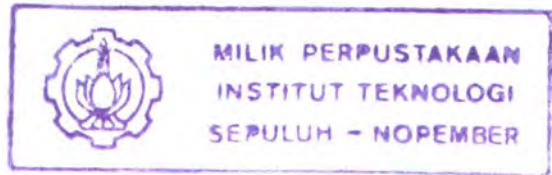


21.0661/ITS/14/05



**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK
ANALISA KERUGIAN PRODUKSI DAN INDIKATOR
KINERJA MELALUI *TOTAL PRODUCTIVE MAINTENANCE*
DI LAMP COMPONENT FACTORY – PT PHILIPS RALIN
ELECTRONICS SURABAYA**

TUGAS AKHIR



RSIF
005.1
Kua
P-1

2003

PERPUSTAKAAN ITS	
Tgl. Terima	19 - 8 - 2003
Terima Dari	H
No. Agenda Prp.	218936

Disusun Oleh :
DEDDI KURNIAWAN
NRP. 5198 100 011

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA**

2003

**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK
ANALISA KERUGIAN PRODUKSI DAN INDIKATOR
KINERJA MELALUI *TOTAL PRODUCTIVE MAINTENANCE*
DI LAMP COMPONENT FACTORY - PT PHILIPS RALIN
ELECTRONICS SURABAYA**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan

Untuk Memperoleh Gelar Sarjana Komputer

Pada

Jurusan Teknik Informatika

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui/Menyetujui,

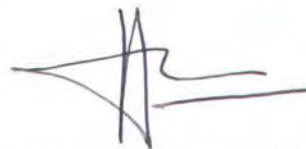
Dosen Pembimbing I



(Febriliyan Samopa S.Kom M.Kom)

NIP. 131 996 151

Dosen Pembimbing II



(Ir. Khakim Ghozali)

NIP. 040 030 950

SURABAYA

JULI, 2003



ABSTRAK

ABSTRAK

Pabrik atau perusahaan akan memerlukan informasi mengenai tingkat kinerja dan kerugian produksi untuk menentukan kebijakan-kebijakan perusahaan. Informasi ini akan sangat berguna untuk melakukan benchmarking terhadap proses produksi dan maintenance. Proses produksi dapat menghasilkan hasil yang maksimal apabila ditunjang dengan proses maintenance yang baik.

Untuk melakukan pengukuran tingkat kinerja dan kerugian produksi dapat dilakukan melalui TPM (Total Productive Maintenance). TPM ini mengintegrasikan seluruh aspek yang berhubungan dengan perusahaan mulai dari pekerja, proses produksi dan proses maintenance untuk bersama-sama membangun suatu sistem produksi dan maintenance yang terintegrasi dengan baik. Dengan penjadwalan proses maintenance yang baik akan menjaga mesin dalam keadaan yang prima sehingga dapat berproduksi secara optimal. Untuk memudahkan dalam melakukan pengukuran tingkat keefektifan dan kerugian produksi dari perusahaan tersebut maka dibuatkan suatu aplikasi berbasis web untuk melakukan analisa. Aplikasi ini akan melibatkan pihak management, maintenance dan production.

Dari perhitungan tersebut dapat diketahui indikator kinerja dan kerugian produksi yang berupa Availability, Direct Yield (Quality Rate), Performance Rate dan Overall Equipment Effectiveness. Dengan adanya perhitungan terhadap kinerja perusahaan ini pihak management akan dapat menentukan kebijakan-kebijakan yang lebih baik dalam proses produksi dan maintenance.

Kata kunci : Total Productive Maintenance, Preventive Maintenance, Corrective Maintenance, Predictive Maintenance, Production Loss, Performance Indicator, Benchmark.



KATA PENGANTAR

KATA PENGANTAR

Segala puji dan syukur semata ditujukan ke hadirat ALLAH SWT yang telah memberikan rahmat dan hidayah-Nya sehingga memungkinkan penulis untuk menyelesaikan Tugas Akhir yang berjudul **“Perancangan dan Pembuatan Perangkat Lunak Analisa Kerugian Produksi dan Indikator Kinerja Melalui Total Productive Maintenance di Lamp Component Factory – PT Philips Ralin Electornics Surabaya”**.

Mata Kuliah Tugas Akhir yang memiliki beban sebesar 4 satuan kredit disusun dan diajukan sebagai salah satu syarat untuk menyelesaikan program strata satu (S-1) pada jurusan Teknik Informatika di Institut Teknologi Sepuluh Nopember Surabaya.

Dalam penyusunan Tugas Akhir ini, Penulis berusaha untuk menerapkan ilmu yang telah didapat selama menjalani perkuliahan dengan tidak terlepas dari petunjuk, bimbingan, bantuan, dan dukungan berbagai pihak.

Dengan tidak lupa akan kodratnya sebagai manusia, Penulis menyadari bahwa dalam karya Tugas Akhir ini masih mengandung kekurangan di sana-sini sehingga dengan segala kerendahan hati Penulis masih dan insya Allah akan tetap terus masih mengharapkan saran serta kritik yang membangun dari rekan-rekan pembaca.

