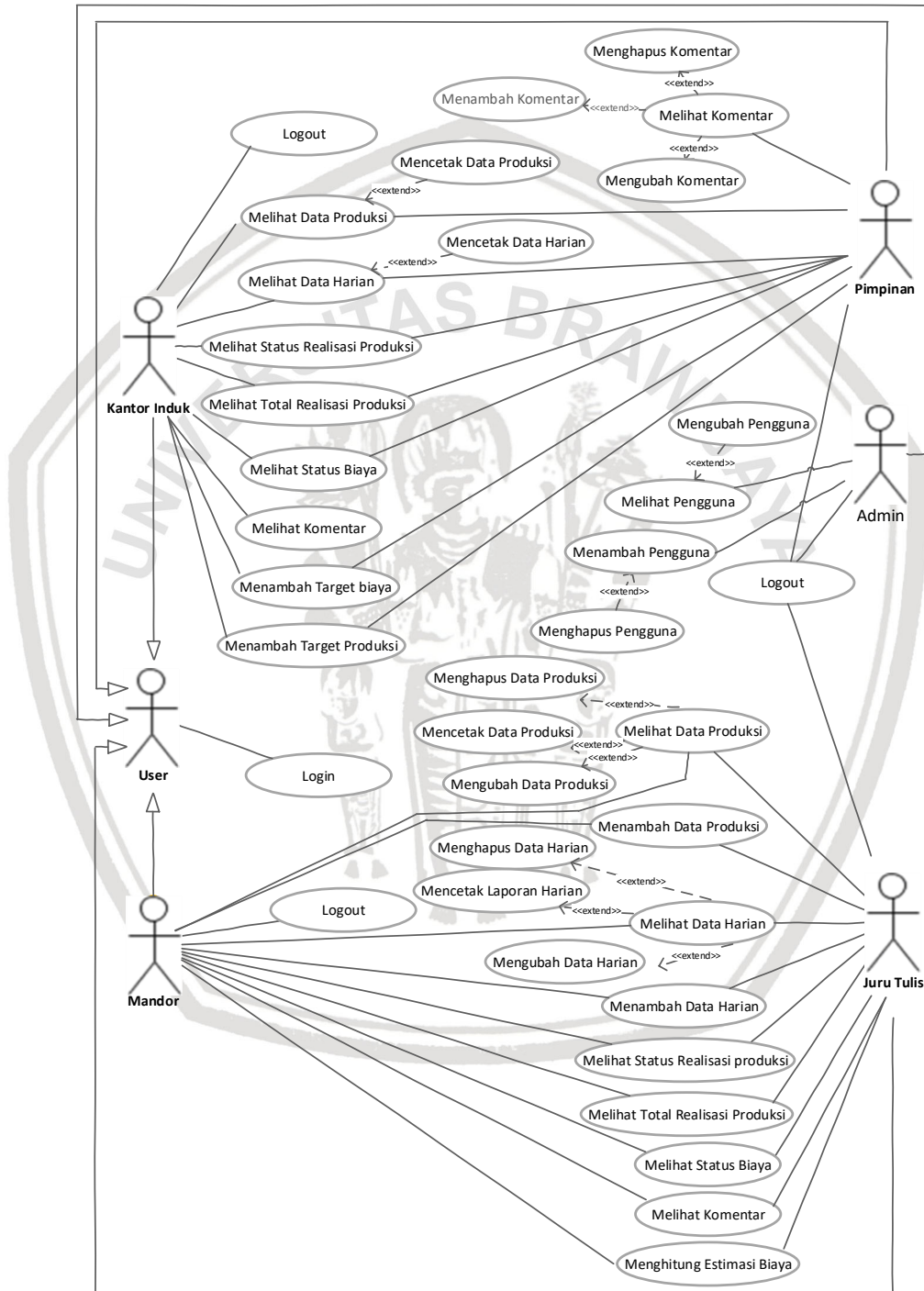
Kebutuhan non fungsional merupakan kebutuhan yang berkaitan dengan batasan atau kualitas dari sistem yang akan dibangun. Dalam penelitian ini kebutuhan non fungsional didapatkan dari hasil wawancara dengan beberapa karyawan PTPN XII Gunung Gambir. Setiap kebutuhan telah divalidasi dan diverifikasi dengan cara melakukan komunikasi dengan calon pengguna. Hasil dari proses validasi menunjukkan bahwa kebutuhan non fungsional dalam penelitian ini telah benar dan lengkap sesuai dengan keinginan calon pengguna. Sedangkan hasil proses verifikasi menunjukkan bahwa kebutuhan non fungsional dalam penelitian ini dapat dengan mudah untuk dipahami, mudah untuk dimengerti serta tidak ambigu. Masing-masing kebutuhan akan diberikan kode DPNF\_ZZ. Spesifikasi kebutuhan non fungsional dapat dilihat dari tabel 4.4 berikut.

Tabel 4.4 Spesifikasi Kebutuhan Non Fungsional Sistem

No	Parameter	Spesifikasi	Kode
1.	<i>Compatibility</i>	Sistem harus dapat dibuka didalam aplikasi <i>browser Mozilla Firefox</i> dan <i>Google Chrome</i> .	DPNF_01

### 4.2.4 Use Case Diagram

Use case diagram digunakan untuk memodelkan perilaku aktor kepada sistem yang akan dibangun. Yang ditekankan adalah “apa” yang diperbuat sistem, bukan “bagaimana”. Use case diagram dari sistem yang akan dibangun dibuat berdasarkan daftar kebutuhan fungsional yang telah dibuat sebelumnya dapat dilihat dari gambar 4.1 berikut.



Gambar 4.1 Use Case Diagram





#### 4.2.5 Use Case Scenario

*Use case scenario* menjelaskan tentang alur dari *use case* yang telah didefinisikan pada *use case diagram*. *Use case scenario* dimodelkan menggunakan tabel yang terdiri dari *actor*, *objective*, *pre-condition*, *main flow*, dan *post condition*. Terdapat 26 *use case scenario* yang dibuat dan akan dijelaskan dalam tabel 4.7 sampai tabel 4.32.

##### 4.2.5.1 Use Case Scenario Login

Pada tabel 4.7 menjelaskan *use case scenario login* mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.5 Use Case Scenario Login**

DPF_0100 : Login	
Objective	Melakukan proses autentikasi
Actor	User
Pre-Condition	Pengguna telah terhubung dengan internet dan membuka halaman awal <i>website</i> Dashboard Produksi PTPN XII Gunung Gambir.
Main Flow	<ol style="list-style-type: none"> <li>1. User mengisi username berupa NIP/NIK dan password pada form login.</li> <li>2. User menekan tombol Login.</li> <li>3. Sistem mengecek apakah username dan password telah terautentikasi.</li> <li>4. User masuk ke halaman Beranda.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>3a. Jika username atau password belum terisi maka sistem akan menampilkan peringatan “Harap isi bidang ini” dan kembali pada main flow nomor 1.</li> <li>4a. Jika username atau password gagal diautentikasi maka sistem akan menampilkan pesan “username atau password salah” dan akan kembali pada main flow nomor 1.</li> </ol>
Post-Condition	User berhasil terotentikasi dan telah masuk kedalam halaman beranda.

##### 4.2.5.2 Use Case Scenario Logout

Pada tabel 4.8 menjelaskan *use case scenario logout* mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.6 Use Case Scenario Logout**

DPF_0200 : Logout	
Objective	Keluar dari sistem



Tabel 4.6 Use Case Scenario Logout

DPF_0200 : Logout	
Actor	Admin, Pimpinan, Kantor Induk, Juru Tulis
Pre-Condition	Pengguna telah berhasil terautentikasi oleh sistem.
Main Flow	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol logout.</li> <li>2. Sistem menghapus session dan menampilkan halaman awal <i>website</i>.</li> </ol>
Alternative Flow	-
Post-Condition	User berhasil keluar dari sistem

#### 4.2.5.3 Use Case Scenario Menambah Data Produksi

Pada tabel 4.9 menjelaskan *use case scenario* Menambah Data Produksi mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

Tabel 4.7 Use Case Scenario Menambah Data Produksi

DPF_0300 : Menambah Data Produksi	
Objective	Menambah Data produksi baru
Actor	Juru Tulis
Pre-Condition	Juru Tulis telah berada didalam menu Data Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Juru Tulis menekan tombol Tambah Laporan.</li> <li>2. Sistem mengirim form untuk diisi Juru Tulis sesuai dengan format laporan produksi.</li> <li>3. Juru Tulis mengisi data laporan produksi seperti tanggal, nama mandor, tahun tanam, realisasi OHK, realisasi produksi, mutu super, mutu infer.</li> <li>4. Juru Tulis menekan tombol simpan.</li> <li>5. Sistem mengambil form yang telah diisi oleh Juru Tulis dan menyimpan data pada database.</li> <li>6. Sistem memberi notifikasi jika laporan telah berhasil disimpan.</li> <li>7. Sistem menampilkan halaman utama dalam menu Laporan Produksi.</li> </ol>
Alternative Flow	5a. Jika salah satu dari field tidak diisi maka sistem akan menampilkan peringatan "Harap isi bidang ini" dan kembali pada main flow nomor 2.
Post-Condition	Laporan produksi telah berhasil disimpan dalam database.

#### 4.2.5.4 Use Case Scenario Mengubah Data Produksi

Pada tabel 4.10 menjelaskan *use case scenario* mengubah Data produksi mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.8 Use Case Scenario Mengubah Data Produksi**

DPF_0400 : Mengubah Data Produksi	
Objective	Mengubah Data produksi
Actor	Juru Tulis
Pre-Condition	Juru Tulis telah berada didalam menu Laporan Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Juru Tulis menekan tombol Edit yang tersedia untuk memilih laporan produksi yang ingin diubah.</li> <li>2. Sistem mengambil data laporan produksi dari database sesuai laporan produksi yang dipilih pengguna.</li> <li>3. Sistem mengirim form untuk diisi Juru Tulis sesuai dengan format laporan produksi.</li> <li>4. Juru Tulis mengisi data laporan produksi yang ingin diubah seperti tanggal, nama mandor, tahun tanam, realisasi OHK, realisasi produksi.</li> <li>5. Juru Tulis menekan tombol simpan.</li> <li>6. Sistem mengambil form yang telah diisi oleh Juru Tulis dan menyimpan data pada database.</li> <li>7. Sistem memberi notifikasi jika laporan telah berhasil disimpan.</li> <li>8. Sistem menampilkan halaman utama dalam menu Laporan Produksi.</li> </ol>
Alternative Flow	6a. Jika salah satu dari field tidak diisi maka sistem akan menampilkan peringatan "Harap isi bidang ini" dan kembali pada main flow nomor 5.
Post-Condition	Laporan produksi telah berhasil diubah dan disimpan dalam database.

#### 4.2.5.5 Use Case Scenario Menghapus Laporan Produksi

Pada tabel 4.11 menjelaskan *use case scenario* menghapus laporan produksi mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.9 Use Case Scenario Menghapus Data Produksi**

DPF_0500 : Menghapus Data Produksi	
Objective	Mengapus Data produksi

**Tabel 4.9 Use Case Scenario Menghapus Data Produksi**

<b>DPF_0500 : Menghapus Data Produksi</b>	
Actor	Juru Tulis
Pre-Condition	Juru Tulis telah berada didalam menu Laporan Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Juru Tulis menekan tombol Hapus untuk memilih laporan produksi yang ingin dihapus.</li> <li>2. Sistem menghapus data laporan produksi dari database sesuai dengan laporan yang dipilih pengguna kemudian menampilkan kepada pengguna.</li> <li>3. Sistem menampilkan halaman utama dalam menu Laporan produksi.</li> </ol>
Alternative Flow	-
Post-Condition	Laporan produksi telah berhasil dihapus dari database.

**4.2.5.6 Use Case Scenario Menambah Data Harian**

Pada tabel 4.12 menjelaskan *use case scenario* menambah Data harian mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.10 Use Case Scenario Menambah Data Harian**

<b>DPF_0600 : Menambah Data Harian</b>	
Objective	Menambah Data Harian baru
Actor	Juru Tulis
Pre-Condition	Juru Tulis telah berada didalam menu Laporan Harian
Main Flow	<ol style="list-style-type: none"> <li>1. Juru Tulis menekan tombol Tambah Laporan</li> <li>2. Sistem mengirim form untuk diisi Juru Tulis sesuai dengan format laporan harian.</li> <li>3. Juru Tulis mengisi data laporan harian seperti tanggal, adalah macam pekerjaan, lokasi (nomor) kebun/ tahun tanam, jumlah karyawan Gol. IA, jumlah karyawan lepas/bor, hasil pekerjaan tetap, hasil pekerjaan lepas, satuan upah IA, satuan upah KHL, satuan upah BOR.</li> <li>4. Juru Tulis menekan tombol simpan.</li> <li>5. Sistem mengambil form yang telah diisi oleh Juru Tulis dan menyimpan data pada database.</li> <li>6. Sistem menampilkan halaman utama dalam menu Laporan Harian.</li> </ol>



**Tabel 4.10 Use Case Scenario Menambah Data Harian**

<b>DPF_0600 : Menambah Data Harian</b>	
Alternative Flow	5a. Jika salah satu dari field tidak diisi maka sistem akan menampilkan peringatan “Harap isi bidang ini” dan kembali pada main flow nomor 2.
Post-Condition	Laporan harian telah berhasil disimpan dalam database.

**4.2.5.7 Use Case Scenario Mengubah Data Harian**

Pada tabel 4.13 menjelaskan *use case scenario* mengubah Data harian mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.11 Use Case Scenario Mengubah Data Harian**

<b>DPF_0700 : Mengubah Data harian</b>	
Objective	Mengubah Data harian
Actor	Juru Tulis
Pre-Condition	Juru Tulis telah berada didalam menu Laporan Harian
Main Flow	<ol style="list-style-type: none"> <li>1. Juru Tulis memilih tanggal bulan, dan tahun yang tersedia untuk memilih laporan harian yang ingin diubah.</li> <li>2. Sistem mengambil data laporan harian dari database sesuai dengan tanggal bulan dan tahun yang telah di inputkan pengguna kemudian menampilkan kepada pengguna.</li> <li>3. Juru Tulis menekan tombol Edit.</li> <li>4. Sistem mengirim form untuk diisi Juru Tulis sesuai dengan format laporan produksi.</li> <li>5. Juru Tulis mengisi data laporan produksi yang ingin diubah seperti tanggal, adalah macam pekerjaan, lokasi (nomor) kebun/ tahun tanam, jumlah karyawan Gol. IA, jumlah karyawan lepas/bor, hasil pekerjaan tetap, hasil pekerjaan lepas, satuan upah IA, satuan upah KHL, satuan upah BOR.</li> <li>6. Juru Tulis menekan tombol simpan.</li> <li>7. Sistem mengambil form yang telah diisi oleh Juru Tulis dan menyimpan data pada database.</li> <li>8. Sistem menampilkan halaman utama dalam menu Laporan Harian.</li> </ol>



**Tabel 4.11 Use Case Scenario Mengubah Data Harian**

<b>DPF_0700 : Mengubah Data harian</b>	
Alternative Flow	7a. Jika salah satu dari field tidak diisi maka sistem akan menampilkan peringatan “Harap isi bidang ini” dan kembali pada main flow nomor 5.
Post-Condition	Laporan harian telah berhasil diubah dan disimpan dalam database.

#### 4.2.5.8 Use Case Scenario Menghapus Data Harian

Pada tabel 4.14 menjelaskan *use case scenario* menghapus Data harian mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.12 Use Case Scenario Menghapus Data Harian**

<b>DPF_0800 : Menghapus Data Harian</b>	
Objective	Mengapus Data harian
Actor	Juru Tulis
Pre-Condition	Juru Tulis telah berada didalam menu Laporan Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Juru Tulis memilih tanggal, bulan, dan tahun yang tersedia untuk memilih laporan harian yang ingin dihapus.</li> <li>2. Sistem menampilkan data laporan harian dari database sesuai dengan tanggal bulan dan tahun yang telah diinputkan pengguna.</li> <li>3. Pengguna menekan tombol Hapus.</li> <li>4. Sistem menampilkan halaman utama dalam menu Laporan produksi.</li> </ol>
Alternative Flow	-
Post-Condition	Laporan harian telah berhasil dihapus dari database.

#### 4.2.5.9 Use Case Scenario Melihat Data Produksi

Pada tabel 4.15 menjelaskan *use case scenario* melihat Data produksi mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.13 Use Case Scenario Melihat Data Produksi**

<b>DPF_0900 : Melihat Data Produksi</b>	
Objective	Menampilkan Data produksi



**Tabel 4.13 Use Case Scenario Melihat Data Produksi**

<b>DPF_0900 : Melihat Data Produksi</b>	
Actor	Pimpinan, Kantor Induk, Juru Tulis
Pre-Condition	Pimpinan, Kantor Induk, Juru Tulis telah berada didalam menu Laporan Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan, Kantor Induk, Juru Tulis memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan produksi yang ingin tampilkan.</li> <li>2. Sistem menampilkan data laporan produksi dari database sesuai dengan tanggal bulan dan tahun yang telah di inputkan pengguna.</li> </ol>
Alternative Flow	-
Post-Condition	Laporan produksi telah berhasil ditampilkan.

**4.2.5.10 Use Case Scenario Melihat Data Harian**

Pada tabel 4.16 menjelaskan *use case scenario* melihat Data harian mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.14 Use Case Scenario Melihat Data Harian**

<b>DPF_1000 : Melihat Data harian</b>	
Objective	Menampilkan Data harian
Actor	Pimpinan, Kantor Induk, Juru Tulis
Pre-Condition	Pimpinan, Kantor Induk, Juru Tulis telah berada didalam menu Laporan Harian
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan, Kantor Induk, Juru Tulis memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan harian yang ingin ditampilkan.</li> <li>2. Sistem mengambil data laporan harian dari database sesuai dengan tanggal bulan dan tahun yang telah di inputkan pengguna kemudian menampilkan kepada pengguna.</li> </ol>
Alternative Flow	-
Post-Condition	Laporan produksi telah berhasil ditampilkan.



#### 4.2.5.11 Use Case Scenario Mencetak Laporan Produksi

Pada tabel 4.17 menjelaskan *use case scenario* mencetak laporan produksi mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.15 Use Case Scenario Mencetak Laporan Produksi**

<b>DPF_1100 : Mencetak Laporan Produksi</b>	
Objective	Mencetak laporan produksi
Actor	Pimpinan, Kantor Induk, Juru Tulis
Pre-Condition	Pimpinan, Kantor Induk, Juru Tulis telah berada didalam menu Laporan Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan, Kantor Induk, Juru Tulis memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan harian yang ingin ditampilkan.</li> <li>2. Sistem mengambil data laporan harian dari database sesuai dengan tanggal bulan dan tahu yang telah di inputkan pengguna kemudian mencetak laporan produksi.</li> </ol>
Alternative Flow	-
Post-Condition	Laporan produksi telah berhasil ditampilkan dan berhasil dicetak.

#### 4.2.5.12 Use Case Scenario Mencetak Laporan Harian

Pada tabel 4.18 menjelaskan *use case scenario* mencetak laporan harian mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.16 Use Case Scenario Mencetak Laporan Harian**

<b>DPF_1200 : Mencetak Laporan Harian</b>	
Objective	Mencetak laporan harian
Actor	Pimpinan, Kantor Induk, Juru Tulis
Pre-Condition	Pimpinan, Kantor Induk, Juru Tulis telah berada didalam menu Laporan Harian
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan, Kantor Induk, Juru Tulis memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan harian yang ingin ditampilkan.</li> <li>2. Sistem mengambil data laporan harian dari database sesuai dengan tanggal bulan dan tahu yang telah di</li> </ol>

**Tabel 4.16 Use Case Scenario Mencetak Laporan Harian**

DPF_1200 : Mencetak Laporan Harian	
	inputkan pengguna kemudian mencetak laporan produksi.
Alternative Flow	-
Post-Condition	Laporan harian telah berhasil ditampilkan dan berhasil dicetak.