Surabaya, Juli 2003

Penulis

UCAPAN TERIMA KASIH

Dengan mengucapkan syukur Alhamdulillah kepada **Allah SWT** atas rahmat, karunia dan pertolongan-Nya sehingga Tugas Akhir ini bisa terselesaikan. Selain itu penulis hendak menyampaikan rasa penghormatan yang setinggi-tingginya serta rasa terima kasih kepada pihak-pihak yang telah memberi bantuan baik itu berupa moril maupun material dan langsung maupun tidak langsung, kepada :

1. Kedua orang tua tercinta, Sulih Ismadji dan Lilik Susriani atas didikan, bimbingan dan dukungannya selama ini.
2. Bapak Febriliyan Samopa S.Kom M.Kom dan Bapak Ir. Khakim Ghozali sebagai dosen pembimbing Tugas Akhir penulis, atas segala bimbingan dan arahnya.
3. Bapak Yudhi Purwananto S.Kom M.Kom selaku Ketua Jurusan Teknik Informatika ITS dan Bapak Ir. Arif Djunaidy M.Sc Ph.D selaku Dekan Fakultas Teknologi Informasi ITS dan sekaligus sebagai dosen wali penulis.
4. Dosen-dosen Jurusan Teknik Informatika ITS serta segenap Karyawan Tata Usaha Teknik Informatika ITS.
5. Karyawan dan karyawan PT. Philips Ralin Electronics Surabaya, Bapak Sugeng Hardjanto, Bapak Mirza, Bapak Bambang, Bapak Istiono, dan yang lainnya atas pengetahuan dan bimbingan yang diberikan kepada penulis.
6. Adikku yang imut, Dini dan keluarga di Blitar semuanya atas dorongan dan dukungan serta doanya sehingga penulis dapat menyelesaikan Tugas Akhir ini.

7. Teman-teman angkatan 98, C0E Ceria semuanya yang tidak dapat disebutkan satu persatu.
8. Teman-teman seperjuangan dalam mengerjakan Tugas Akhir : Deni, Didit, Cphee, Adi, Aby, Reza, Wong, Caca, Mbah Nur, Ario, Bhakti, Nana, Citra, Intan, Ita', Nisa dan yang lainnya.
9. Teman-teman kost di Wisma Permai : Vic, Dimas, Puput, Sambel, Adi dan yang lainnya.
10. Teman-teman di Blitar dan di Malang semuanya atas kebersamaan kita selama ini dan kenangan yang tak terlupakan.
11. Serta semua orang dan semua pihak yang telah membantu penulis selama ini, terima kasih semuanya.



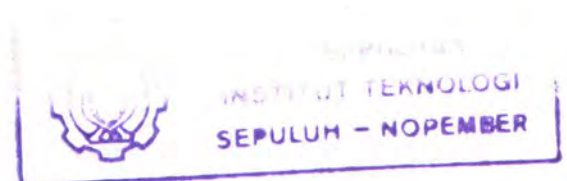
DAFTAR ISI

DAFTAR ISI

ABSTRAK.....	i
KATA PENGANTAR.....	ii
UCAPAN TERIMA KASIH.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	x
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Permasalahan.....	2
1.4 Tujuan Pembuatan Tugas Akhir.....	3
1.5 Metodologi Pembuatan Tugas Akhir.....	3
1.6 Sistematika Penulisan.....	5
BAB 2.....	7
DASAR TEORI.....	7
2.1 Total Productive Maintenance.....	7
2.1.1 Maintenance.....	8
2.1.1.1 Preventive Maintenance.....	8
2.1.1.2 Corrective Maintenance.....	9
2.1.1.3 Predictive Maintenance.....	9
2.1.2 Indikator Kerja dan Kerugian Produksi.....	9
2.1.3 Definisi dasar yang berhubungan dengan waktu produksi.....	10
2.1.4 Availability.....	12
2.1.5 Direct Yield (Quality Rate).....	12
2.1.6 Performance Rate.....	13
2.1.7 Logistic Efficiency.....	13
2.1.8 OEE (Overall Equipment Effectiveness).....	13
2.1.9 OPE (Overall Production Effectiveness).....	14
2.1 Teknologi Web.....	14
2.1.1 HTML (Hypertext Markup Language).....	14
2.1.1.1 Bagian HEAD.....	15
2.1.1.2 Bagian BODY.....	15
2.1.1.3 Bagian FORM.....	16
2.1.2 DHTML (Dynamic Hypertext Markup Language).....	16
2.1.3 VBScript.....	17
2.1.4 JavaScript.....	17
2.1.5 ASP (Active Server Page).....	18
2.2 Data Base.....	19
2.2.1 Pengertian Secara Umum.....	19
2.2.2 SQL (Structured Query Language).....	20
2.2.2.1 DDL (Data Definition Language).....	20
2.2.2.2 DML (Data Manipulation Language).....	20
2.3 UML (Unified Modelling Language).....	21

2.3.1	Definisi.....	21
2.3.2	Diagram UML.....	22
BAB 3	25
DESAIN DAN IMPLEMENTASI SISTEM	25
3.1	Definisi Sistem.....	25
3.1.1	Sistem Yang Sudah Ada.....	25
3.1.1.1	Management.....	25
3.1.1.2	Maintenance Department.....	26
3.1.1.3	Production Department.....	26
3.2	Desain Sistem.....	26
3.2.1	Pembagian Sistem.....	27
3.2.1.1	Sistem Informasi Produksi dan Maintenance.....	27
3.2.1.2	Analisa TPM.....	29
3.2.2	Use Case View.....	29
3.2.2.1	Actor.....	29
3.2.2.2	Use Case Diagram.....	30
3.2.2.3	Activity Diagram.....	35
3.2.2.4	Sequence Diagram.....	45
3.2.3	Desain Data Base.....	54
3.2.4	Desain Antar Muka.....	66
3.3	Implementasi Sistem.....	71
3.3.1	Implementasi Data Base.....	71
3.3.1.1	Pembuatan Koneksi dengan Data Base.....	71
3.3.2	Sistem Informasi Produksi dan Maintenance.....	72
3.3.2.1	Pembuatan Fasilitas Login.....	72
3.3.2.2	Pembuatan Antar Muka Aplikasi.....	73
3.3.2.3	Pembuatan Fasilitas Menampilkan Data.....	74
3.3.2.4	Pembuatan Fasilitas Menambah Data.....	76
3.3.2.5	Pembuatan Fasilitas Mengubah Data.....	77
3.3.2.6	Pembuatan Fasilitas Menghapus Data.....	78
3.3.3	Analisa TPM.....	78
3.3.3.1	Analisa Availability, Direct Yield, Performance Rate dan OEE.....	78
3.3.3.2	Data Flag.....	80
3.3.3.3	Machine Technical Statistics.....	82
3.3.3.4	Production Performance.....	83
3.3.3.5	Grafik.....	83
3.3.4	Pendukung Keputusan.....	85
3.3.4.1	Overall Equipment Effectiveness.....	85
3.3.4.2	Overall Utilisation.....	86
3.3.4.3	Maintenance.....	86
BAB 4	88
UJI COBA DAN ANALISA	88
4.1	Lingkungan Uji Coba.....	88
4.2	Parameter Yang Diamati.....	89
4.3	Skenario Uji Coba.....	90

4.3.1	Kemampuan Multi User.....	90
4.3.2	Kemampuan Fungsionalitas.....	91
4.3.3	Kecepatan Aliran Data.....	93
4.3.4	Efisiensi.....	93
4.4	Hasil Pengujian dan Analisa	93
4.4.1	Terhadap Kemampuan Multi User.....	93
4.4.2	Terhadap Fungsionalitas	94
4.4.3	Terhadap Aliran Data.....	97
4.4.4	Terhadap Efisiensi	97
BAB 5	99
KESIMPULAN DAN SARAN.....		99
5.1	Kesimpulan	99
5.2	Saran	100
DAFTAR PUSTAKA		101
LAMPIRAN I		103



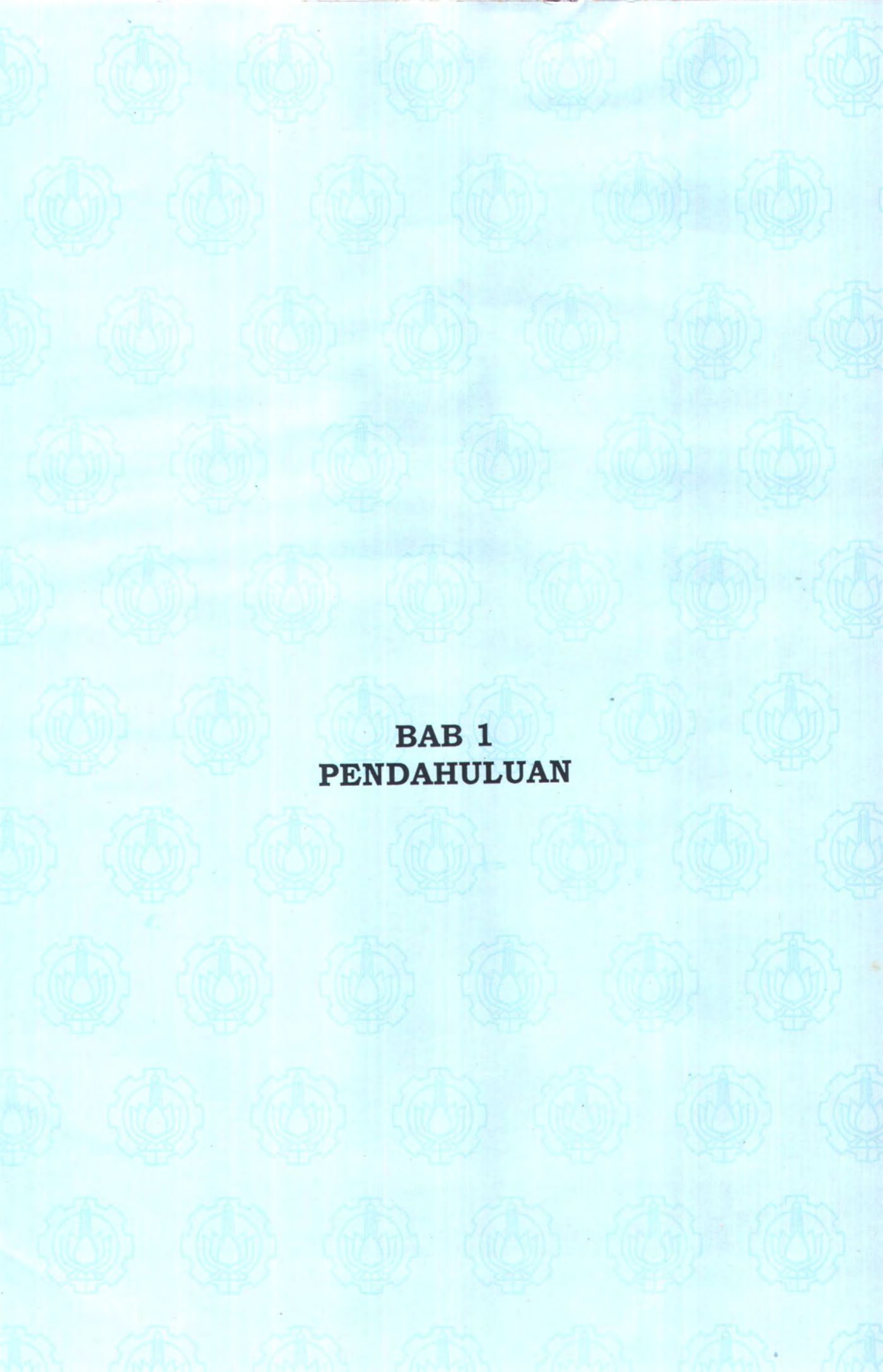
DAFTAR GAMBAR

Gambar 2-1 : Indikator Kerja dan Kerugian Produksi.....	10
Gambar 2-2 : Waktu Produksi dan Non Produksi	11
Gambar 2-3 : Web dinamis dengan ASP	18
Gambar 2-4 : Skema sistem database sederhana	19
Gambar 2-5 : Diagram-diagram yang didefinisikan dalam UML	21
Gambar 2-6 : Notasi pada <i>use case diagram</i>	24
Gambar 3-1 : Aliran kerja dari proses TPM	27
Gambar 3-2 : <i>Actor</i> yang terlibat dalam sistem	30
Gambar 3-3 : <i>Use Case Diagram</i> untuk <i>Actor Administrator</i>	31
Gambar 3-4 : <i>Use Case Diagram</i> untuk <i>Actor Management</i>	32
Gambar 3-5 : <i>Use Case Diagram</i> untuk <i>Actor Maintenance</i>	33
Gambar 3-6 : <i>Use Case Diagram</i> untuk <i>Actor Production</i>	34
Gambar 3-7 : <i>Use Case Diagram</i> untuk <i>Actor Data Entry</i>	35
Gambar 3-8 : <i>Activity Diagram</i> untuk <i>maintenance</i> data pengguna.....	36
Gambar 3-9 : <i>Activity Diagram</i> untuk <i>maintenance</i> database master.....	36
Gambar 3-10 : <i>Activity Diagram</i> untuk melihat hasil analisa.....	37
Gambar 3-11 : <i>Activity Diagram</i> untuk menentukan tipe glass.....	38
Gambar 3-12 : <i>Activity Diagram</i> untuk penjadwalan PM	39
Gambar 3-13 : <i>Activity Diagram</i> untuk mengisi data CM.....	40
Gambar 3-14 : <i>Activity Diagram</i> untuk mengisi data PDM	41
Gambar 3-15 : <i>Activity Diagram</i> penjadwalan produksi <i>glass</i>	42
Gambar 3-16 : <i>Activity Diagram</i> mengisi operasional mesin.....	43
Gambar 3-17 : <i>Activity Diagram</i> menetapkan target operasional.....	43
Gambar 3-18 : <i>Activity Diagram</i> memasukkan data operasional harian.....	44
Gambar 3-19 : <i>Sequence Diagram</i> untuk <i>maintenance</i> data user.....	46
Gambar 3-20 : <i>Sequence Diagram</i> untuk <i>maintenance</i> data <i>glass</i>	47
Gambar 3-21 : <i>Sequence Diagram</i> melihat hasil analisa.....	48
Gambar 3-22 : <i>Sequence Diagram</i> untuk CM / PDM.....	49
Gambar 3-23 : <i>Sequence Diagram</i> untuk PM.....	50
Gambar 3-24 : <i>Segeunce Diagram</i> untuk penjadwalan produksi <i>glass</i>	51