#### 4.2.5.13 Use Case Scenario Melihat Status Realisasi Produksi

Pada tabel 4.19 menjelaskan *use case scenario* melihat status produksi mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.17 Use Case Scenario Melihat Status Realisasi Produksi**

DPF_1300 : Melihat Status Realisasi Porduksi	
Objective	Melihat status produksi saat ini
Actor	Pimpinan, Kantor Induk, Juru Tulis
Pre-Condition	Pimpinan, Kantor Induk, Juru Tulis telah berada didalam menu Monitoring Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan, Kantor Induk, Juru Tulis memilih bulan dan tahun yang tersedia dalam fungsi dropdown yang akan ditampilkan status realisasi produksinya.</li> <li>2. Sistem mengambil data total produksi dari database sesuai dengan tanggal bulan dan tahun yang telah dipilih.</li> <li>3. Sistem menampilkan status realisasi produksi kepada pengguna berupa <i>speedometer</i> yang berwarna merah jika realisasi &lt; 50%. Akan menampilkan <i>speedometer</i> yang berwarna kuning jika realisasi &gt; 50% dan &lt; 100% dan akan menampilkan <i>speedometer</i> yang berwarna hijau jika target &gt; 100%.</li> </ol>
Alternative Flow	-
Post-Condition	Pimpinan, Kantor Induk, atau Juru Tulis mendapatkan infromasi status produksi.

#### 4.2.5.14 Use Case Scenario Melihat Total Produksi

Pada tabel 4.20 menjelaskan *use case scenario* melihat total produksi mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.



**Tabel 4.18 Use Case Scenario Total Produksi**

<b>DPF_1400 : Melihat Total Realisasi Produksi</b>	
Objective	Melihat perbandingan total ealisasi produksi yang telah dicapai
Actor	Pimpinan, Kantor Induk, Juru Tulis
Pre-Condition	Pimpinan, Kantor Induk, atau Juru Tulis telah berada didalam menu Monitoring Porduksi
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan, Kantor Induk, atau Juru Tulis memilih tahun yang tersedia dalam fungsi dropdown yang akan ditampilkan status realisasi produksinya</li> <li>2. Sistem mengambil data hasil produksi didalam database sesuai dengan tahun yang dipilih oleh Pimpinan, Kantor Induk, atau Juru Tulis.</li> <li>3. Sistem akan menampilkan perbandingan total realisasi produksi sesuai dengan tahun yang dipilih.</li> </ol>
Alternative Flow	-
Post-Condition	Pimpinan, Kantor Induk, atau Juru Tulis mendapatkan infromasi mengenai perbandingan realisasi produksi terhadap target berdasarkan tahun yang dipilih.

**4.2.5.15 Use Case Scenario Menghitung Estimasi Biaya**

Pada tabel 4.21 menjelaskan *use case scenario* menghitung estimasi biaya mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.19 Use Case Scenario Menghitung Estimasi Biaya**

<b>DPF_1500 : Menghitung Estimasi Biaya</b>	
Objective	Menghitung estimasi biaya produksi
Actor	Juru Tulis
Pre-Condition	Juru Tulis telah berada didalam menu Anggaran
Main Flow	<ol style="list-style-type: none"> <li>1. Juru tulis menekan tombol Estimasi</li> <li>2. Sistem menampilkan form untuk menghitung estimasi biaya.</li> <li>3. Juru Tulis memilih bulan, tahun dan masa periode yang akan dihitung estimasi biaya produksinya.</li> <li>4. Juru tulis menekan tombol hitung estimasi biaya</li> <li>5. Sistem mengambil data laporan harian didalam database sesuai dengan bulan tahun dan masa periode yang dipilih oleh Juru Tulis.</li> </ol>

Tabel 4.19 *Use Case Scenario* Menghitung Estimasi Biaya

DPF_1500 : Menghitung Estimasi Biaya	
	<ol style="list-style-type: none"> <li>6. Sistem akan menghitung estimasi biaya produksi.</li> <li>7. Sistem akan menampilkan total estimasi biaya produksi yang akan dikeluarkan.</li> </ol>
Alternative Flow	-
Post-Condition	Juru Tulis mendapatkan informasi mengenai total realisasi biaya yang akan dikeluarkan.

#### 4.2.5.16 *Use Case Scenario* Melihat Status Biaya

Pada tabel 4.22 menjelaskan *use case scenario* melihat status biaya mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

Tabel 4.20 *Use Case Scenario* Melihat Status Biaya

DPF_1600 : Melihat Status Biaya	
Objective	Melihat status biaya yang dikeluarkan saat ini
Actor	Pimpinan, Kantor Induk, Juru Tulis
Pre-Condition	Pimpinan, Kantor Induk, atau Juru Tulis telah berada didalam menu Anggaran
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan, Kantor Induk, atau Juru Tulis memilih bulan yang tersedia yang akan ditampilkan status anggarannya.</li> <li>2. Sistem mengambil data biaya yang dikeluarkan didalam database sesuai dengan bulan yang dipilih oleh Pimpinan, Kantor Induk, atau Juru Tulis</li> <li>3. Sistem akan menampilkan status biaya yang dikeluarkan berupa tulisan jika biaya yang dikeluarkan &lt; 50% dari target akan menampilkan <i>speedometer</i> berwarna hijau, jika biaya yang dikeluarkan &lt; 80% dari target akan menampilkan <i>speedometer</i> berwarna kuning, jika biaya yang dikeluarkan &gt; 80% dari target akan menampilkan <i>speedometer</i> berwarna merah.</li> </ol>
Alternative Flow	-
Post-Condition	Pimpinan, Kantor Induk, atau Juru Tulis mendapatkan informasi mengenai status biaya yang dikeluarkan sesuai dengan bulan yang dipilih.



#### 4.2.5.17 Use Case Scenario Menambah Target Produksi

Pada tabel 4.23 menjelaskan *use case scenario* menambah target produksi mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.21 Use Case Scenario Menambah Target Produksi**

DPF_1700 : Menambah Target Produksi	
Objective	Menambah target produksi
Actor	Pimpinan, Kantor Induk
Pre-Condition	Pimpinan atau Kantor Induk telah berada didalam menu Laporan Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan atau kantor induk menekan tombol tambah target.</li> <li>2. Sistem mengirim form untuk diisi Juru Tulis sesuai dengan format target produksi.</li> <li>3. Pimpinan atau kantor Induk mengisi jumlah target produksi.</li> <li>4. Pimpinan atau kantor Induk menekan tombol simpan.</li> <li>5. Sistem mengambil form yang telah diisi oleh Pimpinan atau kantor Induk dan menyimpan data pada database.</li> <li>6. Sistem memberi notifikasi jika target produksi telah berhasil disimpan.</li> <li>7. Sistem menampilkan halaman utama dalam menu Laporan Produksi.</li> </ol>
Alternative Flow	5a. Jika salah satu dari field tidak diisi maka sistem akan menampilkan peringatan "Harap isi bidang ini" dan kembali pada main flow nomor 2.
Post-Condition	Target produksi telah berhasil disimpan dalam database.

#### 4.2.5.18 Use Case Scenario Menambah Target Biaya

Pada tabel 4.24 menjelaskan *use case scenario* menambah target biaya mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.22 Use Case Scenario Menambah Target Biaya**

DPF_1800 : Menambah Target Biaya	
Objective	Menambah target biaya
Actor	Pimpinan, kantor Induk

**Tabel 4.22 Use Case Scenario Menambah Target Biaya**

<b>DPF_1800 : Menambah Target Biaya</b>	
Pre-Condition	Pimpinan atau Kantor Induk telah berada didalam menu Anggaran
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan atau kantor Induk menekan tombol tambah target.</li> <li>2. Sistem mengirim form untuk diisi Juru Tulis sesuai dengan format target biaya.</li> <li>3. Pimpinan atau kantor Induk mengisi jumlah target biaya.</li> <li>4. Pimpinan atau kantor Induk menekan tombol simpan.</li> <li>5. Sistem mengambil form yang telah diisi oleh Pimpinan atau kantor Induk dan menyimpan data pada database.</li> <li>6. Sistem memberi notifikasi jika target biaya telah berhasil disimpan.</li> <li>7. Sistem menampilkan halaman utama dalam menu Anggaran.</li> </ol>
Alternative Flow	5a. Jika salah satu dari field tidak diisi maka sistem akan menampilkan peringatan "Harap isi bidang ini" dan kembali pada main flow nomor 2.
Post-Condition	Target biaya telah berhasil disimpan dalam database.

**4.2.5.19 Use Case Scenario Menambah Komentar**

Pada tabel 4.25 menjelaskan *use case scenario* menambah komentar mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.23 Use Case Scenario Menambah Komentar**

<b>DPF_1900 : Menambah Komentar</b>	
Objective	Menambah komentar
Actor	Pimpinan
Pre-Condition	Pimpinan telah berada didalam menu Laporan Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan memilih tanggal bulan, dan tahun yang tersedia untuk memilih laporan produksi yang ingin ditambah komentar.</li> <li>2. Sistem menampilkan data komentar dari database sesuai dengan tanggal bulan dan tahun yang telah di inputkan pengguna.</li> <li>3. Pimpinan mengisi komentar dikolom yang telah disediakan.</li> </ol>

**Tabel 4.23 Use Case Scenario Menambah Komentar**

DPF_1900 : Menambah Komentar	
	<ol style="list-style-type: none"> <li>4. Pimpinan menekan tombol simpan.</li> <li>5. Sistem mengambil form yang telah diisi oleh Pimpinan dan menyimpan data pada database.</li> <li>6. Sistem menampilkan halaman utama dalam menu Laporan Produksi.</li> </ol>
Alternative Flow	5a. Jika kolom komentar tidak diisi maka sistem akan menampilkan peringatan “Harap isi bidang ini” dan kembali pada main flow nomor 2.
Post-Condition	Komentar telah berhasil disimpan dalam database dan komentar berhasil ditampilkan.

**4.2.5.20 Use Case Scenario mengubah Komentar**

Pada tabel 4.26 menjelaskan *use case scenario* mengubah komentar mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.24 Use Case Scenario Mengubah Komentar**

DPF_2000 : Mengubah Komentar	
Objective	Mengubah komentar
Actor	Pimpinan
Pre-Condition	Pimpinan telah berada didalam menu Laporan Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan produksi yang ingin diubah.</li> <li>2. Sistem menampilkan data komentar dari database sesuai dengan tanggal bulan dan tahun yang telah di inputkan pengguna.</li> <li>3. Pimpinan menekan tombol ubah.</li> <li>4. Sistem akan menampilkan form untuk menambah komentar.</li> <li>5. Pimpinan mengisi komentar dikolom yang telah disediakan.</li> <li>6. Pimpinan menekan tombol simpan.</li> <li>7. Sistem mengambil form yang telah diisi oleh Pimpinan dan menyimpan data pada database.</li> <li>8. Sistem menampilkan halaman utama dalam menu Laporan Produksi.</li> </ol>

**Tabel 4.24 Use Case Scenario Mengubah Komentar**

DPF_2000 : Mengubah Komentar	
Alternative Flow	7a. Jika kolom komentar tidak diisi maka sistem akan menampilkan peringatan “Harap isi bidang ini” dan kembali pada main flow nomor 4.
Post-Condition	Komentar telah berhasil diubah, berhasil disimpan dalam database dan komentar berhasil ditampilkan.

**4.2.5.21 Use Case Scenario Menghapus Komentar**

Pada tabel 4.27 menjelaskan *use case scenario* menghapus komentar mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.25 Use Case Scenario Menghapus Komentar**

DPF_2100 : Menghapus Komentar	
Objective	Menghapus komentar
Actor	Pimpinan
Pre-Condition	Pimpinan telah berada didalam menu Laporan Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan produksi yang ingin dihapus.</li> <li>2. Pimpinan menekan tombol Hapus.</li> <li>3. Sistem menghapus data komentar dari database sesuai dengan tanggal bulan dan tahu yang telah di inputkan pengguna</li> <li>4. Sistem menampilkan halaman utama dalam menu Laporan produksi.</li> </ol>
Alternative Flow	3b. Jika komentar tidak tersedia, maka tombol hapus tidak aktif.
Post-Condition	Komentar berhasil dihapus dari database dan komentar berhasil ditampilkan.

**4.2.5.22 Use Case Scenario Melihat Komentar**

Pada tabel 4.28 menjelaskan *use case scenario* melihat komentar mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.



**Tabel 4.26 Use Case Scenario Melihat Komentar**

<b>DPF_2200 : Melihat Komentar</b>	
Objective	Melihat komentar
Actor	Pimpinan
Pre-Condition	Pimpinan telah berada didalam menu Laporan Produksi
Main Flow	<ol style="list-style-type: none"> <li>1. Pimpinan memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan produksi yang ingin ditampilkan.</li> <li>2. Sistem menampilkan data komentar dari database sesuai dengan tanggal bulan dan tahun yang telah di inputkan pengguna.</li> </ol>
Alternative Flow	-
Post-Condition	Komentar berhasil ditampilkan.

**4.2.5.23 Use Case Scenario Menambah Pengguna**

Pada tabel 4.29 menjelaskan *use case scenario* menambah pengguna mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.27 Use Case Scenario Menambah Pengguna**

<b>DPF_2300 : Menambah Pengguna</b>	
Objective	Menambah pengguna
Actor	Admin
Pre-Condition	Admin telah berhasil terautentikasi oleh sistem.
Main Flow	<ol style="list-style-type: none"> <li>1. Admin menekan tombol Tambah pengguna</li> <li>2. Sistem menampilkan form tambah pengguna.</li> <li>3. Admin mengisi data pengguna yang akan ditambahkan berupa NIP/NIK sebagai username, nama, jabatan dan <i>password</i>.</li> <li>4. Admin menekan tombol simpan.</li> <li>5. Sistem mengambil form yang telah diisi oleh Admin dan menyimpan data pada database.</li> <li>6. Sistem menampilkan halaman utama.</li> </ol>
Alternative Flow	5a. Jika salah satu field tidak diisi maka sistem akan menampilkan peringatan "Harap isi bidang ini" dan kembali pada main flow nomor 2.
Post-Condition	Pengguna telah berhasil disimpan dalam database.





#### 4.2.5.24 Use Case Scenario Mengubah Pengguna

Pada tabel 4.30 menjelaskan *use case scenario* mengubah pengguna mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.28 Use Case Scenario Mengubah Pengguna**

DPF_2400 : Mengubah Pengguna	
Objective	Mengubah data pengguna
Actor	Admin
Pre-Condition	Admin telah berhasil terautentikasi oleh sistem.
Main Flow	<ol style="list-style-type: none"> <li>1. Admin menekan tombol Ubah pengguna</li> <li>2. Sistem menampilkan form tambah pengguna.</li> <li>3. Admin mengisi data pengguna yang akan ditambahkan berupa nama, jabatan dan <i>password</i>.</li> <li>4. Admin menekan tombol simpan.</li> <li>5. Sistem mengambil form yang telah diisi oleh Admin dan menyimpan data pada database.</li> <li>6. Sistem menampilkan halaman utama.</li> </ol>
Alternative Flow	5b. Jika salah satu field tidak diisi maka sistem akan menampilkan peringatan “Harap isi bidang ini” dan kembali pada main flow nomor 2.
Post-Condition	Pengguna telah berhasil diubah, disimpan dalam database dan berhasil melihat semua data pengguna yang terdaftar.

#### 4.2.5.25 Use Case Scenario Melihat Pengguna

Pada tabel 4.31 menjelaskan *use case scenario* melihat pengguna mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

**Tabel 4.29 Use Case Scenario Melihat Pengguna**

DPF_2500 : Melihat Pengguna	
Objective	Melihat semua pengguna yang terdaftar
Actor	Admin
Pre-Condition	Admin telah berhasil terautentikasi oleh sistem.
Main Flow	<ol style="list-style-type: none"> <li>1. Sistem mengambil data semua pengguna dari database dan menampilkan.</li> </ol>
Alternative Flow	-

Tabel 4.29 Use Case Scenario Melihat Pengguna

DPF_2500 : Melihat Pengguna	
Post-Condition	Admin telah berhasil melihat semua data pengguna yang terdaftar.

#### 4.2.5.26 Use Case Scenario Menghapus Pengguna

Pada tabel 4.32 menjelaskan *use case scenario* menghapus pengguna mulai dari tujuan, aktor yang melakukan, kondisi awal, alian utama, aliran alternatif, serta kondisi akhir.

Tabel 4.30 Use Case Scenario Menghapus Pengguna

DPF_2600 : Menghapus Pengguna	
Objective	Menghapus data pengguna
Actor	Admin
Pre-Condition	Admin telah berhasil terautentikasi oleh sistem.
Main Flow	<ol style="list-style-type: none"> <li>1. Admin menekan tombol Hapus.</li> <li>2. Sistem menghapus data pengguna dari database.</li> <li>3. Sistem memberi notifikasi jika pengguna telah berhasil dihapus.</li> <li>4. Sistem menampilkan halaman utama.</li> </ol>
Alternative Flow	-
Post-Condition	Pengguna telah berhasil dihapus dari database dan berhasil melihat semua data pengguna yang terdaftar.

### 4.3 Validasi

Proses validasi dilakukan untuk mengecek bahwa sistem telah memenuhi semua kebutuhan ataupun harapan dari calon pengguna. Dalam tahap ini memastikan kebutuhan-kebutuhan yang telah disepakati sudah terpenuhi. Proses validasi pada penelitian ini dilakukan oleh juru tulis, kantor induk dan pimpinan.

#### 4.3.1 Validasi Kebutuhan Fungsional

Validasi kebutuhan fungsional dilakukan dengan cara melakukan wawancara langsung dengan juru tulis, kantor induk dan pimpinan yang merupakan calon pengguna sistem yang akan dibangun. Dari proses wawancara tersebut telah disepakati kebutuhan yang akan tersedia didalam sistem yang akan dibangun dapat dilihat dari tabel 4.4 berikut.

Tabel 4.31 Validasi Kebutuhan Fungsional

Kode	Kebutuhan fungsional	Validasi
DPF_0100	Login	√
DPF_0200	Logout	√
DPF_0300	Menambah Data Produksi	√
DPF_0400	Mengubah Data Produksi	√
DPF_0500	Menghapus Data Produksi	√
DPF_0600	Menambah Data Harian	√
DPF_0700	Mengubah Data Harian	√
DPF_0800	Menghapus Data Harian	√
DPF_0900	Melihat Data Produksi	√
DPF_1000	Melihat Data Harian	√
DPF_1100	Mencetak Laporan Produksi	√
DPF_1200	Mencetak Laporan Harian	√
DPF_1300	Status Porduksi	√
DPF_1400	Melihat Total Realisasi Produksi	√
DPF_1500	Menghitung Estimasi Biaya	√
DPF_1600	Melihat Status Biaya	√
DPF_1700	Menambah Target Produksi	√
DPF_1800	Menambah Target Biaya	√
DPF_1900	Menambah Komentar	√
DPF_2000	Mengubah Komentar	√
DPF_2100	Menghapus Komentar	√
DPF_2200	Melihat Komentar	√
DPF_2300	Menambah Pengguna	√
DPF_2400	Mengubah Pengguna	√
DPF_2500	Melihat Pengguna	√
DPF_2600	Menghapus Pengguna	√

#### 4.3.2 Validasi Kebutuhan *Non* Fungsional

Validasi kebutuhan *non*-fungsional dilakukan dengan cara melakukan wawancara langsung dengan juru tulis, kantor induk dan pimpinan yang merupakan calon pengguna sistem yang akan dibangun. Dari proses wawancara tersebut telah

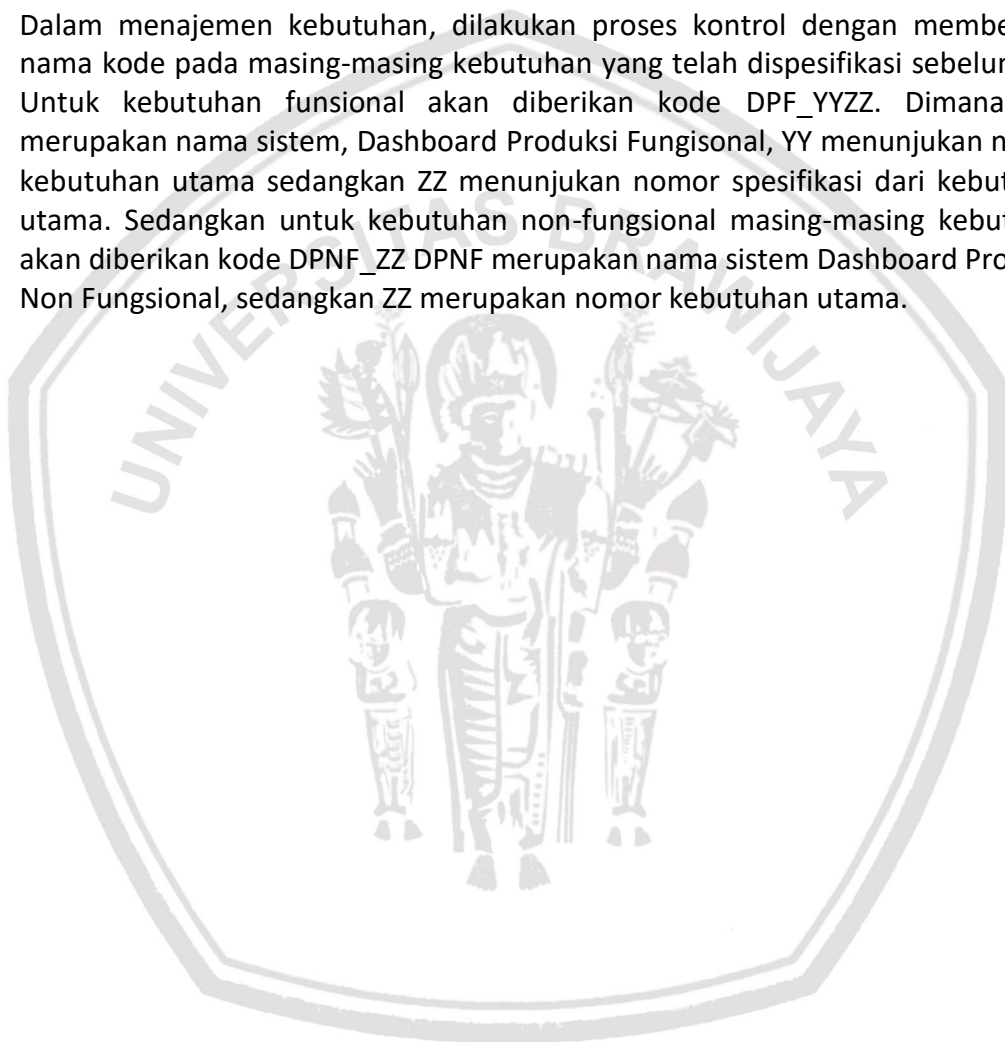
disepakati kebutuhan yang akan tersedia didalam sistem yang akan dibangun dapat dilihat dari tabel 4.7.