Gambar 3-25 : <i>Sequence Diagram</i> untuk mengisi operasional mesin.....	52
Gambar 3-26 : <i>Sequence Diagram</i> untuk menetapkan target operasional.....	53
Gambar 3-27 : <i>Sequence Diagram</i> data produksi harian.....	54
Gambar 3-28 : Desain database aplikasi.....	55
Gambar 3-29 : Diagram aliran sistem untuk administrator	67
Gambar 3-30 : Diagram aliran sistem untuk <i>management</i>	68
Gambar 3-31 : Diagram aliran sistem untuk <i>maintenance</i>	69
Gambar 3-32 : Diagram aliran sistem untuk <i>production</i>	69
Gambar 3-33 : Diagram aliran sistem untuk <i>data entry</i>	70
Gambar 3-34 : <i>Script</i> koneksi ke database.....	72
Gambar 3-35 : Tampilan halaman login	72
Gambar 3-36 : <i>Script</i> proses login.....	73
Gambar 3-37 : Tampilan antar muka pengguna	74
Gambar 3-38 : Tampilan untuk menampilkan data dalam tabel.....	74

Gambar 3-39 : <i>Script</i> menampilkan data dalam tabel.....	75
Gambar 3-40 : Tampilan untuk menampilkan data secara detail	75
Gambar 3-41 : <i>Script</i> menampilkan data secara detail	76
Gambar 3-42 : Tampilan halaman untuk menambah data.....	76
Gambar 3-43 : <i>Script</i> menambah data baru	77
Gambar 3-44 : Tampilan halaman untuk mengubah data.....	77
Gambar 3-45 : <i>Script</i> untuk meng- <i>update</i> data.....	78
Gambar 3-46 : <i>Script</i> untuk menghapus data.....	78
Gambar 3-47 : <i>Script</i> untuk <i>view Total Machine Hours</i>	79
Gambar 3-48 : <i>Script</i> untuk <i>view Total PM break down</i>	79
Gambar 3-49 : <i>Script</i> untuk <i>view Total CM break down</i>	79
Gambar 3-50 : <i>Script</i> untuk <i>view Total PDM break down</i>	79
Gambar 3-51 : <i>Script</i> total data harian produksi.....	80
Gambar 3-52 : Tampilan halaman hasil analisa TPM	80
Gambar 3-53 : <i>Script</i> menampilkan Data Target.....	81
Gambar 3-54 : Tampilan halaman untuk <i>Data Flag</i>	81
Gambar 3-55 : Tampilan untuk halaman <i>Machine Technical Statistics</i>	82
Gambar 3-56 : <i>Script</i> untuk total CM berdasarkan jenisnya.....	82
Gambar 3-57 : <i>Script</i> untuk total PM berdasarkan jenisnya.....	82
Gambar 3-58 : <i>Script</i> untuk total PDMD berdasarkan jenisnya	83
Gambar 3-59 : Tampilan untuk halaman <i>Production Performance</i>	83
Gambar 3-60 : <i>Script</i> membuat grafik dengan <i>office web component</i>	84
Gambar 3-61 : Tampilan halaman grafik.....	84
Gambar 3-62 : <i>Script</i> untuk menentukan nilai OEE	85
Gambar 3-63 : <i>Script</i> untuk menentukan nilai <i>Overall Utilisation</i>	86
Gambar 3-64 : <i>Script</i> untuk menentukan nilai <i>maintenance</i>	87

DAFTAR TABEL

Tabel 3-1 : Tabel groups	56
Tabel 3-2 : Tabel userLogin.....	56
Tabel 3-3 : Tabel dayProd.....	57
Tabel 3-4 : Tabel PDM	57
Tabel 3-5 : Tabel PM.....	58
Tabel 3-6 : Tabel CM.....	59
Tabel 3-7 : Tabel prodType	60
Tabel 3-8 : Tabel dailyProd	60
Tabel 3-9 : Tabel dailyProdType	61
Tabel 3-10 : Tabel machProd.....	62
Tabel 3-11 : Tabel machProdDet.....	63
Tabel 3-12 : Tabel prodTarget	64
Tabel 3-13 : Tabel ProdTypeTarget.....	65
Tabel 3-14 : Tabel GlassType.....	65
Tabel 4-1 : Tabel rencana produksi untuk minggu ke 30	91
Tabel 4-2 : Tabel penjadwalan PM.....	92
Tabel 4-3 : Tabel data <i>breakdown</i>	92
Tabel 4-4 : Tabel data produksi	92
Tabel 4-5 : Tabel data produksi untuk <i>glass</i> tipe A60	93
Tabel 4-6 : Tabel <i>data flag</i> minggu ke 30.....	94
Tabel 4-7 : Tabel <i>technical statistic machine</i> minggu ke 30	95
Tabel 4-8 : Tabel <i>production performance</i> minggu ke 30	95
Tabel 4-9 : Tabel TPM untuk minggu ke 30.....	97



BAB 1
PENDAHULUAN

BAB 1

PENDAHULUAN

Pada bab ini akan dijelaskan mengenai beberapa hal dasar yang ada dalam Tugas Akhir ini yang meliputi latar belakang, perumusan masalah, tujuan dan manfaat pembuatan Tugas Akhir serta metodologi dan sistematika pembuatan buku Tugas Akhir. Dari uraian dibawah ini diharapkan gambaran umum Tugas Akhir bisa dipahami dengan baik.

1.1 Latar Belakang

Saat ini dengan kompetisi waktu dalam dunia industri yang sangat ketat, waktu sangat berharga dalam proses produksi. Karena hanya dalam hitungan menit suatu pabrik dapat menghasilkan ratusan produk, maka dapatlah dibayangkan apabila pabrik tersebut mengalami gangguan produksi dalam beberapa menit, jam bahkan hari maka sangatlah besar kerugian yang akan dialami. Oleh karena itu diperlukan suatu metoda untuk melakukan perhitungan Kerugian Produksi dan Indikator Kinerja, sehingga dapat diketahui sejauh mana pabrik tersebut mengalami kerugian dan bagaimana kinerja dari pabrik tersebut.

Salah satu cara untuk melakukan analisa Kerugian Produksi dan Indikator Kinerja adalah melalui TPM (*Total Productive Maintenance*). TPM mempunyai tujuan untuk memaksimalkan efektivitas dan berusaha untuk mengurangi kerugian produksi. Dengan adanya TPM ini maka pabrik dapat mengetahui seberapa besar efektifitas produksinya.

Saat ini proses *maintenance* tidak lagi merupakan kegiatan yang tidak menghasilkan nilai tambah ataupun keuntungan terhadap perusahaan. Karena TPM juga mengintegrasikan program CM (*Corrective Maintenance*), PM (*Preventive Maintenance*) maupun PDM (*Predictive maintenance*).

Penerapan TPM sangat fleksibel sehingga dapat disesuaikan dalam berbagai bidang. TPM tidak hanya berguna pada pabrik-pabrik saja, tetapi juga dibidang konstruksi, perawatan bangunan, transportasi dan dalam berbagai macam bidang yang lain.

1.2 Perumusan Masalah

Permasalahan yang diangkat dalam Tugas Akhir ini adalah :

1. Bagaimana menentukan besarnya Kerugian Produksi dan Indikator Kinerja melalui TPM (*Total Productive Maintenance*).
2. Bagaimana mengintegrasikan *Corective Maintenance*, *Preventive Maintenance* dan *Predictive Maintenance* dalam *Total Productive Maintenance* sehingga dapat mengurangi Kerugian Produksi dan meningkatkan Indikator Kinerja.
3. Bagaimana hasil perhitungan Kerugian Produksi dan Indikator Kinerja dapat digunakan untuk mendukung pengambilan keputusan dalam meningkatkan efektivitas produksi.

1.3 Batasan Permasalahan

Dari permasalahan-permasalahan diatas, maka batasan dalam Tugas Akhir ini adalah :

1. Data yang digunakan dalam proses *Total Productive Maintenance* ini diperoleh dari *Lamp Component Factory* - PT Philips Ralin Electronics Surabaya.
2. Sistem ini hanya digunakan untuk menganalisa *Total Productive Maintenance* yang berhubungan dengan proses Produksi dan *Maintenance*.
3. Sistem pendukung keputusan yang dipakai adalah sistem pohon keputusan sederhana.
4. Sistem dibangun dengan menggunakan database SQL server 2000 dan aplikasi web ASP.

1.4 Tujuan Pembuatan Tugas Akhir

Tujuan dari pembuatan Tugas Akhir ini adalah untuk membuat aplikasi berbasis web *Total Productive Maintenance (TPM)* yang digunakan untuk menganalisa Kerugian Produksi dan Indikator Kinerja dari suatu pabrik atau perusahaan dan mendukung pengambilan keputusan untuk meningkatkan dan memaksimalkan kinerja dari pabrik tersebut.