**Tabel 4.7 Validasi Kebutuhan *non-Fungsional***

Kode	Kebutuhan fungsional	Validasi
DPNF_01	<i>Compatibility</i>	√

#### 4.4 Manajemen Kebutuhan

Dalam manajemen kebutuhan, dilakukan proses kontrol dengan memberikan nama kode pada masing-masing kebutuhan yang telah dispesifikasi sebelumnya. Untuk kebutuhan fungsional akan diberikan kode DPF\_YYZZ. Dimana DPF merupakan nama sistem, Dashboard Produksi Fungsional, YY menunjukan nomor kebutuhan utama sedangkan ZZ menunjukan nomor spesifikasi dari kebutuhan utama. Sedangkan untuk kebutuhan non-fungsional masing-masing kebutuhan akan diberikan kode DPNF\_ZZ DPNF merupakan nama sistem Dashboard Produksi Non Fungsional, sedangkan ZZ merupakan nomor kebutuhan utama.



## BAB 5 PERANCANGAN DAN IMPLEMENTASI

Perancangan dan implementasi berisi tentang model perangkat lunak yang terdiri dari sequence diagram, class diagram yang mana merupakan perancangan arsitektur, kemudian perancangan komponen, perancangan data dan perancangan UI. Model ini diperlukan agar developer dapat mengembangkan sistem secara lebih terarah sehingga sistem nantinya akan *traceable* serta membantu dalam proses pengembangan selanjutnya.

### 5.1 Perancangan

Perancangan atau bisa disebut pemodelan dilakukan untuk membangun model abstrak dari sebuah sistem, yang masing-masing modelnya memberikan perspektif berbeda terhadap sistem. Proses pemodelan digunakan untuk mempresentasikan sistem yang merepresentasikan *Unified Modeling Language* (UML). Pada proses ini akan dilakukan menjadi beberapa tahap yaitu perancangan arsitektur, perancangan komponen, perancangan data dan perancangan antarmuka.

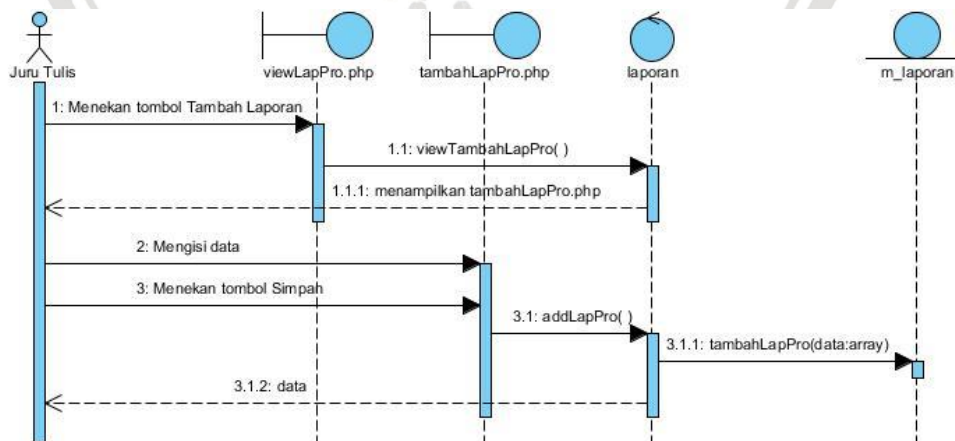
#### 5.1.1 Perancangan Arsitektur

Perancangan arsitektur menjelaskan elemen-elemen utama serta relasi antar elemen struktural utama yang terdapat didalam sistem. Didalam perancangan arsitektur ini nantinya akan dijelaskan dengan menggunakan *sequence* diagram dan *class* diagram.

##### 5.1.1.1 Sequence Diagram

*Sequence* Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu (Nugroho, 2010). *Sequence* diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu.

##### 5.1.1.1.1 Sequence Diagram Menambah Data Produksi

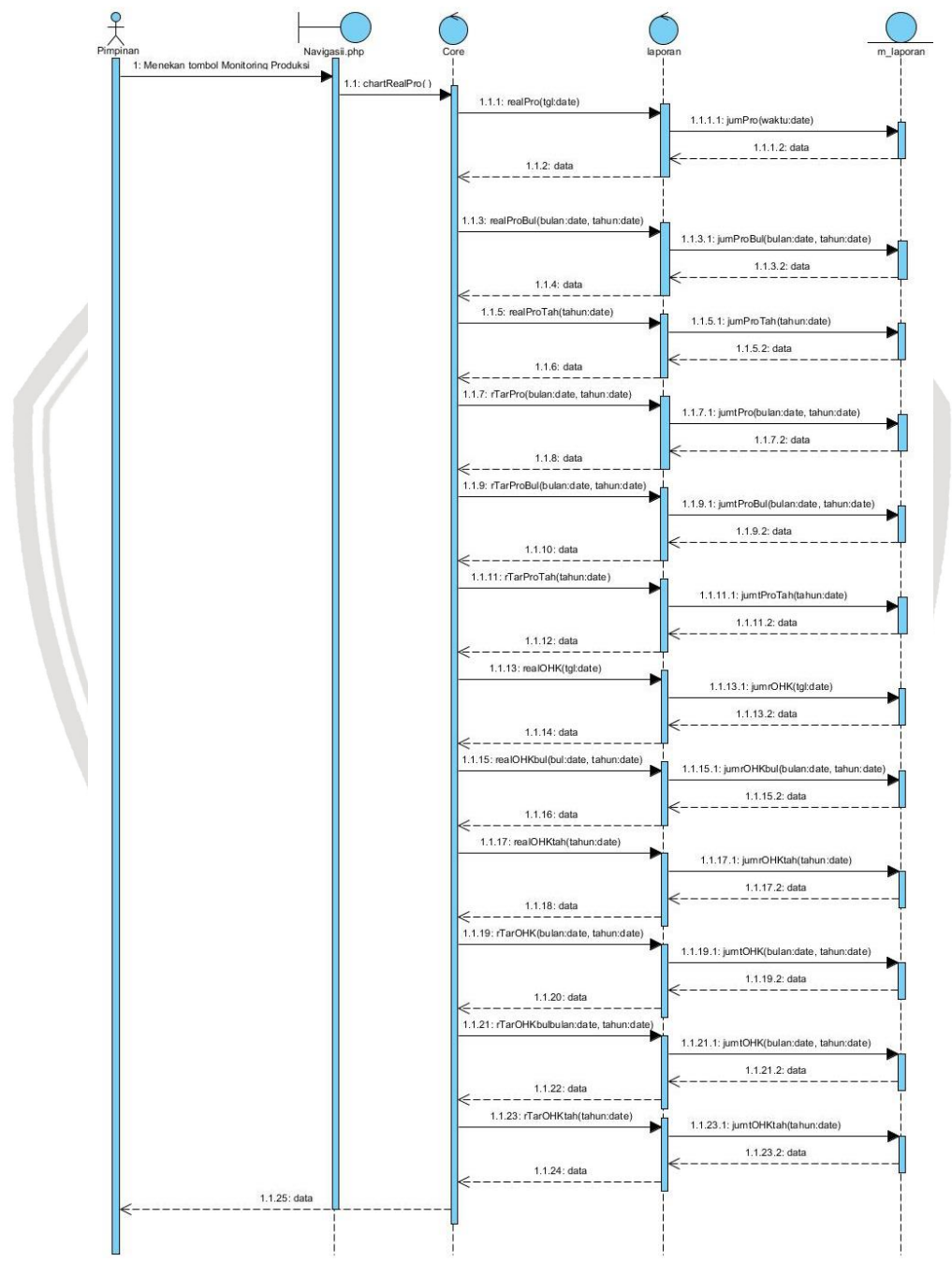


Gambar 5.1 *Sequence Diagram* Menambah Data Produksi



Sequence diagram menambah data produksi menjelaskan tentang proses yang terjadi ketika menambahkan laporan produksi. Proses dimulai ketika juru tulis menekan tombol tambah kemudian sistem akan mengirimkan form tambah laporan produksi melalui operasi viewTambahLApPro( ). Kemudian juru tulis mengisi form yang telah disediakan yaitu tanggal, nama mandor, tahun tanam, realisasi OHK, realisasi produksi. Kemudian juru tulis menekan tombol simpan, kemudian sistem menyimpan data menggunakan operasi addLapPro( ).

5.1.1.1.2 Sequence Diagram Melihat Status Realisasi Produksi

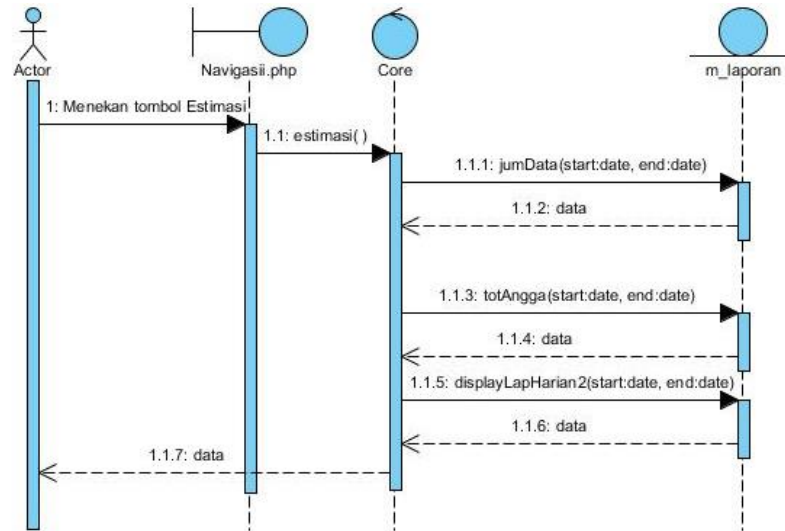


Gambar 5.2 Sequence Diagram Melihat Status Realisasi Produksi



Sequence diagram melihat status realisasi produksi menggambarkan proses yang sedang terjadi untuk mengetahui status realisasi produksi yang terjadi saat ini. Prosesnya dimulai ketika pimpinan menekan tombol Monitoring Produksi yang berada didalam navigasi, kemudian sistem mengambil data realisasi produksi menggunakan operasi chartRealPro berisi parameter bulan untuk menentukan realisasi produksi pada bulan apa yang akan ditampilkan. Kemudian sistem menampilkan data.

**5.1.1.1.3 Sequence Diagram Menghitung Estimasi Biaya**



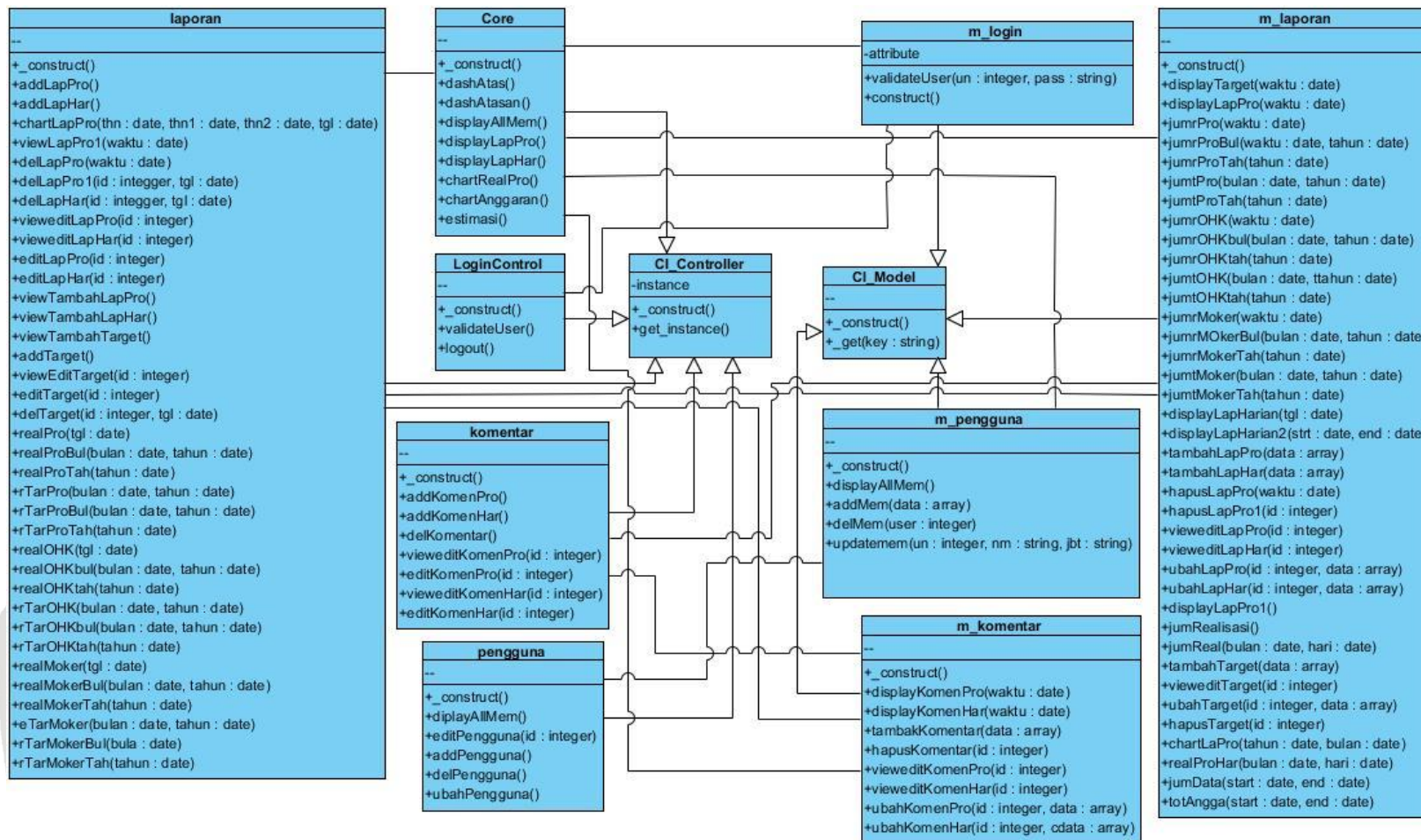
**Gambar 5.3 Sequence Diagram Menghitung Estimasi Biaya**

Sequence diagram menghitung estimasi biaya menggambarkan proses yang terjadi dalam menghitung estimasi biaya yang dikeluarkan untuk memudahkan dalam perencanaan modal kerja yang dilakukan oleh juru tulis. Prosesnya dimulai saat juru tulis menekan tombol Anggaran, kemudian juru tulis memilih bulan dan periode yang akan dihitung estimasi biaya yang dikeluarkan. Kemudian sitem akan mengambil data berdasarkan inputan menggunakan operasi hitungAnggaran dengan parameter periode dan bulan, kemudia menghitung estimasi biaya yang akan dilekuarkan, kemudian menampilkan hasilnya.

**5.1.1.2 Class Diagram**

Gambar 5.4 menjelaskan class diagram yang digunakan oleh penulis dalam mengembangkan sistem. Dalam diagram ini terdapat sebuah controller yang membatasi pengguna dengan sistem. Kemudian terdapat entity yang berfungsi sebagai penghubung antara sistem dengan database. Kelas controller yang terdapat pada sistem ini antara lain "Core", "laporan", "komentar", "LoginControl", dan "pengguna". Sedangkan kelas entity anatar lain "m\_komentar", "m\_laporan", "m\_login", dan "m\_pengguna".





Gambar 5.4 Class Diagram

### 5.1.2 Perancangan Komponen

Perancangan komponen merupakan sebuah kegiatan mengambil setiap komponen yang terdapat di dalam sistem tentang bagaimana sebuah komponen tersebut berjalan (Sommerville, 2011). Model ini nantinya akan mempermudah programmer dalam proses implementasi. Perancangan komponen berisi tentang algoritme-algoritme yang akan di implementasikan pada klas-klas yang akan di definisikan. Algoritma ini dituliskan dalam bentuk *pseudocode* yang menjelaskan alur atau langkah-langkah dalam menyelesaikan permasalahan dalam kode-kode yang ditulis dengan lebih sederhana menggunakan bahasa yang mudah dipahami secara universal. Dalam perancangan komponen Pembangunan Dashboard Produksi PTPN XII, sampel yang diambil yaitu algoritme fungsi melihat data harian, melihat status realisasi anggaran dan menghitung estimasi biaya.

#### 5.1.2.1 Algoritme Melihat Data Harian

Algoritme melihat data harian merupakan algoritme untuk menampilkan laporan harian. Tabel 5.1 akan menjelaskan algoritme melihat laporan harian.

**Tabel 5.1 Algoritme Melihat Data Harian**

<b>Fungsi Melihat Data Harian</b>
Mengambil nilai tanggal berdasarkan inputan
Jika tanggal bernilai kosong
Tanggal bernilai tanggal hari ini dikurangi 1 hari
Mengambil data komentar berdasarkan tanggal
Mengambil data laporan produksi berdasarkan tanggal
Menampilkan data laporan harian

#### 5.1.2.2 Algoritme Melihat Status Realisasi Anggaran

Algoritme melihat status realisasi anggaran merupakan algoritme untuk mengetahui status realisasi anggaran yang terjadi saat ini. Tabel 5.2 akan menjelaskan algoritme melihat status realisasi anggaran.

**Tabel 5.2 Algoritme Melihat Status Realisasi Produksi**

<b>Fungsi Melihat Status Realisasi Produksi</b>
Mengambil nilai tanggal berdasarkan inputan
Jika tanggal bernilai kosong
Tanggal bernilai tanggal hari ini
Bulan bernilai bulan ini
Tahun bernilai tahun ini
Mengambil data realisasi moker pada hari tertentu



Mengambil data realisasi moker pada bulan tertentu  
 Mengambil data realisasi moker pada tahun tertentu  
 Mengambil data target moker pada hari tertentu  
 Mengambil data target moker pada bulan tertentu  
 Mengambil data target moker pada tahun tertentu  
 Jika data realisasi moker atau target moker pada hari tertentu bernilai kosong  
     Status realisasi moker pada hari tertentu bernilai '0'  
 Jika tidak  
     Status realisasi moker bernilai  $\text{realisasi moker} / \text{target moker} * 100$   
 Jika data realisasi moker atau target moker pada bulan tertentu bernilai kosong  
     Status realisasi moker pada hari tertentu bernilai '0'  
 Jika tidak  
     Status realisasi moker bernilai  $\text{realisasi moker} / \text{target moker} * 100$   
 Jika data realisasi moker atau target moker pada tahun tertentu bernilai kosong  
     Status realisasi moker pada hari tertentu bernilai '0'  
 Jika tidak  
     Status realisasi moker bernilai  $\text{realisasi moker} / \text{target moker} * 100$   
 Menampilkan status realisasi moker

### 5.1.2.3 Algoritme Menghitung Estimasi Biaya

Algoritme menghitung estimasi biaya menjelaskan algoritme untuk menghitung estimasi biaya. Table 5.3 akan menjelaskan algoritme menghitung estimasi biaya.