1.5 Metodologi Pembuatan Tugas Akhir

Metodologi yang digunakan dalam pembuatan tugas akhir ini adalah sebagai berikut :

1. Studi Literatur dan Pemahaman Sistem

Tahap ini merupakan tahap pengumpulan informasi yang dipergunakan untuk pembuatan sistem. Informasi yang diperlukan diperoleh dengan membaca literature-literatur yang berhubungan untuk kelancaran proses pengerjaan selanjutnya.

2. Pengumpulan dan Analisa Data

Tahap ini merupakan tahap pengumpulan data-data yang diperlukan dalam pembuatan system. Selanjutnya data-data yang terkumpul dianalisa berdasarkan kriteria-kriteria yang menunjang.

3. Pengembangan Sistem

Tahap ini merupakan tahap yang paling banyak memerlukan waktu karena meliputi beberapa proses yaitu perancangan system, pembuatan prototipe sistem dan terakhir pembangunan sistem itu sendiri.

4. Evaluasi dan Revisi Sistem

Dalam tahap ini dilakukan evaluasi terhadap sistem yang telah selesai dikembangkan dengan tujuan untuk menemukan kesalahan-kesalahan yang mungkin terjadi serta melakukan perbaikan-perbaikan bagi kesempurnaan sistem tersebut.

5. Penyusunan Buku Tugas Akhir

Tahap penulisan buku Tugas Akhir bertujuan untuk menuliskan informasi-informasi mengenai sistem yang telah dibangun. Tahap ini sekaligus merupakan tahap terakhir dari pembentukan sistem secara keseluruhan.

6. Revisi Buku Tugas Akhir

Dalam tahap ini dilakukan perbaikan-perbaikan atau penambahan-penambahan yang diperlukan terhadap buku Tugas Akhir yang telah dibuat.

1.6 Sistematika Penulisan

Sistematika pembahasan mengenai perangkat lunak yang dibuat dalam Tugas Akhir ini disusun dalam beberapa bab, yang dijelaskan sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini berisi pendahuluan yang memuat latar belakang pembuatan sistem informasi, perumusan masalah, pembatasan masalah, tujuan dan manfaat yang bisa diperoleh dalam Tugas Akhir ini serta sistematika dan metodologi yang digunakan dalam pembuatan buku Tugas Akhir.

BAB 2 DASAR TEORI

Bab ini berisi mengenai konsep dan teori pembelajaran yang menjadi landasan pembuatan Tugas Akhir ini serta langkah-langkah apa saja yang diperlukan dalam desain dan implementasi sistem. Dasar teori disini meliputi pengertian tentang aliran kerja beserta penggunaan dan penerapannya. Kemudian dijelaskan mengenai teknologi web yang meliputi HTML (*Hyper Text Markup Language*), ASP (*Active Server Pages*), JavaScript. Selain itu dasar teori mengenai teknologi database yang meliputi SQL (*Structured Query Language*), dan DDL (*Data Definition Language*). Dijelaskan juga mengenai teknologi pemodelan sistem, UML (*Unified Modelling Language*).

BAB 3 DESAIN SISTEM

Bab ini berisikan mengenai perancangan sistem, yang meliputi deskripsi singkat sistem informasi, komponen-komponen sistem, pihak-pihak yang terlibat dalam pembuatan dan implementasi sistem informasi, perancangan data dan


perancangan proses. Juga digambarkan mengenai desain sistem dan desain antar muka.

BAB 4 IMPLEMENTASI DAN UJI COBA SISTEM

Bab ini berisikan bagaimana implementasi dari sistem informasi yang telah di buat berdasarkan desain yang telah dibuat sebelumnya. Juga dijelaskan mengenai hasil uji coba dan implementasi serta evaluasi terhadap sistem informasi tersebut.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dan saran untuk pengembangan lebih lanjut dalam upaya memperbaiki kelemahan yang ada guna mendapatkan hasil yang lebih akurat.



BAB 2
DASAR TEORI

BAB 2

DASAR TEORI

2.1 Total Productive Maintenance

Total Productive Maintenance (TPM) adalah program pemeliharaan yang melibatkan konsep baru dalam pemeliharaan pabrik dan perlengkapan. Definisi secara lengkap dari TPM memasukkan lima unsur berikut ini [SN]:

1. TPM bertujuan untuk memaksimalkan keefektifan dari peralatan (efektifitas secara keseluruhan)
2. TPM membangun sistem yang menyeluruh dari *Preventive Maintenance* untuk keseluruhan waktu hidup peralatan.
3. TPM diterapkan oleh berbagai macam departemen (teknik, operasi, *maintenance*).
4. TPM melibatkan setiap pekerja, mulai dari manajemen pusat sampai pekerja kasar.
5. TPM di dasarkan pada peningkatan dari *Preventive Maintenance* melalui manajemen motivasi dan aktifitas mandiri melalui grup kecil.

Kata “Total” dalam “*Total Productive Maintenance*” mempunyai tiga arti yang menggambarkan keunggulan-keunggulan yang pokok dari TPM [SN]:

1. *Total Effectiveness* (mengacu pada poin 1 diatas) yang mengindikasikan pencarian TPM untuk efisiensi ekonomi dan keuntungan.
2. *Total Maintenance System* (poin 2) memasukkan *Maintenance Prevention (MP)* dan *Maintenance Improvement (MI)* dalam *Preventive maintenance*.

3. *Total Participation of All Eemployees* (poin 3,4 dan 5) termasuk *outonomous maintenance* oleh operator melalui aktifitas grup kecil.

Tujuan TPM adalah untuk meningkatkan produktivitas sementara disisi yang lain meningkatkan moral pekerja dan kepuasan kerja. TPM mirip dengan program *Total Quality Management (TQM)* atau bisa dikatakan merupakan pengembangan dari TQM dengan memasukkan unsur *maintenance* dalam proses produksi.

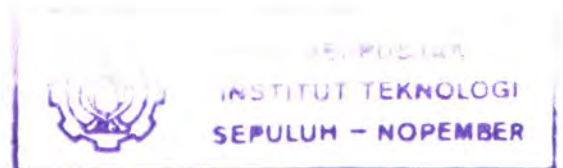
TPM membangun hubungan kuat antara *Maintenance* dan *Productivity*, TPM akan menunjukkan bahwa dengan pemeliharaan peralatan dengan baik akan menghasilkan produktivitas yang lebih tinggi. TPM menjadikan *maintenance* sebagai fokus dan bagian penting dari bisnis perusahaan.

Dengan konsep TPM, paradigma *maintenance* bukan lagi sebagai kegiatan tanpa nilai tambah. Dengan TPM, *down time* untuk *maintenance* dapat dijadwalkan sebagai bagian dari kelangsungan manufaktur dan tidak lagi dianggap sebagai proses perbaikan mesin saat *break down* saja. Dengan penanganan demikian, reduksi keadaan darurat dan *maintenance* tak terjadwal dapat dilakukan seoptimal mungkin.

2.1.1 Maintenance

2.1.1.1 Preventive Maintenance

Preventive Maintenance adalah proses *maintenance* yang dijadwalkan. Proses *maintenance* ini dilakukan sebelum mesin tersebut mengalami kerusakan. *Preventive maintenance* ini bertujuan agar mesin dapat berproduksi dengan baik dan mengurangi terjadinya kerusakan yang tidak terduga.



2.1.1.2 Corrective Maintenance

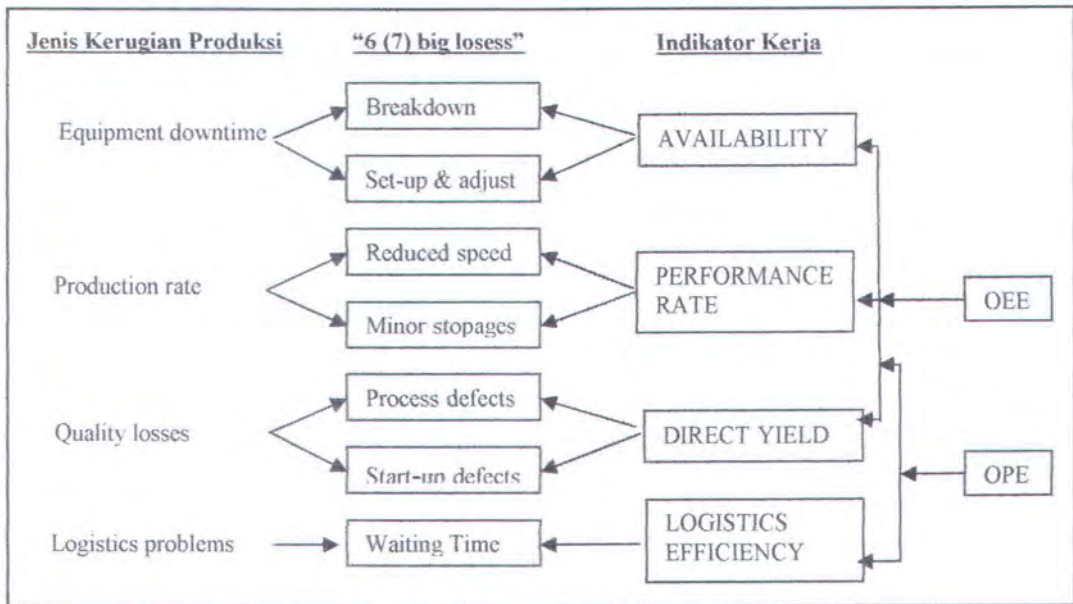
Correcive maintenance adalah proses *maintenance* yang dilakukan pada saat mesin mengalami kerusakan. *Corrective maintenance* mempunyai tingkat prioritas, prioritas tertinggi diberikan pada *maintenance* yang dilakukan pada mesin yang mengalami kerusakan dan menyebabkan proses produksi terhenti, prioritas yang lebih rendah diberikan apabila mesin tersebut masih dapat memproduksi dan proses *maintenance* dapat dilakukan pada saat mesin tersebut berjalan.

2.1.1.3 Predictive Maintenance

Predictive maintenance adalah proses *maintenance* yang dilakukan berdasarkan pembacaan parameter tertentu. Parameter ini bisa bermacam-macam, misalnya adalah apabila mesin sudah mengalami penurunan kecepatan produksi karena roda giginya sudah mulai aus, maka segera dilakukan proses *maintenance* walaupun proses *maintenance* tersebut belum dijadwalkan melalui *Preventive maintenance* maupun mesin tersebut tidak berhenti beroperasi.

2.1.2 Indikator Kerja dan Kerugian Produksi

TPM mengategorikan kerugian produksi kedalam kelompok yang berhubungan dengan tidak beroperasinya mesin, berhubungan dengan tingkat kecepatan produksi dan yang berhubungan dengan penurunan kualitas. Yang menjadi dasarnya adalah jadwal waktu produksi dari mesin. Kerugian produksi dibagi kedalam “*Six Big Losses*”, yang menetapkan nilai dari Indikator Kinerja, “*Availability*”, “*Performance Rate*” dan “*Direct Yield*”. Total dari indikator kerja TPM disebut dengan OEE (*Overall Equipment Effectiveness*).