**Tabel 5.3 Algoritme Menghitung Estimasi Biaya**

<b>Fungsi Menghitung Estimasi Biaya</b>
Mengambil nilai bulan
Mengambil nilai periode
Jika bulan bernilai kosong
Bulan bernilai bulan ini
Tanggal start bernilai tanggal 01
Tanggal end bernilai tanggal 10
Jika periode bernilai 0 atau 1
Periode bernilai Pertama
Tanggal start bernilai tanggal 1



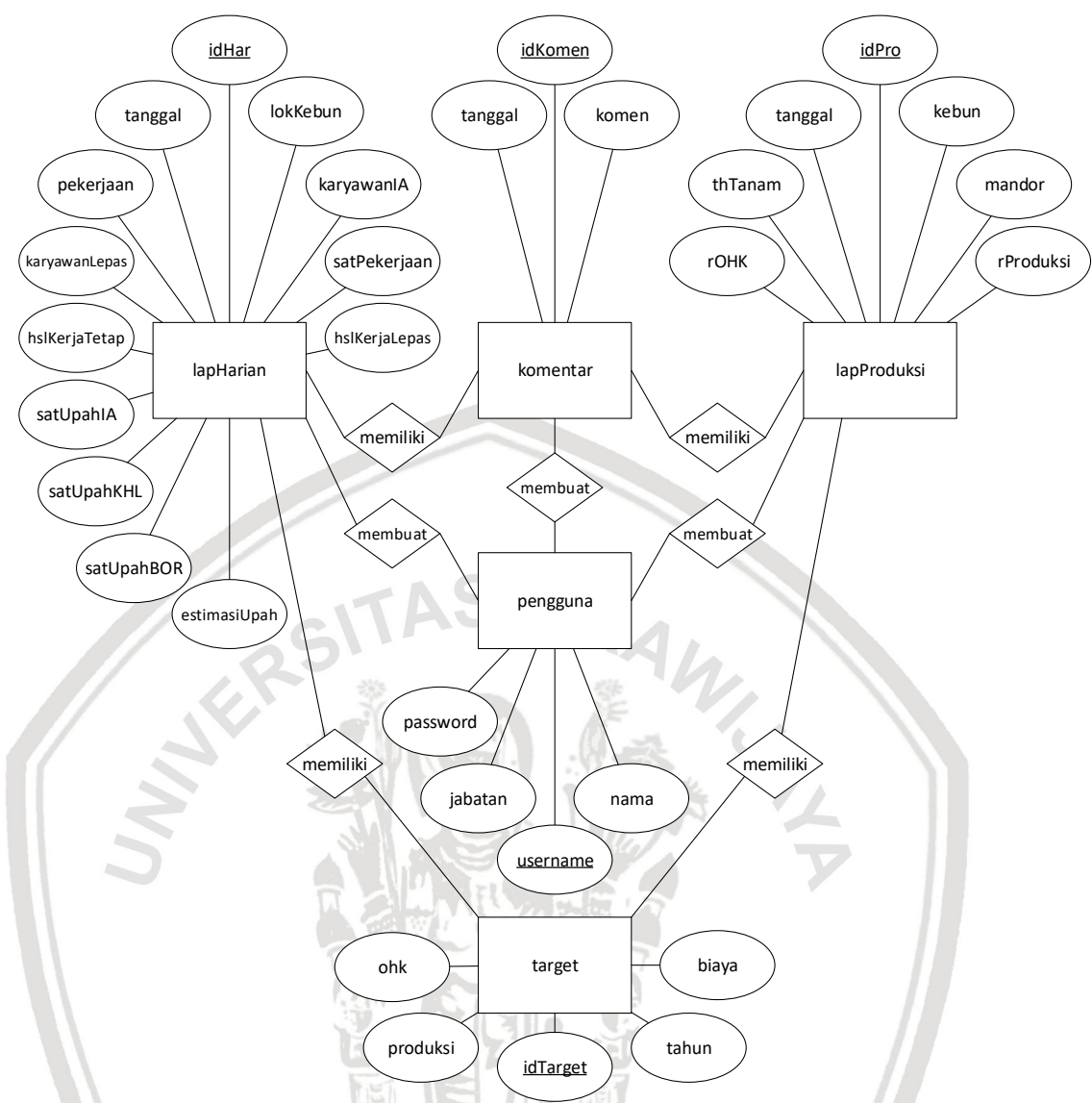
Tanggal end bernilai tanggal 10  
Jika periode bernilai 2  
Periode bernilai Kedua  
Tanggal start bernilai tanggal 6  
Tanggal end bernilai tanggal 25  
Mengambil jumlah data laporan harian dari tanggal start sampai tanggal end  
Jika jumlah data bernilai kosong  
Jumlah data bernilai 1  
Mengambil data laporan harian dari tanggal start sampai tanggal end  
Rata rata anggaran bernilai jumlah data laporan harian / jumlah data  
Estimasi moker bernilai rata rata anggaran \* 15  
Menampilkan estimasi moker

### 5.1.3 Perancangan Data

Perancangan data bertujuan untuk memberikan gambaran tentang bagaimana nantinya basis data yang akan disusun didalam sistem yang akan dibangun. Perancangan ini nantinya juga akan menggambarkan objek-objek beserta atribut apa saja yang diperlukan dalam membangun sistem.

#### 5.1.3.1 Entity Relational Diagram

*Entity Relational Diagram* (ERD) menggambarkan rancangan database secara konseptual. ERD dalam sistem dashboard produksi PTPN XII Gunung Gambir akan dijelaskan pada gambar 5.5. Terdapat lima entity, diantaranya pengguna, lapproduksi, lapharian, komentar, dan target.

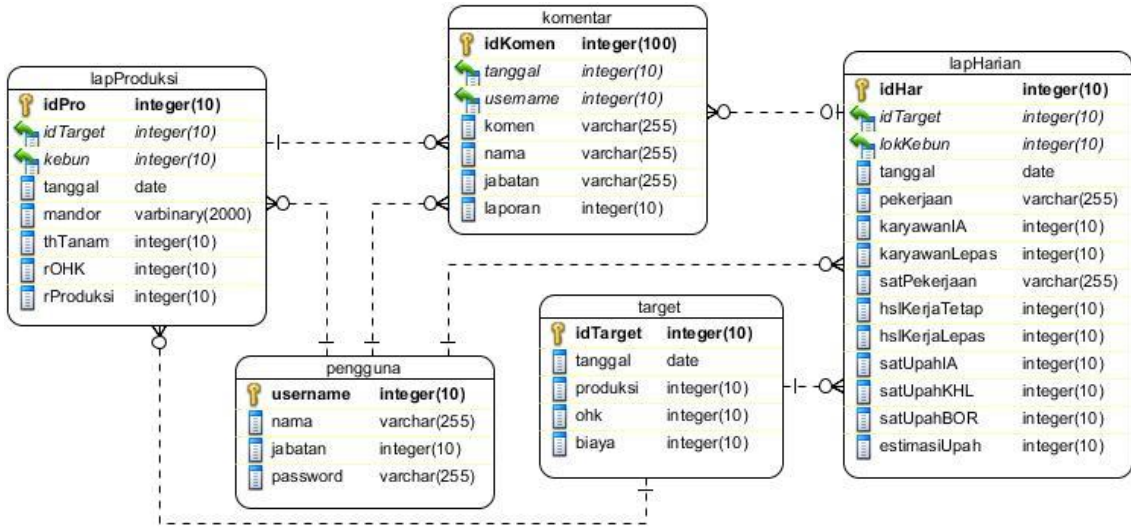


**Gambar 5.5 Entity Relatonal Diagram Dashboard Produksi PTPN XII Gunung Gambir**

**5.1.3.2 Physical Data Model**

*Physical Data Model* (PDM) merupakan gambaran secara detail basis data dalam bentuk fisik. Rancangan PDM memperlihatkan struktur penyimpanan data yang benar pada basis data yang sesungguhnya. Pada perancangan data *physical data model* akan dijelaskan oleh gambar 5.6.





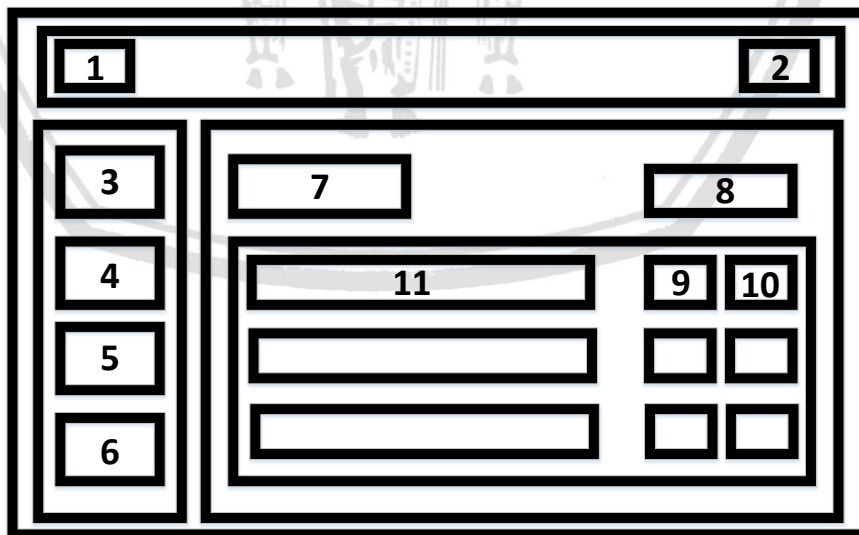
Gambar 5.6 Physical Data Model Dashboard Produksi PTPN XII Gunung Gambir

### 5.1.4 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk menghasilkan antarmuka sistem yang dapat menghasilkan output yang dapat diterima oleh pengguna dengan baik. Output yang dihasilkan antara lain pelaporan realisasi produksi dan harian, status realisasi produksi dan anggaran, serta hasil perhitungan estimasi biaya yang akan dikeluarkan. Berikut adalah rancangan antarmuka yang terdapat dalam Dashboard Produksi PTPN XII Gunung Gambir.

#### 5.1.4.1 Antarmuka Halaman Lihat Data Produksi

Halaman ini merupakan halaman untuk melihat laporan produksi yang telah dibuat oleh juru tulis. Laporan produksi akan dikelompokkan berdasarkan bulan dan ditampilkan berdasarkan hari. Rancangan antarmuka halaman lihat laporan produksi dapat dilihat pada gambar 5.7.



Gambar 5.7 Antarmuka Halaman Lihat Data Produksi

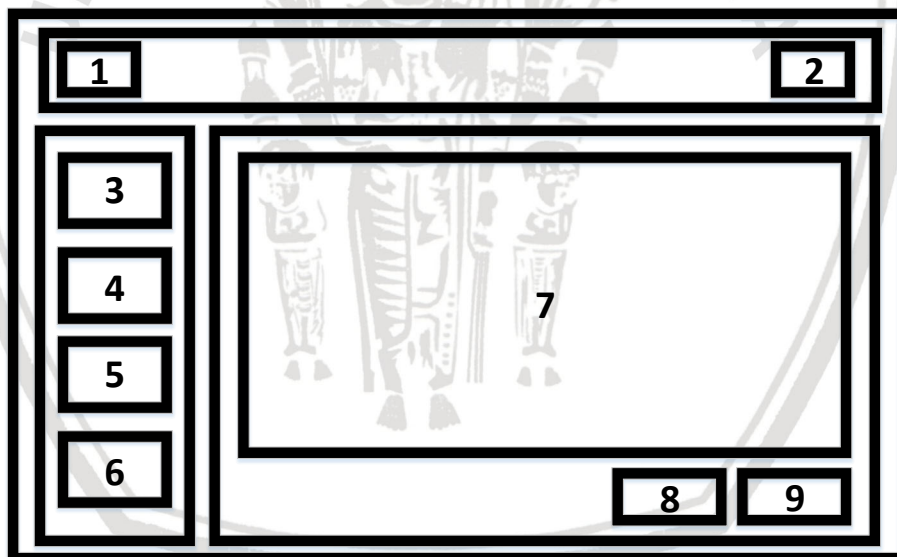


Keterangan :

1. Logo perusahaan
2. Tombol Logout
3. Tombol Dashboard
4. Tombol Laporan
5. Tombol Monitoring Produksi
6. Tombol Anggaran
7. Tombol filter bulan yang akan ditampilkan produksinya
8. Tombol Tambah Laporan
9. Tombol Lihat Detail
10. Tombol Hapus
11. Keterangan produksi yang diperoleh dalam bentuk tabel

#### 5.1.4.2 Antarmuka Halaman Tambah Data Produksi

Halaman ini merupakan halaman untuk menambah laporan produksi. Rancangan antarmuka halaman tambah laporan produksi dapat dilihat pada gambar 5.8.



Gambar 5.8 Antarmuka Halaman Tambah Data Produksi

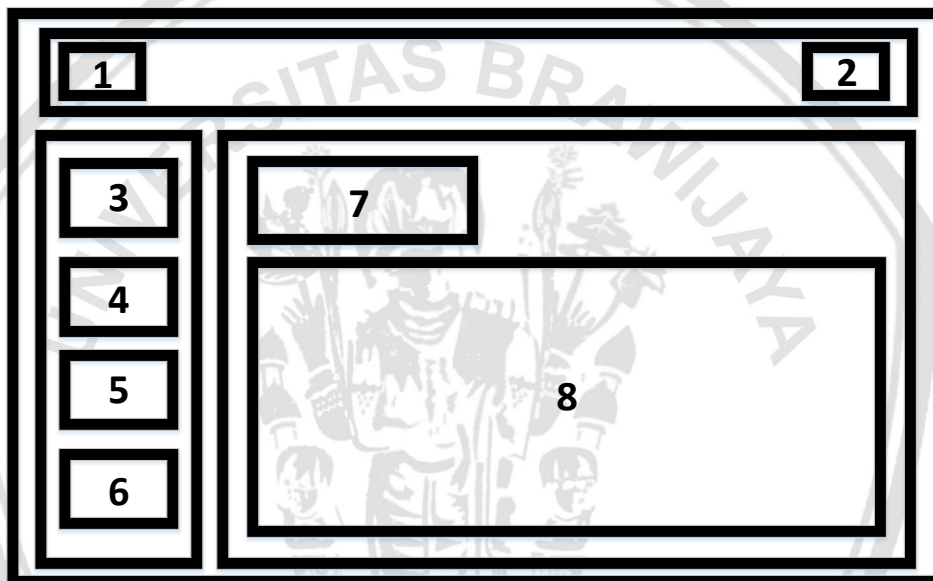
Keterangan :

1. Logo perusahaan
2. Tombol Logout
3. Tombol Dashboard
4. Tombol Laporan

5. Tombol Monitoring Produksi
6. Tombol Anggaran
7. Form tambah laporan produksi yang harus diisi
8. Tombol Tambah Laporan
9. Tombol Batal

#### 5.1.4.3 Antarmuka Halaman Status Ralisasi Produksi

Halaman ini merupakan halaman untuk melihat status realisasi produksi pada bulan tertentu. Informasi yang ditampilkan berupa chart atau diagram beserta informasi tentang realisasi produksi dan target produksi laporan produksi. Rancangan antarmuka halaman status realisasi produksi dapat dilihat pada gambar 5.9.



Gambar 5.9 Antarmuka Halaman Status Ralisasi Produksi

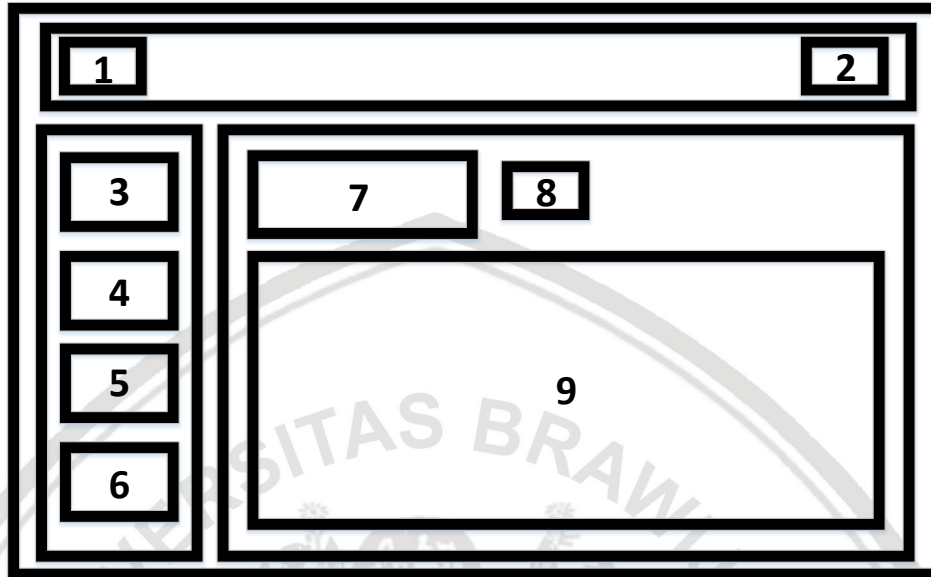
Keterangan :

1. Logo perusahaan
2. Tombol Logout
3. Tombol Dashboard
4. Tombol Laporan
5. Tombol Monitoring Produksi
6. Tombol Anggaran
7. Tombol filtering Bulan yang akan ditampilkan status realisasi produksinya
8. Informasi status realisasi produksi



#### 5.1.4.4 Antarmuka Halaman Menghitung Estimasi Biaya

Halaman ini merupakan halaman untuk menghitung estimasi biaya pada bulan dan periode tertentu. Rancangan antarmuka halaman menghitung estimasi biaya dapat dilihat pada gambar 5.10.



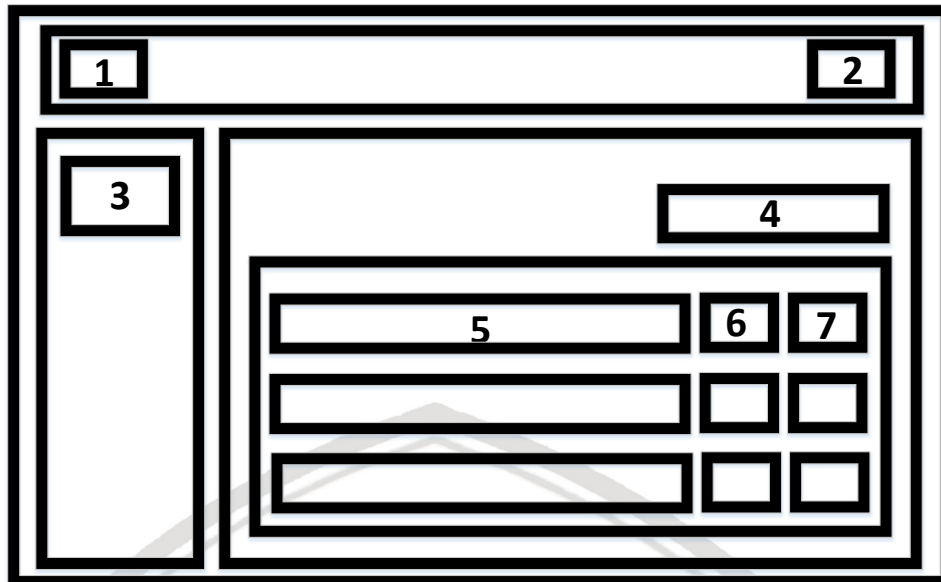
**Gambar 5.10 Antarmuka Halaman Menghitung Estimasi Biaya**

Keterangan :

1. Logo perusahaan
2. Tombol Logout
3. Tombol Dashboard
4. Tombol Laporan
5. Tombol Monitoring Produksi
6. Tombol Anggaran
7. Tombol filtering Bulan yang akan dihitung estimasi biayanya
8. Tombol Hitung
9. Informasi hasil estimasi biaya yang telah dihitung

#### 5.1.4.5 Antarmuka Halaman Lihat Pengguna

Halaman ini merupakan halaman untuk melihat daftar pengguna yang dapat masuk kedalam sistem. Halaman ini hanya dapat diakses oleh admin. Rancangan antarmuka halaman lihat pengguna dapat dilihat pada gambar 5.11.



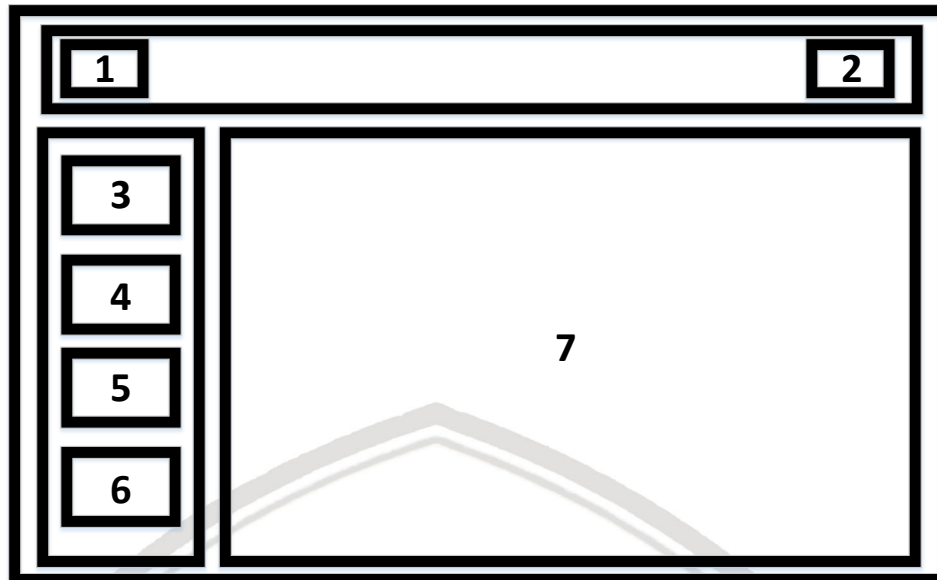
**Gambar 5.11 Antarmuka Halaman Lihat Pengguna**

Keterangan :

1. Logo perusahaan
2. Tombol Logout
3. Tombol Pengguna
4. Tombol Tambah Pengguna
5. Detail informasi pengguna dalam bentuk tabel
6. Tombol Edit
7. Tombol Hapus

#### **5.1.4.6 Antarmuka Halaman Home**

Halaman ini merupakan halaman untuk home. Halaman home merupakan halaman yang pertama kali ditampilkan ketika pengguna berhasil masuk ke dalam sistem. Di dalam halaman home pengguna disajikan beberapa data dalam bentuk grafik.



Gambar 5.12 Antarmuka Halaman Home

Keterangan :

1. Logo perusahaan
2. Tombol Logout
3. Tombol Dashboard
4. Tombol Laporan
5. Tombol Monitoring Produksi
6. Tombol Anggaran
7. Grafik tentang produksi

## 5.2 Implementasi

Pada tahap ini memaparkan tahapan implementasi *Dasboard* Produksi PTPN XII Gunung Gambir yang mengacu dari hasil analisis kebutuhan serta perancangan yang telah dimodelkan sebelumnya. Pembahasan pada sub bab ini terdiri dari spesifikasi sistem, batasan implementasi, implementasi basis data, implementasi komponen, dan implementasi antarmuka.

### 5.2.1 Spesifikasi Sistem

Pada tahapan ini akan memaparkan 2 sub bab yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak. Berikut adalah spesifikasi perangkat keras dan spesifikasi perangkat lunak.

#### 5.2.1.1 Spesifikasi Perangkat Keras

Pembangunan Dashboard Produksi PTPN XII Gunung Gambir menggunakan spesifikasi perangkat keras yang dapat dilihat pada tabel 5.4 berikut.

Tabel 5.4 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Processor	Intel Core i5
Memori (RAM)	8 GB
Kartu Grafis	Nvidia Geforce 930M
Harddisk	640GB
Monitor	14.0" Resolusi 1366 x 768

### 5.2.1.2 Spesifikasi Perangkat Lunak

Pembangunan Dashboard Produksi PTPN XII Gunung Gambir menggunakan spesifikasi perangkat lunak yang dapat dilihat pada tabel 5.5 berikut.

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 10 Pro
Browser	Google Chrome versi 66.0.3359.181
Code Editor	Sublime Text 3
Basis Data	MySQL
Pemodelan, UML	Microsoft Visio 2013, Visual Paradigm 10.0
Framework	Code Igniter, Bootstrap
Dokumentasi	Microsoft Word 2013

### 5.2.2 Batasan Implementasi

Beberapa batasan implementasi Dashboard Produksi PTPN XII Gunung Gambir diantaranya :

1. Pembangunan Dashboard Produksi PTPN XII Gunung Gambir ini dirancang dan dibangun menggunakan bahasa pemrograman PHP, *framework* Code Igniter dan Bootstrap dalam pembangunan *website*, serta menggunakan basis data MySQL.
2. Dashboard Produksi PTPN XII Gunung Gambir memiliki fitur utama berupa Pengelolaan laporan produksi, pengelolaan laporan harian, menghitung estimasi biaya, menampilkan status realisasi produksi terhadap target produksi, menampilkan realisasi anggaran terhadap target anggaran, serta menampilkan diagram realisasi produksi untuk memudahkan proses monitoring produksi.
3. Data yang digunakan dalam Dashboard Produksi PTPN XII Gunung Gambir diambil dari PTPN XII Persero kantor wilayah III, Gunung Gambir kebun / afdeling karang anom.





dimodelkan sebelumnya dengan memperhatikan algoritme-algoritme utama yang berada didalam sistem yaitu menambah laporan produksi, melihat status realisasi produksi, dan menghitung estimasi biaya.

### 5.2.4.1 Implementasi Program Melihat Laporan Harian

Melihat laporan harian merupakan kegiatan yang dilakukan oleh juru tulis untuk melihat laporan harian dimana juru tulis dapat memilih tanggal tertentu yang ingin ditampilkan laporan hariannya. Implementasi program melihat laporan harian dapat dilihat pada tabel 5.5 berikut.

**Tabel 5.5 Implementasi Program Melihat Data Harian**

Implementasi Program : Melihat Data Harian	
1	public function displayLapHar ()
2	{
3	\$tgl = \$this->input->post('date');
4	if (\$tgl == null)
5	{
6	\$tgl = date('d-m-Y', strtotime("-1 day", strtotime(date("d-m- 7 Y"))));
8	}
9	\$data['tgl'] = \$tgl;
10	\$data['komentar'] = \$this->Dbkomentar->displayKomenHar(\$tgl);
11	\$data['hasil'] = \$this->Dblaporan->displayLapHarian(\$tgl);
12	\$data['ha'] = '3';
13	\$data['modules'] = 'viewLapHar';
14	\$this->load->view('appAtasan', \$data);
15	}

### 5.2.4.2 Implementasi Program Melihat Status Realisasi Anggaran

Melihat status realisasi merupakan kegiatan yang dilakukan untuk mengetahui status realisasi anggaran pada tanggal atau bulan tertentu. Implementasi program melihat status realisasi anggaran dapat dilihat pada tabel 5.6 berikut.

**Tabel 5.6 Implementasi Program Melihat Status Realisasi Anggaran**

Implementasi Program : Melihat Status Realisasi Produksi	
1	public function chartAnggaran()
2	{
3	\$tgl = \$this->input->post('date');
4	if (\$tgl == '0')
5	{
6	\$tgl = date('d-m-Y');
7	}
8	\$waktu = new DateTime(\$tgl);
9	\$tahun = \$waktu->format('Y');
10	\$bulan = \$waktu->format('m');
11	
12	\$realMoker = \$this->lap->realMoker(\$tgl);
13	\$realMokerBul = \$this->lap->realMokerBul(\$bulan, \$tahun);
14	\$realMokerTah = \$this->lap->realMokerTah(\$tahun);
15	
16	\$rTarMoker = \$this->lap->rTarMoker(\$bulan, \$tahun);
17	\$rTarMokerBul = \$this->lap->rTarMokerBul(\$bulan, \$tahun);
18	\$rTarMokerTah = \$this->lap->rTarMokerTah(\$tahun);
19	
20	if (\$realMoker == null or \$rTarMoker == null)
21	{

```

22     $data['pMoker']      = '0';
23 }
24 else
25 {
26     $data['pMoker']      = round ($realMoker / $rTarMoker * 100);
27 }
28
29 if ($realMokerBul == null or $rTarMokerBul == null)
30 {
31     $data['pMokerBul']   = '0';
32 }
33 else
34 {
35     $data['pMokerBul']   = $realMokerBul / $rTarMokerBul * 100;
36 }
37
38 if ($realMokerTah == null or $rTarMokerTah == null)
39 {
40     $data['pMokerTah']   = '0';
41 }
42 else
43 {
44     $data['pMokerTah']   = round ($realMokerTah / $rTarMokerTah * 100);
45 }
46
47 $data['tgl']           = $tgl;
48 $data['bulan']        = $bulan;
49 $data['tahun']        = $tahun;
50 $data['ha']           = '5';
51 $data['modules']      = 'viewStatusAngga';
52 $this->load->view('appAtasan', $data);
53 }

```

### 5.2.4.3 Implementasi Program Menghitung Estimasi Biaya

Menghitung estimasi biaya merupakan kegiatan yang dilakukan oleh juru tulis untuk menghitung perkiraan biaya yang akan dikeluarkan dimana juru tulis meliuh bulan dan periode yang akan dihitung estimasinya pada field telah disediakan. Implementasi program menghitung estimasi biaya dapat dilihat pada tabel 5.7 berikut.

**Tabel 5.7 Implementasi Program Menghitung Estimasi Biaya**

Implementasi Program : Menghitung Estimasi Biaya	
1	public function estimasi()
2	{
3	\$bulan = \$this->input->post('date');
4	\$periode = \$this->input->post('periode');
5	
6	if (\$bulan == null)
7	{
8	\$bulan = date('m-Y');
9	\$tglStart = '01'."-".\$bulan;
10	\$tglEnd = '10'."-".\$bulan;
11	}
12	
13	if (\$periode == '0' or \$periode == '1')
14	{
15	\$data['periode'] = 'Pertama';
16	\$tglStart = '01'."-".\$bulan;
17	\$tglEnd = '10'."-".\$bulan;
18	}
19	else



```

20 {
21     $data['periode'] = 'Kedua';
22     $tglStart = '16'."-".$bulan;
23     $tglEnd   = '25'."-".$bulan;
24 }
25
26 $jumData    = $this->Dblaporan->jumData($tglStart, $tglEnd);
27 if ($jumData == null)
28 {
29     $jumData = 1;
30 }
31
32 $totAngga   = $this->Dblaporan->totAngga($tglStart, $tglEnd);
33 $ratAngga   = $totAngga / $jumData;
34 $estMoker   = 15 * $ratAngga;
35
36 $data['bulan']      = $bulan;
37 $data['totAngga']   = $totAngga;
38 $data['ratAngga']   = $ratAngga;
39 $data['estMoker']   = $estMoker;
40 $data['hasil']      = $this->Dblaporan-
41 >displayLapHarian2($tglStart, $tglEnd);
42 $data['ha']         = '7';
43 $data['modules']    = 'viewEstimasi';
44 $this->load->view('appAtasan', $data);
45 }

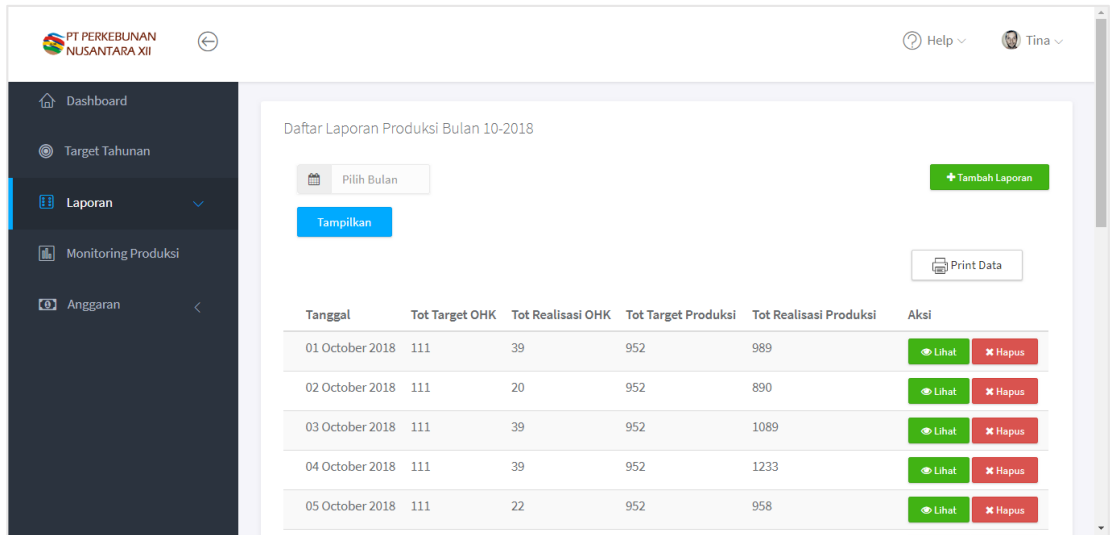
```

## 5.2.5 Implementasi Antarmuka

Implementasi antarmuka merupakan proses pembuatan antarmuka sistem berdasarkan rancangan antarmuka yang telah dimodelkan sebelumnya. Berikut adalah implementasi antarmuka.

### 5.2.5.1 Implementasi Antarmuka Lihat Data Produksi

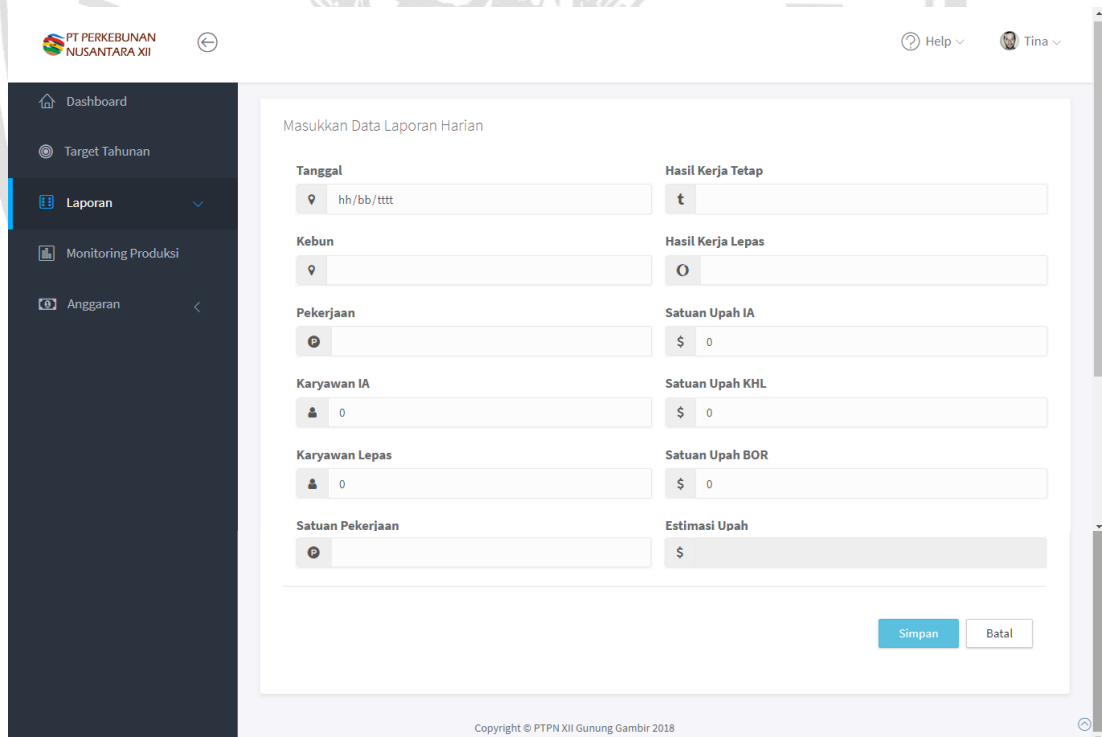
Antarmuka ini merupakan halaman untuk melihat laporan produksi yang telah dibuat oleh juru tulis. Laporan produksi akan dikelompokkan berdasarkan bulan dan ditampilkan berdasarkan hari. Dimana pengguna dapat mengolah laporan produksi seperti menambah laporan produksi, menghapus laporan produksi atau mengubah laporan produksi. Implementasi antarmuka halaman lihat laporan produksi dapat dilihat pada gambar berikut.



Gambar 5.14 Implementasi Antarmuka Lihat Data Produksi

### 5.2.5.2 Implementasi Antarmuka Tambah Data Harian

Antarmuka ini merupakan halaman untuk menambah laporan produksi. Dimana pengguna dapat mengolah laporan harian seperti menambah laporan harian, menghapus laporan harian atau mengubah laporan harian. implementasi antarmuka halaman tambah laporan produksi dapat dilihat pada gambar 5.15

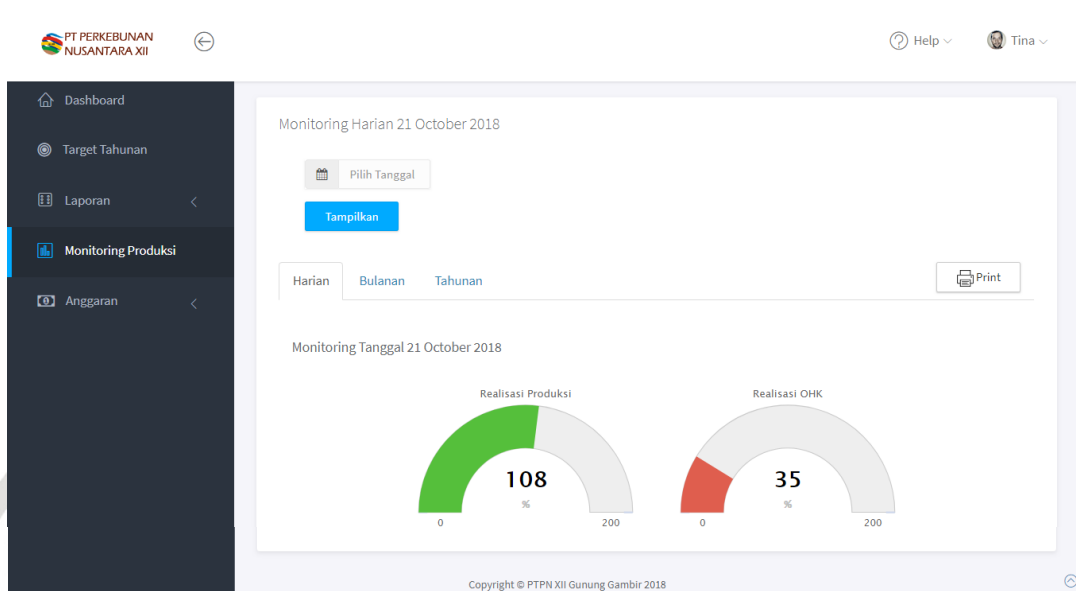


Gambar 5.15 Implementasi Antarmuka Tambah Data Harian



### 5.2.5.3 Implementasi Antarmuka Status Realisasi Produksi

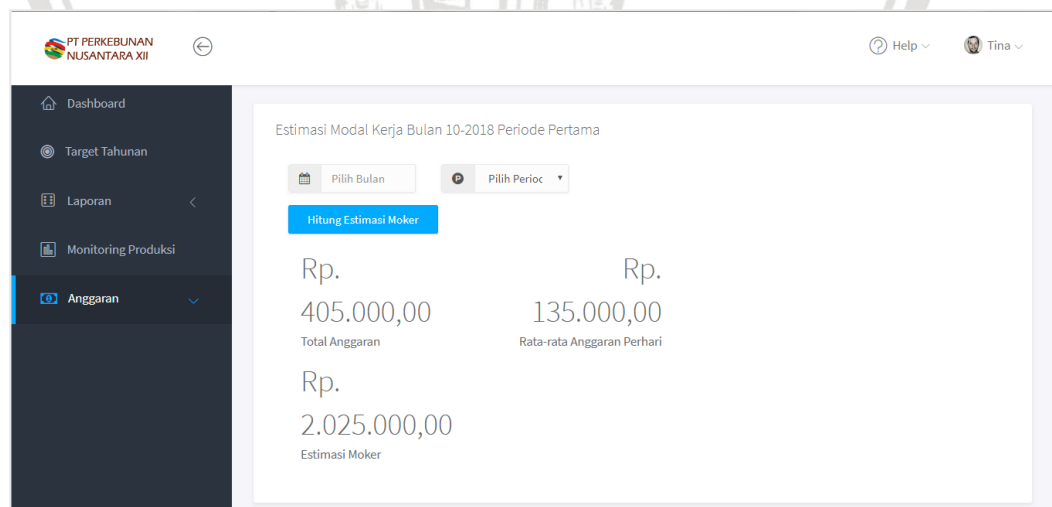
Antarmuka ini merupakan halaman untuk melihat status realisasi produksi pada hari, bulan dan tahun tertentu. Informasi yang ditampilkan berupa chart atau diagram beserta informasi tentang realisasi produksi terhadap target produksi sesuai dengan laporan produksi. Implementasi antarmuka halaman status realisasi produksi dapat dilihat pada gambar 5.16.



Gambar 5.16 Implementasi Antarmuka Status Realisasi Produksi

### 5.2.5.4 Implementasi Antarmuka Menghitung Estimasi Biaya

Antarmuka ini merupakan halaman untuk menghitung estimasi biaya pada bulan dan periode tertentu. Implementasi antarmuka halaman menghitung estimasi biaya dapat dilihat pada gambar 5.17.

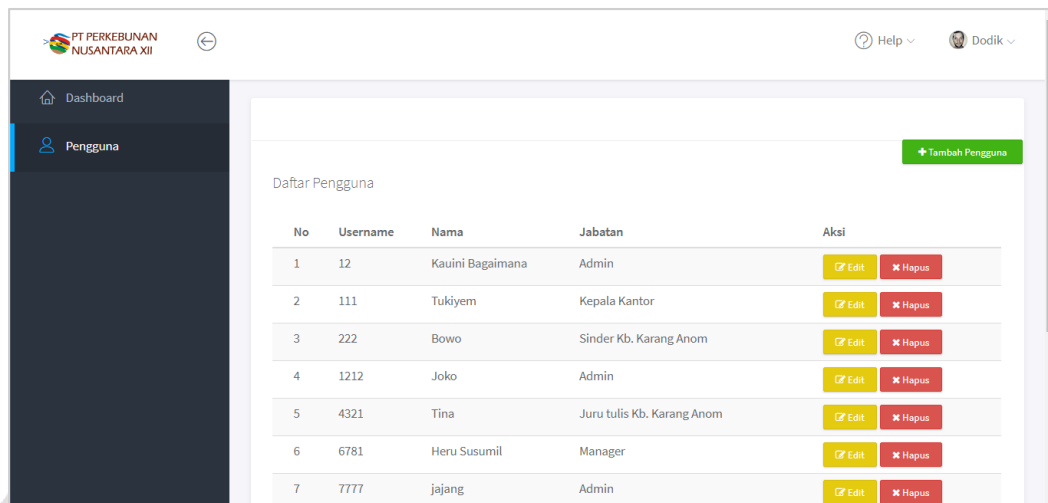


Gambar 5.17 Implementasi Antarmuka Menghitung Estimasi Biaya



### 5.2.5.5 Implementasi Antarmuka Lihat Pengguna

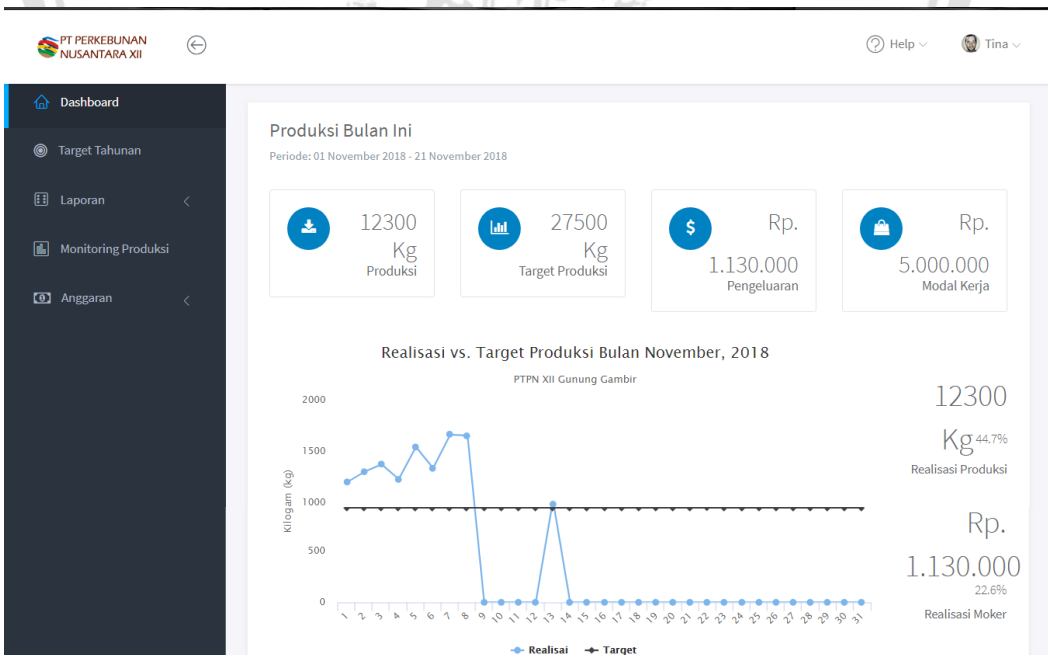
Antarmuka ini merupakan halaman untuk melihat daftar pengguna yang dapat masuk kedalam sistem. Halaman ini hanya dapat diakses oleh admin. Didalam halaman ini admin dapat mengelola pengguna seperti menghapus pengguna, menambah pengguna, atau mengubah pengguna. Implementasi antarmuka halaman lihat pengguna dapat dilihat pada gambar 5.18 berikut.



Gambar 5.18 Implementasi Antarmuka Lihat Pengguna

### 5.2.5.6 Implementasi Antarmuka Home

Antarmuka ini merupakan halaman utama sistem. Terdapat beberapa informasi penting yang ditampilkan berupa grafik. Implementasi antarmuka halaman home dapat dilihat pada gambar 5.19 berikut.



Gambar 5.19 Implementasi Antarmuka Lihat Pengguna

## BAB 6 PENGUJIAN

Pengujian dilakukan setelah proses implementasi telah selesai dilakukan. Pada tahap ini akan dilakukan pengecekan kembali apakah sistem yang telah diimplementasi telah berjalan sesuai dengan analisis kebutuhan dan perancangan yang sebelumnya telah didefinisikan. Dalam tahap ini akan dilakukan pengujian unit, pengujian integrasi, dan pengujian validasi.

### 6.1 Pengujian Unit

Pengujian unit merupakan sebuah pengujian yang dilakukan untuk mengetahui apakah unit atau komponen dari suatu perangkat yang diimplementasikan telah sesuai dengan tujuan dari fungsionalitas itu sendiri. Pengujian ini menerapkan metode *whitebox testing*, dengan menggunakan teknik pengujian *basis path testing*. Pada pengujian unit ini akan dilakukan kepada tiga sampel uji yakni tiga operasi Melihat Data Harian, Melihat Status Realisasi Anggaran dan Menghitung Estimasi Biaya.

#### 6.1.1 Pengujian Unit Melihat Data Harian

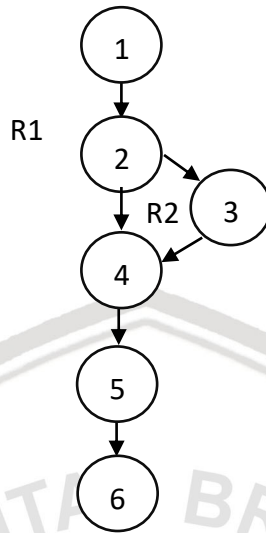
Berikut adalah algoritme dari operasi melihat data harian.

**Tabel 6.1 Algoritme Melihat Data Harian**

No	Pseudocode
1.	<i>tgl = nilai inputan post date</i>
2.	<i>Jika tgl bernilai 0</i>
3.	<i>tgl = bernilai tanggal hari ini dikurangi satu hari.</i>
4.	<i>end If</i>
5.	<i>array tgl = tgl</i> <i>array hasil = hasil query pengambilan data laporan harian dengan parameter tgl</i> <i>array ha = 3</i> <i>array modules = viewLapHar</i>
6.	<i>Load view appAtasan dengan membawa data array</i>

a. *Flow graph*

Berikut adalah *Flow graph* dari algoritme melihat data harian.



Gambar 6.1 Flow Graph fungsi Melihat Data Harian

b. *Cyclomatic Complexity*

Berikut merupakan perhitungan *Cyclomatic Complexity* dari fungsi menambah data produksi :

- 1).  $V(G) = \text{Total ruang (R)}. \text{Total R} = 2$
- 2).  $V(G) = \text{Jumlah edge} - \text{Jumlah node} + 2 = 6 - 6 + 2 = 2$
- 3).  $V(G) = \text{Total Percabangan (P)} + 1 = 1 + 1 = 2$

c. *Independent Path*

Berikut adalah jalur-jalur *independent* dari fungsi melihat data harian :

- 1). Jalur 1 = 1 - 2 - 4 - 5 - 6
- 2). Jalur 2 = 1 - 2 - 3 - 4 - 5 - 6

d. Kasus Uji dan Hasil Pengujian

Berikut merupakan kasus uji dan hasil uji dari fungsi melihat data harian berdasarkan jalur *independent* yang dihasilkan :

Tabel 6.2 Test Case Melihat Data Harian

No.	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Variabel tgl = "0". Variabel tgl bernilai "2018-07-17", array tgl = "2018-07-17", array	Berhasil meload view appAtasan serta	Berhasil meload view appAtasan serta	Valid



Tabel 6.2 Test Case Melihat Data Harian

No.	No Jalur	Prosedur Uji	Expected Result	Result	Status
		hasil = array lapHarian, array ha = "3", array modules = viewLapHar.	menampilkan data laporan harian.	menampilkan data laporan harian.	
2.	2	Variabel tgl bernilai "2018-07-17", array tgl = "2018-07-17", array hasil = array lapHarian, array ha = "3", array modules = viewLapHar.	Berhasil meload view appAtasan serta menampilkan data laporan harian.	Berhasil meload view appAtasan serta menampilkan data laporan harian.	Valid

### 6.1.2 Pengujian Unit Melihat Status Realisasi Anggaran

e. Pseudocode

Berikut adalah algoritme dari operasi melihat status realisasi anggaran.

Tabel 6.3 Algoritme Melihat Status Realisasi Anggaran

No	Pseudocode
1.	<i>tgl = nilai inputan post date</i>
2.	<i>Jika tgl bernilai null</i>
3.	<i>tgl = bernilai tanggal hari ini dikurangi satu hari.</i>
4.	<i>waktu = tgl bulan = bulan pada tgl tahun = tahun pada tgl realMoker = hasil query pengambilan data moker dengan parameter tgl realMokerBul = hasil query pengambilan data moker dengan parameter bulan realMokerTah = hasil query pengambilan data moker dengan parameter tahun rTarMoker = hasil query pengambilan data target moker dengan parameter tgl rTarMokerBul = hasil query pengambilan data target moker dengan parameter bulan rTarMokerTah = hasil query pengambilan data target moker dengan parameter tahun</i>
5.	<i>end if</i>

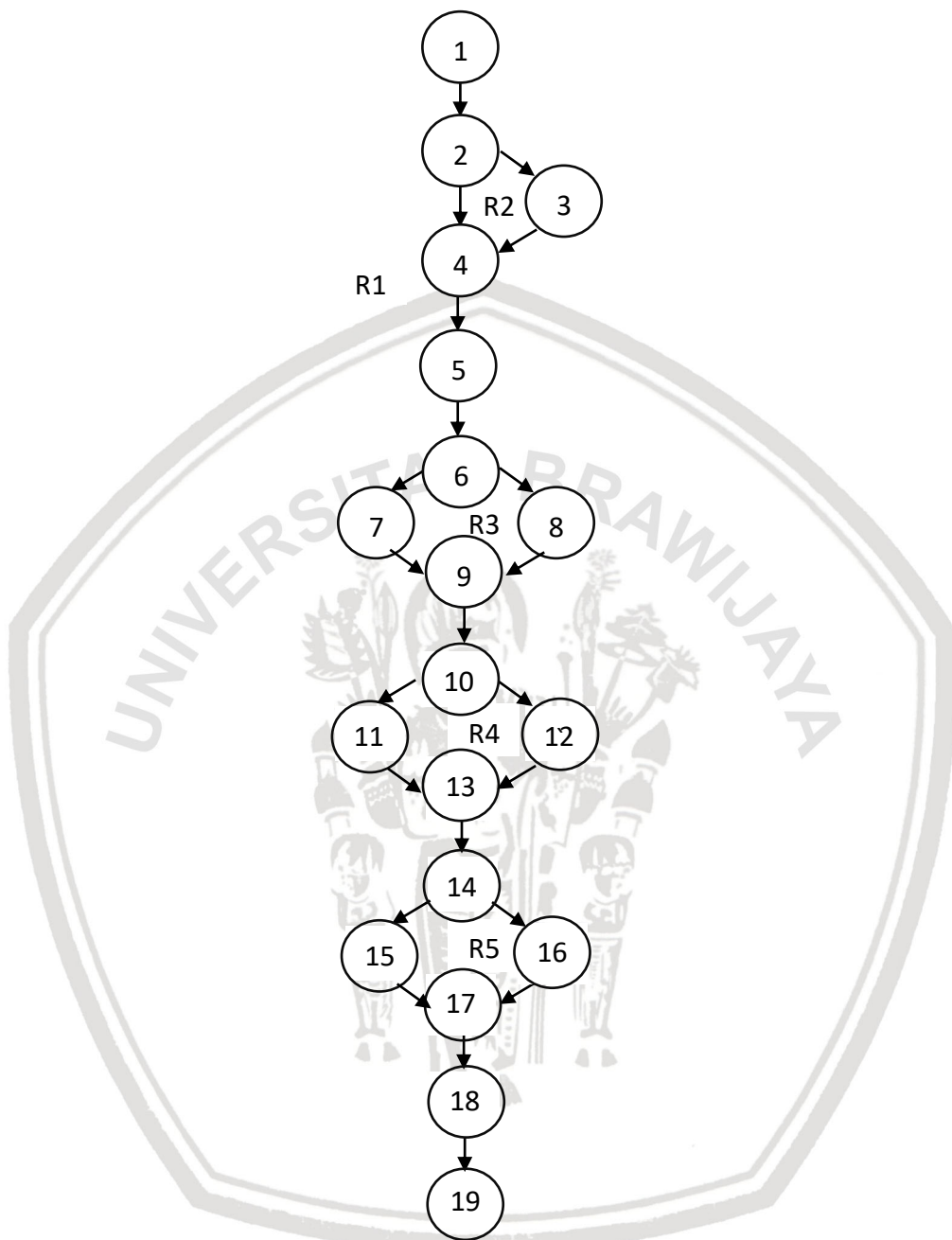
Tabel 6.3 Algoritme Melihat Status Realisasi Anggaran

No	Pseudocode
6.	<i>Jika realMoker atau rTarMoker = null</i>
7.	<i>array pMoker = '0'</i>
8.	<i>else</i> <i>array pMoker = realMoker / rTarMoker * 100</i>
9.	<i>end if</i>
10.	<i>Jika realMokerBul atau rTarMokerBul = null</i>
11.	<i>array pMokerBul = '0'</i>
12.	<i>else</i> <i>array pMokerBul = realMokerBul / rTarMokerBul * 100</i>
13.	<i>end if</i>
14.	<i>Jika realMokerTah atau rTarMokerTah = null</i>
15.	<i>array pMokerTah = '0'</i>
16.	<i>else</i> <i>array pMokerTah = realMokerTah / rTarMokerTah * 100</i>
17.	<i>array tgl = tgl</i> <i>array bulan = bulan</i> <i>array tahun = tahun</i> <i>array ha = 3</i> <i>array modules = viewStatusAngga</i>
18.	<i>end if</i>
19.	<i>Load view appAtasan dengan membawa data array</i>



f. *Flow graph*

Berikut adalah *Flow graph* dari algoritme melihat laporan harian.



**Gambar 6.2 Flow Graph fungsi Melihat Status Realisasi Anggaran**

g. *Cyclomatic Complexity*

Berikut merupakan perhitungan *Cyclomatic Complexity* dari fungsi melihat status realisasi anggaran :

1).  $V(G) = \text{Total ruang (R)}$ . Total R = 5

- 2).  $V(G) = \text{Jumlah edge} - \text{Jumlah node} + 2 = 22 - 19 + 2 = 5$
- 3).  $V(G) = \text{Total Percabangan (P)} + 1 = 4 + 1 = 5$

h. *Independent Path*

Berikut adalah jalur-jalur *independent* dari fungsi melihat status realisasi anggaran :

- 1). Jalur 1 = 1 - 2 - 4 - 5 - 6 - 7 - 9 - 10 - 11 - 13 - 14 - 15 - 17 - 18 - 19
- 2). Jalur 2 = 1 - 2 - 4 - 5 - 6 - 8 - 9 - 10 - 11 - 13 - 14 - 15 - 17 - 18 - 19
- 3). Jalur 3 = 1 - 2 - 4 - 5 - 6 .. 9 - 10 - 12 - 13 - 14 - 15 - 17 - 18 - 19
- 4). Jalur 4 = 1 - 2 - 4 - 5 - 6 .. 9 - 10 .. 13 - 14 - 16 - 17 - 18 - 19
- 5). Jalur 5 = 1 - 2 - 3 - 4 - 5 - 6 .. 9 - 10 .. 13 - 14 .. - 17 - 18 - 19

i. Kasus Uji dan Hasil Pengujian

Berikut merupakan kasus uji dan hasil uji dari fungsi melihat status realisasi anggaran berdasarkan jalur *independent* yang dihasilkan :

**Tabel 6.4 Test Case Melihat Status Realisasi Anggaran**

No.	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Variabel tgl bernilai "2018-07-17", realMoker bernilai null, array pMoker bernilai 0, realMokerBul bernilai null, array pMokerBul bernilai 0, realMokerTah bernilai null, array pMoker bernilai 0.	Berhasil menampilkan status realisasi anggaran sesuai dengan tanggal.	Berhasil menampilkan status realisasi anggaran sesuai dengan tanggal.	Valid
2.	2	Variabel tgl bernilai "2018-07-17", realMoker bernilai 750, array pMoker bernilai 35, realMokerBul bernilai null, array pMokerBul bernilai 0, realMokerTah bernilai null, array pMoker bernilai 0.	Berhasil menampilkan status realisasi anggaran sesuai dengan tanggal.	Berhasil menampilkan status realisasi anggaran sesuai dengan tanggal.	Valid
3.	3	Variabel tgl bernilai "2018-07-17", realMoker bernilai 750, array pMoker bernilai 35, realMokerBul bernilai	Berhasil menampilkan status realisasi anggaran	Berhasil menampilkan status realisasi anggaran	Valid



Tabel 6.4 Test Case Melihat Status Realisasi Anggaran

No.	No Jalur	Prosedur Uji	Expected Result	Result	Status
		300, array pMokerBul bernilai 18, realMokerTah bernilai null, array pMoker bernilai 0.	sesuai dengan tanggal.	sesuai dengan tanggal.	
4.	4	Variabel tgl bernilai "2018-07-17", realMoker bernilai 750, array pMoker bernilai 35, realMokerBul bernilai 800, array pMokerBul bernilai 18, realMokerTah bernilai 1029, array pMoker bernilai 12.	Berhasil menampilkan status realisasi anggaran sesuai dengan tanggal.	Berhasil menampilkan status realisasi anggaran sesuai dengan tanggal.	Valid
5.	5	Variable tgl bernilai null, Variabel tgl bernilai "2018-07-17", realMoker bernilai 750, array pMoker bernilai 35, realMokerBul bernilai 800, array pMokerBul bernilai 18, realMokerTah bernilai 1029, array pMoker bernilai 12.	Berhasil menampilkan status realisasi anggaran sesuai dengan tanggal.	Berhasil menampilkan status realisasi anggaran sesuai dengan tanggal.	Valid

### 6.1.3 Pengujian Unit Menghitung Estimasi Biaya

a. Pseudocode

Berikut adalah algoritme dari operasi menghitung estimasi biaya.

Tabel 6.5 Algoritme Menghitung Estimasi Biaya

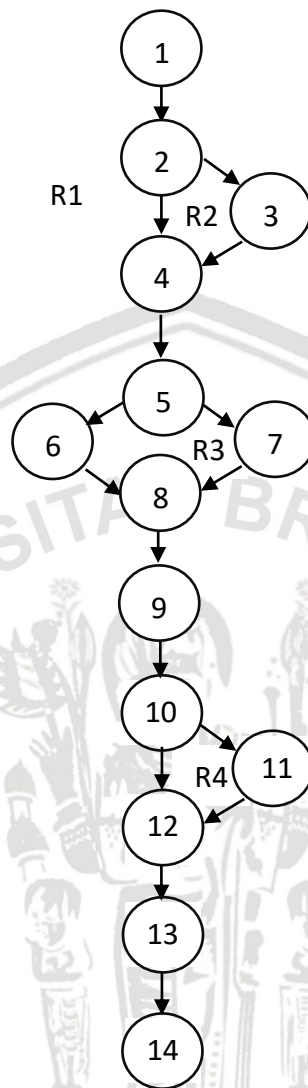
No	Pseudocode
1.	<i>bulan = nilai inputan post date</i> <i>period = nilai inputan post periode</i>
2.	<i>Jika bulan bernilai 0</i>
3.	<i>bulan = bernilai bulan ini</i> <i>tglStart = 01</i>

Tabel 6.5 Algoritme Menghitung Estimasi Biaya

No	Pseudocode
	<i>tglEnd = 10</i>
4.	<i>end if</i>
5.	<i>Jika period bernilai 0 atau 1</i>
6.	<i>array periode bernilai "pertama"</i> <i>tglStart = 01</i> <i>tglEnd = 10</i>
7.	<i>else</i> <i>array periode bernilai "kedua"</i> <i>tglStart = 16</i> <i>tglEnd = 25</i>
8.	<i>end if</i>
9.	<i>jumData bernilai hasil query jumlah data laporan harian dari tglStart sampai tglEnd</i>
10.	<i>Jika jumData bernilai kosong</i>
11.	<i>jumData = 1</i>
12.	<i>end if</i>
13.	<i>totAngga bernilai hasil query jumlah anggaran laporan harian dari tglStart sampai tglEnd</i> <i>ratAngga bernilai totAngga / jumData</i> <i>estMoker bernilai ratAngga * 15</i> <i>array hasil bernilai hasil query data laporan harian dari tglStart sampai tglEnd</i> <i>array ha bernilai 7</i> <i>array modules bernilai viewEstimasi</i>
14.	<i>Load view appAtasan dengan membawa data array</i>

b. *Flow graph*

Berikut adalah *Flow graph* dari algoritme menghitung estimasi biaya.



**Gambar 6.3 Flow Graph fungsi Menghitung Estimasi Biaya**

c. *Cyclomatic Complexity*

Berikut merupakan perhitungan *Cyclomatic Complexity* dari fungsi menghitung estimasi biaya :

- 1).  $V(G) = \text{Total ruang (R)}. \text{ Total R} = 4$
- 2).  $V(G) = \text{Jumlah edge} - \text{Jumlah node} + 2 = 13 - 12 + 2 = 4$
- 3).  $V(G) = \text{Total Percabangan (P)} + 1 = 3 + 1 = 4$

d. *Independent Path*

Berikut adalah jalur-jalur *independent* dari fungsi menghitung estimasi biaya :

- 1). Jalur 1 = 1 - 2 - 4 - 5 - 6 - 8 - 9 - 10 - 12 - 13 - 14



- 2). Jalur 2 = 1 - 2 - 4 - 5 - 7 - 8 - 9 - 10 - 12 - 13 - 14
- 3). Jalur 3 = 1 - 2 - 3 - 4 - 5 . . 8 - 9 - 10 - 12 - 13 - 14
- 4). Jalur 4 = 1 - 2 - 3 - 4 - 5 . . 8 - 9 - 10 - 11 - 12 - 13 - 14

e. Kasus Uji dan Hasil Pengujian

Berikut merupakan kasus uji dan hasil uji dari fungsi menghitung estimasi biaya berdasarkan jalur *independent* yang dihasilkan :

**Tabel 6.6 Test Case Menghitung Estimasi Biaya**

No.	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Variabel bulan bernilai "07-18", variabel period bernilai 0, array peride bernilai "Pertama" variabel tglStart bernilai 01, variable tglEnd bernilai 10, variabel jumData bernilai 3, variable totAngga bernilai 10000, variabel rat angga bernilai 250, vaiabel est moker bernilai 77.8, array bulan bernilai "07-18".	Berhasil menghitung estimasi moker sesuai dengan tanggal yang dipilih.	Berhasil menghitung estimasi moker sesuai dengan tanggal yang dipilih.	Valid
2.	2	Variabel bulan bernilai "07-18", variabel period bernilai 2, array peride bernilai "Kedua" variabel tglStart bernilai 16, variable tglEnd bernilai 25, variabel jumData bernilai 3, variable totAngga bernilai 10000, variabel rat angga bernilai 250, vaiabel est moker bernilai 77.8, array bulan bernilai "07-18".	Berhasil menghitung estimasi moker sesuai dengan tanggal yang dipilih.	Berhasil menghitung estimasi moker sesuai dengan tanggal yang dipilih.	Valid
3.	3	Variabel bulan bernilai kosong. Variabel bulan bernilai "07-18", variabel period bernilai 2, array peride bernilai	Berhasil menghitung estimasi moker sesuai dengan	Berhasil menghitung estimasi moker sesuai dengan	Valid



Tabel 6.6 Test Case Menghitung Estimasi Biaya

No.	No Jalur	Prosedur Uji	Expected Result	Result	Status
		“Kedua” variabel tglStart bernilai 16, variable tglEnd bernilai 25, variabel jumData bernilai 3, variable totAngga bernilai 10000, variabel rat angga bernilai 250, vaiabel est moker bernilai 77.8, array bulan bernilai “07-18”.	tanggal yang dipilih.	tanggal yang dipilih.	
4.	4	Variabel bulan bernilai kosong. Variabel bulan bernilai “07-18”, variabel period bernilai 2, array peride bernilai “Kedua” variabel tglStart bernilai 16, variable tglEnd bernilai 25, variabel jumData bernilai kosong, variabel jumdata bernilai 1, variable totAngga bernilai 0, variabel rat angga bernilai 0, vaiabel est moker bernilai 0.0, array bulan bernilai “07-18”.	Berhasil menghitung estimasi moker sesuai dengan tanggal yang dipilih.	Berhasil menghitung estimasi moker sesuai dengan tanggal yang dipilih.	Valid

## 6.2 Pengujian Integrasi

Pengujian integrasi merupakan jenis pengujian yang memfokuskan untuk menguji kelas-kelas yang saling berkaitan antar satu sama lain. Pengujian ini berfokus pada hubungan dan interaksi antar kelas yang telah dibangun berdasarkan arsitektur kelas yang sebelumnya telah dirancang. Pada tabel 6.7 menunjukkan identifikasi dan rencana uji pada pengujian integrasi dalam pembangunan Dashboard Produksi PTPN XII Gunung Gambir.

**Tabel 6.7 Identifikasi dan Rencana Pengujian Integrasi**

No	Nama Kelas	Nama <i>Method</i>	Tujuan
1	Core	displayLapHar( )	Menampilkan data laporan harian
	Dblaporan	displayLapHarian( )	
2	laporan	addLapPro( )	Menambahkan data laporan produksi ke dalam database
	Dblaporan	tambahLapPro( )	
3	laporan	viewLapPro1( )	Menampilkan daftar komentar pada laporan produksi
	Dbkomentar	displayKomenPro( )	

Hasil pengujian integrasi untuk nomor 1 dapat dilihat pada tabel 6.8 berikut:

**Tabel 6.8 Hasil Pengujian Integrasi Nomor 1**

Nomor Uji	1
Input pertama	"tgl" = "2018-01-02"
Method dari kelas Core	displayLapHar( )
Output pertama / input kedua	"tgl" = "2018-01-02"
Method dari kelas Dblaporan	displayLapHarian( )
<i>Expected result</i>	Berhasil menampilkan data laporan harian tanggal 02 bulan 01 tahun 2018.
<i>Result</i>	Berhasil menampilkan data laporan harian tanggal 02 bulan 01 tahun 2018.
Status	Valid

Hasil pengujian integrasi untuk nomor 2 dapat dilihat pada tabel 6.9 berikut:

**Tabel 6.9 Hasil Pengujian Integrasi Nomor 2**

Nomor Uji	2
Input pertama	kebun = "Karang Anom" tanggal = "2018-06-03" mandor = "Suyit" thTanam = "2010" rOhk = "15" rProduksi = "157"

**Tabel 6.9 Hasil Pengujian Integrasi Nomor 2**

Method dari kelas laporan	addLapPro( )
Output pertama / input kedua	kebun = "Karang Anom" tanggal = "2018-06-03" mandor = "Suyit" thTanam = "2010" rOhk = "15" rProduksi = "157"
Method dari kelas Dblaporan	tambahLapPro( )
<i>Expected result</i>	Data laporan produksi baru dapat ditambahkan atau tercatat ke dalam database sesuai dengan data yang telah dimasukkan
<i>Result</i>	Data laporan produksi baru dapat ditambahkan atau tercatat ke dalam database sesuai dengan data yang telah dimasukkan
Status	Valid

Hasil pengujian integrasi untuk nomor 3 dapat dilihat pada tabel 6.10 berikut:

**Tabel 6.10 Hasil Pengujian Integrasi Nomor 3**

Nomor Uji	3
Input pertama	"tgl" = "2018-06-01"
Method dari kelas laporan	viewLapPro1( )
Output pertama / input kedua	"tgl" = "2018-06-01" "laporan" = "1"
Method dari kelas Dblaporan	displayKomenpro( )
<i>Expected result</i>	Berhasil menampilkan data komentar laporan harian sesuai dengan tanggal 01 bulan 06 tahun 2018.
<i>Result</i>	Berhasil menampilkan data komentar laporan harian sesuai dengan tanggal 01 bulan 06 tahun 2018.
Status	Valid

## 6.3 Pengujian Validasi

Pengujian validasi bertujuan untuk memeriksa apakah kebutuhan yang telah di implementasi telah sesuai dengan yang telah dirancang sebelumnya didalam analisis kebutuhan. Pengujian validasi dilakukan untuk memastikan agar tidak ada fungsi yang menyimpang, serta memastikan semua fungsi dapat berjalan dengan benar. Pengujian validasi menerapkan metode *blackbox testing* yaitu pengujian yang dilakukan tanpa perlu memperhatikan kode program, hanya menjalankan setiap fungsi dengan menyesuaikan terhadap kebutuhan yang telah didefinisikan sebelumnya.

### 6.3.1 Kebutuhan Fungsional

#### 6.3.1.1 Pengujian Validasi *Login*

Pada tabel 6.11 menjelaskan pengujian validasi *login* mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.11 Pengujian Validasi *Login***

DPF_0100 : Login	
Kode Kebutuhan	DPF_0100
Nama Kasus Uji	Login
Prosedur	<ol style="list-style-type: none"> <li>1. Mengisi username berupa NIP/NIK.</li> <li>2. Mengisi password.</li> <li>3. User menekan tombol Login.</li> </ol>
<i>Expected Result</i>	User berhasil terotentikasi dan masuk kedalam halaman beranda.
<i>Result</i>	User berhasil terotentikasi dan masuk kedalam halaman beranda.
Status	Valid

#### 6.3.1.2 Pengujian Validasi *Logout*

Pada tabel 6.12 menjelaskan pengujian validasi *logout* mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.12 Pengujian Validasi *Logout***

DPF_0200 : Logout	
Kode Kebutuhan	DPF_0200
Nama Kasus Uji	Logout
Prosedur	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol logout.</li> </ol>



**Tabel 6.12 Pengujian Validasi Logout**

<b>DPF_0200 : Logout</b>	
<i>Expected Result</i>	User berhasil keluar dari sistem
<i>Result</i>	User berhasil keluar dari sistem
Status	Valid

### 6.3.1.3 Pengujian Validasi Menambah Data Produksi

Pada tabel 6.13 menjelaskan pengujian validasi Menambah Data Produksi mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.13 Pengujian Validasi Menambah Data Produksi**

<b>DPF_0300 : Menambah Data Produksi</b>	
Kode Kebutuhan	DPF_0300
Nama Kasus Uji	Menambah Data Produksi
Prosedur	<ol style="list-style-type: none"> <li>1. Juru Tulis menekan tombol tambah.</li> <li>2. Juru Tulis mengisi data laporan produksi seperti tanggal, nama mandor, tahun tanam, realisasi OHK, realisasi produksi, mutu super, mutu infer.</li> <li>3. Juru Tulis menekan tombol simpan.</li> </ol>
<i>Expected Result</i>	Laporan produksi telah berhasil disimpan dalam database.
<i>Result</i>	Laporan produksi telah berhasil disimpan dalam database.
Status	Valid

### 6.3.1.4 Pengujian Validasi Mengubah Data Produksi

Pada tabel 6.14 menjelaskan pengujian validasi mengubah Data produksi mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.14 Pengujian Validasi Mengubah Data Produksi**

<b>DPF_0400 : Mengubah Data Produksi</b>	
Kode Kebutuhan	DPF_0400
Nama Kasus Uji	Mengubah Data Produksi
Prosedur	<ol style="list-style-type: none"> <li>1. Juru Tulis memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan produksi yang ingin diubah.</li> <li>2. Juru Tulis menekan tombol ubah.</li> </ol>

**Tabel 6.14 Pengujian Validasi Mengubah Data Produksi**

<b>DPF_0400 : Mengubah Data Produksi</b>	
	<ol style="list-style-type: none"> <li>3. Juru Tulis mengisi data laporan produksi yang ingin diubah seperti tanggal, nama mandor, tahun tanam, realisasi OHK, realisasi produksi, mutu super, mutu infer.</li> <li>4. Juru Tulis menekan tombol simpan.</li> </ol>
<i>Expected Result</i>	Laporan produksi telah berhasil diubah dan disimpan dalam database.
<i>Result</i>	Laporan produksi telah berhasil diubah dan disimpan dalam database.
Hasil	Valid

### 6.3.1.5 Pengujian Validasi Menghapus Data Produksi

Pada tabel 4.11 menjelaskan pengujian validasi menghapus Data produksi mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.15 Pengujian Validasi Menghapus Data Produksi**

<b>DPF_0500 : Menghapus Data Produksi</b>	
Kode Kebutuhan	DPF_0500
Nama Kasus Uji	Menghapus Data Produksi
Prosedur	<ol style="list-style-type: none"> <li>1. Juru Tulis memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan produksi yang ingin dihapus.</li> <li>2. Juru tulis menekan tombol Hapus.</li> </ol>
<i>Expected Result</i>	Laporan produksi telah berhasil dihapus dari database.
<i>Result</i>	Laporan produksi telah berhasil dihapus dari database.
Status	Valid

### 6.3.1.6 Pengujian Validasi Menambah Data Harian

Pada tabel 6.16 menjelaskan pengujian validasi menambah Data harian mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.16 Pengujian Validasi Menambah Data Harian**

<b>DPF_0600 : Menambah Data Harian</b>	
Kode Kebutuhan	DPF_0600

**Tabel 6.16 Pengujian Validasi Menambah Data Harian**

<b>DPF_0600 : Menambah Data Harian</b>	
Nama Kasus Uji	Menambah Data Harian
Prosedur	<ol style="list-style-type: none"> <li>1. Juru Tulis menekan tombol tambah.</li> <li>2. Juru Tulis mengisi data laporan harian seperti tanggal, adalah macam pekerjaan, lokasi (nomor) kebun/ tahun tanam, jumlah karyawan Gol. IA, jumlah karyawan lepas/bor, hasil pekerjaan tetap, hasil pekerjaan lepas, satuan upah IA, satuan upah KHL, satuan upah BOR.</li> <li>3. Juru Tulis menekan tombol simpan.</li> </ol>
<i>Expected Result</i>	Laporan harian telah berhasil disimpan dalam database.
<i>Result</i>	Laporan harian telah berhasil disimpan dalam database.
Status	Valid

**6.3.1.7 Pengujian Validasi Mengubah Data Harian**

Pada tabel 6.17 menjelaskan pengujian validasi mengubah Data harian mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.17 Pengujian Validasi Mengubah Data Harian**

<b>DPF_0700 : Mengubah Data harian</b>	
Kode Kebutuhan	DPF_0700
Nama Kasus Uji	Mengubah Data harian
Prosedur	<ol style="list-style-type: none"> <li>1. Juru Tulis memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan harian yang ingin diubah.</li> <li>2. Juru Tulis menekan tombol ubah.</li> <li>3. Juru Tulis mengisi data laporan produksi yang ingin diubah seperti tanggal, adalah macam pekerjaan, lokasi (nomor) kebun/ tahun tanam, jumlah karyawan Gol. IA, jumlah karyawan lepas/bor, hasil pekerjaan tetap, hasil pekerjaan lepas, satuan upah IA, satuan upah KHL, satuan upah BOR.</li> <li>4. Juru Tulis menekan tombol simpan.</li> </ol>
<i>Expected Result</i>	Laporan harian telah berhasil diubah dan disimpan dalam database.
<i>Result</i>	Laporan harian telah berhasil diubah dan disimpan dalam database.

**Tabel 6.17 Pengujian Validasi Mengubah Data Harian**

<b>DPF_0700 : Mengubah Data harian</b>	
Status	Valid

**6.3.1.8 Pengujian Validasi Menghapus Data Harian**

Pada tabel 6.18 menjelaskan pengujian validasi menghapus Data harian mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.18 Pengujian Validasi Menghapus Data Harian**

<b>DPF_0800 : Menghapus Data Harian</b>	
Kode Kebutuhan	DPF_0800
Actor	Menghapus Data harian
Prosedur	1. Juru Tulis memilih tanggal, bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan harian yang ingin dihapus.
Expected Result	Laporan harian telah berhasil dihapus dari database.
Result	Laporan harian telah berhasil dihapus dari database.
Status	Valid

**6.3.1.9 Pengujian Validasi Melihat Data Produksi**

Pada tabel 6.19 menjelaskan pengujian validasi melihat Data produksi mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.19 Pengujian Validasi Melihat Data Produksi**

<b>DPF_0900 : Melihat Data Produksi</b>	
Kode Kebutuhan	DPF_0900
Nama Kasus Uji	Pimpinan, Kantor Induk, Juru Tulis
Prosedur	1. Pimpinan, Kantor Induk, atau Juru Tulis memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan produksi yang ingin tampilkan.
Expected Result	Laporan produksi telah berhasil ditampilkan.
Result	Laporan produksi telah berhasil ditampilkan.
Status	Valid

### 6.3.1.10 Pengujian Validasi Melihat Data Harian

Pada tabel 6.20 menjelaskan pengujian validasi melihat Data mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.20 Pengujian Validasi Melihat Data Harian**

<b>DPF_1000 : Melihat Data harian</b>	
Kode Kebutuhan	DPF_1000
Nama Kasus Uji	Melihat Data harian
Prosedur	1. Pimpinan, Kantor Induk, atau Juru Tulis memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan harian yang ingin ditampilkan.
Expected Result	Laporan produksi telah berhasil ditampilkan.
Result	Laporan produksi telah berhasil ditampilkan.
Status	Valid

### 6.3.1.11 Pengujian Validasi Mencetak Laporan Produksi

Pada tabel 6.21 menjelaskan Pengujian validasi mencetak laporan produksi mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.21 Pengujian Validasi Mencetak Laporan Produksi**

<b>DPF_1100 : Mencetak Laporan Produksi</b>	
Kode Kebutuhan	DPF_1100
Nama Kasus Uji	Mencetak Laporan Produksi
Prosedur	1. Pimpinan, Kantor Induk, Juru Tulis memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan harian yang ingin ditampilkan.
Expected Result	2a. Laporan produksi telah berhasil ditampilkan dan berhasil dicetak.
Result	Laporan produksi telah berhasil ditampilkan dan berhasil dicetak.
Status	Valid



### 6.3.1.12 Pengujian Validasi Mencetak Laporan Harian

Pada tabel 6.22 menjelaskan *use case scenario* mencetak laporan harian mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.22 Pengujian Validasi Mencetak Laporan Harian**

<b>DPF_1200 : Mencetak Laporan Harian</b>	
Kode Kebutuhan	DPF_1200
Nama kasus Uji	Mencetak Laporan Harian
Prosedur	1. Pimpinan, Kantor Induk, Juru Tulis memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan harian yang ingin ditampilkan.
Expected Result	Laporan harian telah berhasil ditampilkan dan berhasil dicetak.
Result	Laporan harian telah berhasil ditampilkan dan berhasil dicetak.
Status	Valid

### 6.3.1.13 Pengujian Validasi Melihat Status Realisasi Produksi

Pada tabel 6.23 menjelaskan pengujian validasi melihat status produksi mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.23 Pengujian Validasi Melihat Status Realisasi Produksi**

<b>DPF_1300 : Melihat Status Realisasi Porduksi</b>	
Kode Kebutuhan	DPF_1300
Nama Kasus Uji	Melihat Status Realisasi Porduksi
Prosedur	1. Pimpinan, Kantor Induk, Juru Tulis memilih bulan dan tahun yang tersedia dalam fungsi dropdown yang akan ditampilkan status realisasi produksinya.
Expected Result	Pimpinan, Kantor Induk, atau Juru Tulis mendapatkan infromasi status produksi.
Result	Pimpinan, Kantor Induk, atau Juru Tulis mendapatkan infromasi status produksi.
Status	Valid

### 6.3.1.14 Pengujian Validasi Melihat Total Produksi

Pada tabel 6.24 menjelaskan pengujian validasi melihat total produksi mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.24 Pengujian Validasi Total Produksi**

<b>DPF_1400 : Melihat Total Realisasi Produksi</b>	
Kode Kebutuhan	DPF_1400
Nama Kasus Uji	Melihat Total Realisasi Produksi
Prosedur	1. Pimpinan, Kantor Induk, atau Juru Tulis memilih tahun yang tersedia dalam fungsi dropdown yang akan ditampilkan status realisasi produksinya.
Expected Result	Pimpinan, Kantor Induk, atau Juru Tulis mendapatkan informasi mengenai perbandingan realisasi produksi terhadap target berdasarkan tahun yang dipilih.
Result	Pimpinan, Kantor Induk, atau Juru Tulis mendapatkan informasi mengenai perbandingan realisasi produksi terhadap target berdasarkan tahun yang dipilih.
Status	Valid

### 6.3.1.15 Pengujian Validasi Menghitung Estimasi Biaya

Pada tabel 6.25 menjelaskan pengujian validasi menghitung estimasi biaya mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.25 Pengujian Validasi Menghitung Estimasi Biaya**

<b>DPF_1500 : Menghitung Estimasi Biaya</b>	
Kode Kebutuhan	DPF_1500
Nama Kasus Uji	Menghitung Estimasi Biaya
Prosedur	1. Juru tulis menekan tombol Estimasi biaya 2. Juru Tulis memilih bulan, tahun dan masa periode yang tersedia dalam fungsi dropdown yang akan dihitung estimasi biaya produksinya.
Expected Result	Juru Tulis mendapatkan informasi mengenai tptal realisasi biaya yang akan dikeluarkan.
Result	Juru Tulis mendapatkan informasi mengenai tptal realisasi biaya yang akan dikeluarkan.
Status	Valid

### 6.3.1.16 Pengujian Validasi Melihat Status Biaya

Pada tabel 6.26 menjelaskan pengujian validasi melihat status mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.26 Pengujian Validasi Melihat Status Biaya**

<b>DPF_1600 : Melihat Status Biaya</b>	
Kode Kebutuhan	DPF_1600
Nama Kasus Uji	Melihat Status Biaya
Prosedur	1. Pimpinan, Kantor Induk, atau Juru Tulis memilih bulan yang tersedia dalam fungsi dropdown yang akan ditampilkan status anggarannya.
Expected Result	Pimpinan, Kantor Induk, atau Juru Tulis mendapatkan informasi mengenai status biaya yang dikeluarkan sesuai dengan bulan yang dipilih.
Result	Pimpinan, Kantor Induk, atau Juru Tulis mendapatkan informasi mengenai status biaya yang dikeluarkan sesuai dengan bulan yang dipilih.
Status	Valid

### 6.3.1.17 Pengujian Validasi Menambah Target Produksi

Pada tabel 6.27 menjelaskan pengujian validasi menambah target produksi mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.27 Pengujian Validasi Menambah Target Produksi**

<b>DPF_1700 : Menambah Target Produksi</b>	
Kode Kebutuhan	DPF_1700
Nama Kasus Uji	Menambah Target Produksi
Prosedur	<ol style="list-style-type: none"> <li>1. Pimpinan atau kantor induk menekan tombol tambah target.</li> <li>2. Pimpinan atau kantor Induk mengisi jumlah target produksi.</li> <li>3. Pimpinan atau kantor Induk menekan tombol simpan.</li> </ol>
Expected Result	Target produksi telah berhasil disimpan dalam database.
Result	Target produksi telah berhasil disimpan dalam database.
Status	Valid

### 6.3.1.18 Pengujian Validasi Menambah Target Biaya

Pada tabel 6.28 menjelaskan pengujian validasi menambah target mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.28 Pengujian Validasi Menambah Target Biaya**

<b>DPF_1800 : Menambah Target Biaya</b>	
Kode Kebutuhan	DPF_1800
Nama Kasus Uji	Menambah Target Biaya
Prosedur	<ol style="list-style-type: none"> <li>1. Pimpinan atau kantor Induk menekan tombol tambah target.</li> <li>2. Pimpinan atau kantor Induk mengisi jumlah target biaya.</li> <li>3. Pimpinan atau kantor Induk menekan tombol simpan.</li> </ol>
Expected Result	Target biaya telah berhasil disimpan dalam database.
Result	Target biaya telah berhasil disimpan dalam database.
Status	Valid

### 6.3.1.19 Pengujian Validasi Menambah Komentar

Pada tabel 6.29 menjelaskan pengujian validasi menambah komentar mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.29 Pengujian Validasi Menambah Komentar**

<b>DPF_1900 : Menambah Komentar</b>	
Kode Kebutuhan	DPF_1900
Nama Kasus Uji	Menambah Komentar
Prosedur	<ol style="list-style-type: none"> <li>1. Pimpinan memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan produksi yang ingin ditambah komentar.</li> <li>2. Pimpinan mengisi komentar dikolom yang telah disediakan.</li> <li>3. Pimpinan menekan tombol simpan.</li> </ol>
Expected Result	Komentar telah berhasil disimpan dalam database dan komentar berhasil ditampilkan.
Result	Komentar telah berhasil disimpan dalam database dan komentar berhasil ditampilkan.
Status	Valid

### 6.3.1.20 Pengujian Validasi Menghapus Komentar

Pada tabel 6.30 menjelaskan pengujian validasi menghapus komentar mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.30 Pengujian Validasi Menghapus Komentar**

<b>DPF_2100 : Menghapus Komentar</b>	
Kode Kebutuhan	DPF_2100
Nama Kasus Uji	Menghapus Komentar
Prosedur	<ol style="list-style-type: none"> <li>1. Pimpinan memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan produksi yang ingin dihapus.</li> <li>2. Pimpinan menekan tombol hapus.</li> </ol>
Expected Result	Komentar berhasil dihapus dari database dan komentar berhasil ditampilkan.
Result	Komentar berhasil dihapus dari database dan komentar berhasil ditampilkan.
Status	Valid

### 6.3.1.21 Pengujian Validasi Melihat Komentar

Pada tabel 6.31 menjelaskan pengujian validasi melihat komentar mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.31 Pengujian Validasi Melihat Komentar**

<b>DPF_2200 : Melihat Komentar</b>	
Kode Kebutuhan	DPF_2200
Nama Kasus Uji	Melihat Komentar
Prosedur	<ol style="list-style-type: none"> <li>1. Pimpinan memilih tanggal bulan, dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan produksi yang ingin ditampilkan.</li> <li>2. Sistem menampilkan data komentar dari database sesuai dengan tanggal bulan dan tahun yang telah diinputkan pengguna.</li> </ol>
Expected Result	Komentar berhasil ditampilkan.
Result	Komentar berhasil ditampilkan.
Status	Valid



### 6.3.1.22 Pengujian Validasi Menambah Pengguna

Pada tabel 6.32 menjelaskan pengujian validasi menambah pengguna mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.32 Pengujian Validasi Menambah Pengguna**

DPF_2300 : Menambah Pengguna	
Kode Kebutuhan	DPF_2300
Nama Kasus Uji	Menambah Pengguna
Prosedur	<ol style="list-style-type: none"> <li>1. Admin menekan tombol Tambah pengguna</li> <li>2. Admin mengisi data pengguna yang akan ditambahkan berupa NIP/NIK sebagai username, nama, jabatan dan <i>password</i>.</li> <li>3. Admin menekan tombol simpan.</li> </ol>
Expected Result	Pengguna telah berhasil disimpan dalam database.
Result	Pengguna telah berhasil disimpan dalam database.
Status	Valid

### 6.3.1.23 Pengujian Validasi Mengubah Pengguna

Pada tabel 6.33 menjelaskan pengujian validasi mengubah pengguna mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.33 Pengujian Validasi Mengubah Pengguna**

DPF_2400 : Mengubah Pengguna	
Kode Kebutuhan	DPF_2400
Nama Kasus Uji	Mengubah Pengguna
Prosedur	<ol style="list-style-type: none"> <li>1. Admin menekan tombol Ubah</li> <li>2. Admin mengisi data pengguna yang akan ditambahkan berupa nama, jabatan dan <i>password</i>.</li> <li>3. Admin menekan tombol simpan.</li> </ol>
Expected Result	Pengguna telah berhasil diubah, disimpan dalam database dan berhasil melihat semua data pengguna yang terdaftar.
Result	Pengguna telah berhasil diubah, disimpan dalam database dan berhasil melihat semua data pengguna yang terdaftar.
Status	Valid

### 6.3.1.24 Pengujian Validasi Melihat Pengguna

Pada tabel 6.34 menjelaskan pengujian validasi melihat pengguna mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.34 Pengujian Validasi Melihat Pengguna**

<b>DPF_2500 : Melihat Pengguna</b>	
Kode Kebutuhan	DPF_2500
Nama Kasus Uji	Melihat Pengguna
Prosedur	1. Admin menekan tombol pengguna.
Expected Result	Admin telah berhasil melihat semua data pengguna yang terdaftar.
Result	Admin telah berhasil melihat semua data pengguna yang terdaftar.
Status	Valid

### 6.3.1.25 Pengujian Validasi Menghapus Pengguna

Pada tabel 6.35 menjelaskan pengujian validasi menghapus pengguna mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

**Tabel 6.35 Pengujian Validasi Menghapus Pengguna**

<b>DPF_2600 : Menghapus Pengguna</b>	
Kode Kebutuhan	DPF_2600
Nama Kasus Uji	Menghapus Pengguna
Prosedur	1. Admin menekan tombol Hapus pengguna.
Expected Result	Pengguna telah berhasil dihapus dari database dan berhasil melihat semua data pengguna yang terdaftar.
Post-Condition	Pengguna telah berhasil dihapus dari database dan berhasil melihat semua data pengguna yang terdaftar.
Status	Valid

## 6.3.2 Kebutuhan Non Fungsional

### 6.3.2.1 Pengujian Validasi *Compatibility*

Pengujian *compatibility* dilakukan dengan cara menguji sistem yang telah dibangun menggunakan *browser* yang beragam, yang umum digunakan oleh pengguna di dalam menjalankan aplikasi berbasis web. Dalam hal ini dilakukan pengujian menggunakan aplikasi *browser* Mozilla Firefox dan Google Chrome. Pengujian dilakukan dengan menggunakan tools Cross Browser Testing yg terdapat dihalaman *crossbrowstesting.com*. David (2011) mengemukakan indikator yang digunakan adalah *Font size validation, Page Content Alignment, All images and alignment, Header and footer sections*. Hasil pengujian validasi *compatibility* dapat dilihat dalam tabel 6.36 berikut:

**Tabel 6.36 Pengujian Validasi *Compatibility***

No	Indikator	Google Chrome	Mozilla Firefox
1.	<i>Font size validation</i> (Ukuran font tidak berubah).	Valid	Valid
2.	<i>Page Content Alignment</i> (penempatan dan tata letak konten di halaman tetap).	Valid	Valid
3.	<i>All Image Aligment</i> (Semua gambar dapat ditampilkan dan ukurannya sesuai).	Valid	Valid
4.	<i>Header and Footer Sections</i> (Tampilan dan tata letak header dan footer sesuai)	Valid	Valid

## BAB 7 PENUTUP

Pada bab ini akan dilakukan penarikan kesimpulan mulai dari tahap analisis hingga tahap pengujian sistem. Berdasarkan penjabaran mengenai latar belakang masalah sistem *dashboard*, landasan pustaka yang membahas metode yang akan di gunakan. Selain penarikan kesimpulan, nantinya juga akan jabarkan saran. Penjabaran saran dilakukan untuk mengetahui keasalahan maupun kekurangan yang terjadi, sehingga nantinya jika ada penelitian selanjutnya yang dilakukan untuk mengembangkan ataupun melakukan perbaikan.

### 7.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan sesuai dengan metodologi yang telah dideskripsikan, menghasilkan beberapa kesimpulan sebagai berikut:

1. Pada proses analisis kebutuhan di dalam pembangunan dashboard produksi pada PTPN XII persero kantor wilayah III gunung gambir berbasis web, tahapan yang dilakukan yakni mendeskripsikan gambaran umum sistem yang kemudian didapatkan 5 orang aktor serta 26 kebutuhan fungsional dan 1 kebutuhan non fungsional. Setiap kebutuhan fungsional tersebut dimodelkan kedalam use case diagram serta use case scenario.
2. Proses pembangunan dashboard produksi pada PTPN XII persero kantor wilayah III gunung gambir berbasis web dilakukan berdasarkan pemodalan kebutuhan yang telah di definisikan sebelumnya. Pada perancangan dilakukan dengan menggunakan sequence diagram untuk menggambarkan interaksi antar komponen sistem serta antara aktor dan sistem. Sequence diagram yang dihasilkan berjumlah sama dengan kebutuhan yang didefinisikan. Pemodelan perancangan pada pengembangan sistem ini juga menggunakan class diagram, perancangan komponen, perancangan basis data serta perancangan antar muka. Class diagram digunakan untuk menggambarkan kelas-kelas yang berinteraksi didalam sistem. Pada perancangan komponen di hasilkan 3 algoritme dari fungsi utama yang menyusun sistem. Pada perancangan basis data menghasilkan 5 tabel utama yang diperoleh pada tahapan ERD dan PDM. Pada pengembangan sistem ini juga dilakukan tahapan perancangan antar muka yang digunakan untuk memberikan gambaran umum halaman yang menyusun sistem, dan dihasilkan sebuah halaman utama yang seluruh fungsinya terdapat pada halaman tersebut.
3. Implementasi di dalam pembangunan dashboard produksi pada PTPN XII persero kantor wilayah III gunung gambir berbasis *web* dilakukan dengan menggunakan bahasa pemrograman php, html, javascript sesuai dengan bahasa yang mendukung pengembangan sistem berbasis *web*. Pada proses implementasi juga menghasilkan implementasi basis data yang ditunjukkan dalam bentuk *physical data model* serta implementasi antarmuka berdasarkan hasil perancangan antar muka yang ditetapkan sebelumnya.

4. Pada proses pengujian dilakukan dengan menggunakan beberapa strategi pengujian, yaitu pengujian unit, pengujian integrasi, serta pengujian validasi. Pengujian unit dalam penelitian ini dilakukan terhadap 3 fungsi utama yang menyusun sistem dan memberikan status valid untuk setiap kasus ujinya. Pengujian integrasi dilakukan terhadap 2 fungsi yang memiliki hubungan antara beberapa kelas dan memberikan status valid untuk setiap kasus ujinya. Pengujian validasi dilakukan terhadap seluruh kebutuhan fungsional dan kebutuhan non fungsional dan memberikan status valid, yang artinya sistem yang dibangun berjalan sesuai tujuan dan bebas dari adanya kegagalan.

## 7.2 Saran

Dalam pembangunan dashboard produksi pada PTPN XII persero kantor wilayah III gunung gambir berbasis web yang dilakukan dalam penelitian ini tentunya masih terdapat beberapa kekurangan, sehingga saran untuk pengembangan lebih lanjut yaitu:

1. Adanya penyempurnaan model chart atau diagram yang lebih menarik dan lebih informatif, sehingga nantinya akan memudahkan dalam memperoleh informasi dan memonitoring untuk menyesuaikan pelaksanaan strategi.
2. Adanya penyempurnaan atau penggunaan algoritme lain yang berkaitan dengan probabilitas dan statistika yang digunakan untuk memprediksi data, sehingga hasil estimasi dapat lebih optimal.
3. Adanya penambahan fitur untuk upload data dokumen atau lapaoran yang sebelumnya telah direcord dalam bentuk excel, sehingga nantinya akan dengan mudah mencari data dokumen atau laporan pada tahun tahun sebelumnya.



## DAFTAR PUSTAKA

- A.S Rosa dan Salahuddin M, 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung : Modula.
- Al Fatta, Hanif. 2007. Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern. Yogyakarta: Andi.
- Alexander, Michael. 2014. *Excel Dashboards and Reports For Dummies, 2nd Edition*. Paperback.
- Anhar. 2010. *Panduan Menguasai PHP & MySQL Secara Otodidak*. Jakarta: MediaKita.
- Aziz, Sholecul. 2017. *Menguasai PHP dan MySQL*. Jakarta: Kuncikom.
- Black, Rex. 2009. *Managing the Testing Process : Practical Tools and Techniques for Managing Hardware and Software Testing, 3rd Edition*. Indiana : Wiley Publishing, Inc.
- Collins, Heidi. 2003. *Enterprise Knowledge Portal*. New york: Amacom. Tersedia di: Google Books <[https://books.google.co.id/books?id=Y97G7A4bpS0C&printsec=frontcover&dq=isbn:0814413153&hl=id&sa=X&ved=0ahUKEwj-3-TMya\\_aAhXFQ48KHfzgBH4QuwUILTAA#v=onepage&q&f=false](https://books.google.co.id/books?id=Y97G7A4bpS0C&printsec=frontcover&dq=isbn:0814413153&hl=id&sa=X&ved=0ahUKEwj-3-TMya_aAhXFQ48KHfzgBH4QuwUILTAA#v=onepage&q&f=false)> [Diakses 18 Januari 2018].
- Conor. 1974. *Monitoring Pekerjaan*. Yogyakarta: Nur Cahaya
- David, Assaf Ben. (2011). *Mobile Application Testing (Best Practices to Ensure Quality)*. Amdocs, 2.
- Few, Stephen. 2006. *Information Dashboard Design*. Sebastopol, Calif.: O'Reilly. ISBN: 0-596-10016-7.
- Hidayat, Rahmat. 2010. *Cara Praktis Membangun Website Gratis*. Jakarta: PT Elex Media Komputindo. Tersedia di: Google Books <<https://books.google.co.id/books?id=zRq2O7VkNSgC&printsec=frontcover#v=onepage&q&f=true>> [Diakses 18 Januari 2018]
- Ilhamsyah dan Rahmayuda S. 2017. *Perancangan Model Dashboard Untuk Monitoring Evaluasi Mahasiswa*. Jurnal Informatika : Jurnal Pengembangan IT (JPIT). Universitas Tanjungpura.
- Irawan, Davit., 2015. *Dashboard Profile Calon Mahasiswa Baru (Studi Kasus: Pada STMIK MURA Lubuklinggau)*. Jurnal Teknologi Indonesia. STMIK Musi Rawas Lubuklinggau.
- Malik, S., 2005. *Enterprise Dashboards: Design and Best Practices for IT*. Hoboken: John Wiley & Sons, Inc. Tersedia di: Google Books <<https://books.google.co.id/books?id=MAGN-UTxJwMC&printsec=frontcover&hl=id#v=onepage&q&f=false>> [Diakses 10 Januari 2018]

- Nugroho, Hadi. 2009. *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Yogyakarta: Andi.
- Oktavia, N., 2012. *Sistem Pengawasan Kinerja Unit Produksi Dalam Pemenuhan Marketing Order Di PT Sanbe Farma Bandung Berbasis Web*. Jurnal Ilmiah Komputer dan Informatika (KOMPUTA). Universitas Komputer Indonesia.
- Pradipta, A., Prasetyo, Y., Ambarsari, M., 2015. *Pengembangan Web E-Commerce Bajana Sari Menggunakan Metode Prototype*. Jurnal Tugas Akhir. Universitas Telkom
- Pressman, R., 2010. *Software Engineering : A Practitioner's Approach, 7th ed*. New York: Mc Grow Hill.
- Ptpn12. 2015. Ptpn12.com. *Sejarah Singkat*. <https://www.ptpn12.com/index.php/tentang-kami/profil>. [Diakses 14 februari 2018].
- Rasmussen, H., Nils, Bansal, Manish, dan Chen, Claire Y. 2009. *Business Dashboards: A Visual Catalog for Design and Deployment*. Wiley.
- Sanusi. 2017. Tribunnews.com. *Industri Perkebunan Jadi Penopang Ekonomi Nasional*. Tersedia di: <<http://www.tribunnews.com/bisnis/2017/12/11/industri-perkebunan-jadi-penopang-ekonomi-nasional>>. [Diakses 13 januari 2018].
- Sunaryanto, Agus., et al. 2012. *Monitoring Penegakan Hukum*. Jakarta Selatan: Indonesia Corruption Watch.
- Susanto, R., Andriana, A.D., 2006. *Perbandingan Model Waterfall Dan Prototyping Untuk Pengembangan Sistem Informasi*. MIU. Universitas Komputer Indonesia.
- Wicaksono, S.R., 2017. *Rekayasa Perangkat Lunak*. Malang: Seribu Bintang. Tersedia di: Google Books <<https://books.google.co.id/books?id=h7o0DwAAQBAJ&printsec=frontcover&hl=id#v=onepage&q&f=false>> [Diakses 10 Februari 2018]
- Yuhfizar. 2008. *10 Jam Menguasai Internet, Teknologi & Aplikasinya + CD*. Elex Media Komputindo : Jakarta.