Gambar 2-1 : Indikator Kerja dan Kerugian Produksi

2.1.3 Definisi dasar yang berhubungan dengan waktu produksi

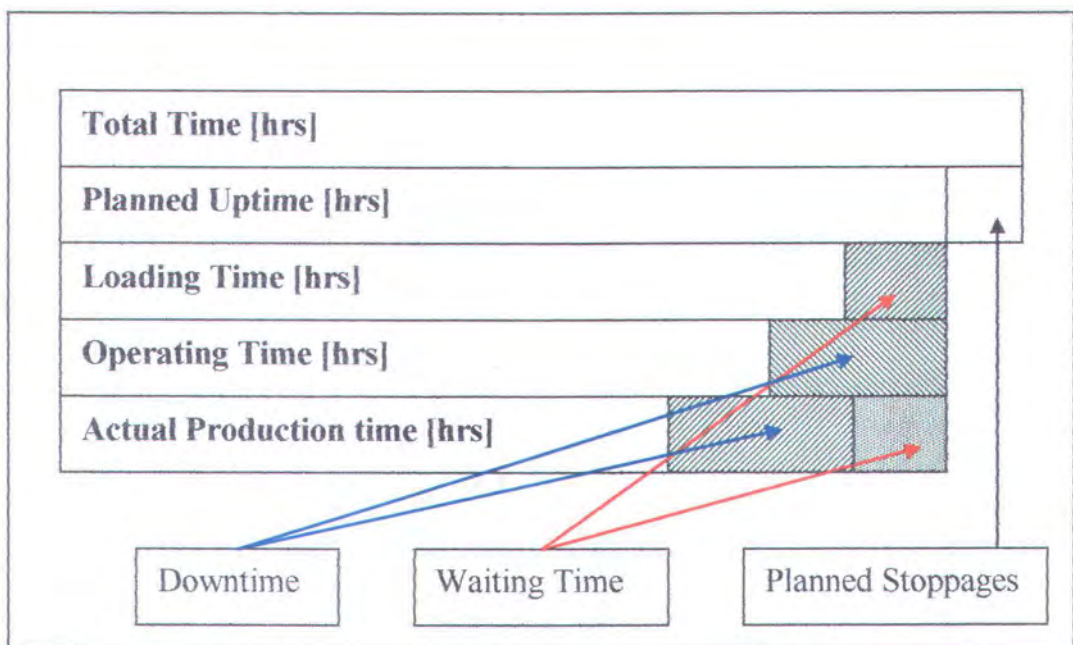
Perhitungan dimulai dari *planned uptime* dari mesin. Untuk menghindari salah pengertian, akan dijelaskan beberapa definisi yang berhubungan dengan waktu produksi dan non produksi.

- *Total Time* : jumlah jam dalam jangka waktu tertentu.

Contoh : Feb 2003, terdiri atas minggu 200306-200309 = 4 minggu * 7 hari * 24 jam = 672 jam.

- *Planned Stoppages* : rencana mesin tidak bekerja, termasuk hari libur dan karena adanya proses *maintenance* (yang telah dijadwalkan).
- *Planned Uptime* : total waktu kerja yang direncanakan.
- $Planned\ Uptime = Total\ Time - Planned\ Stoppages$

- *Operating Time* : jumlah jam yang tersedia untuk mesin melakukan proses produksi.
 - $Operating\ Time = Planned\ Uptime - Downtime$
- *Downtime* : pemberhentian proses produksi yang tidak direncanakan yang hanya dikarenakan oleh mesin yang rusak, *set-up* dan penyetelan.
- *Loading Time* : total waktu kerja.
 - $Loading\ Time = Planned\ Uptime - Waiting\ Time$
- *Waiting Time* : waktu yang diperlukan untuk menunggu datangnya material, peralatan, karyawan dan lain-lain.
- *Actual Production Time* : jumlah jam dari proses produksi.
 - $Actual\ Production\ Time = Planned\ uptime - Waiting\ Time - Downtime$
 - $= Loading\ Time - Down\ Time$



Gambar 2-2 : Waktu Produksi dan Non Produksi

2.1.4 Availability

$$AV = \text{Availability} = \frac{\text{Loading Time} - \text{Down Time}}{\text{Loading Time}}$$

$$= \frac{\text{Actual Production Time}}{\text{Loading Time}}$$

- Satuan yang digunakan untuk perhitungan *availability* adalah jam [hrs]

Dua “*Big Losses*” yang mempengaruhi adalah :

- Break Down* mesin, dimana mesin yang rusak menyebabkan aliran produksi berhenti.
- Set-up* dan Penyetelan, contohnya apabila terjadi pergantian tipe produksi harus ada penyesuaian terhadap mesin termasuk didalamnya adalah pemanasan mesin sebelum memulai produksi.

2.1.5 Direct Yield (Quality Rate)

$$DY = \text{Direct Yield} = \frac{\text{Produced Unit} - \text{Defect}}{\text{Produced Unit}}$$

- Satuan yang digunakan dalam perhitungan *Direct Yield* adalah buah [pcs]
- Produced Unit adalah jumlah produk yang dihasilkan selama proses produksi.

Dua “*Big Losses*” yang mempengaruhi adalah :

- Start up Defect* adalah produk yang ditolak yang dihasilkan selama proses *star-up*, hal ini biasanya terjadi kerana mesin belum mencapai kondisi yang stabil untuk berproduksi.
- In-process Defect* adalah produk yang ditolak yang dihasilkan selama proses produksi.

2.1.6 Performance Rate

$$\begin{aligned} \text{PR} = \text{Performance Rate} &= \text{Net Operating Rate} \times \text{Speed Ratio} \\ &= \frac{\text{Actual Output}}{\text{Potensial Output}} \end{aligned}$$

$$\text{Net Operating Rate} = \frac{\text{Total Input} \times \text{Actual Cycle Time}}{\text{Actual Production Time}}$$

$$\text{Speed Ratio} = \frac{\text{Ideal Cycle Time}}{\text{Actual Cycle Time}}$$

Dua “*Big Losses*” yang mempengaruhi adalah :

- a. *Minor Stop* terjadi ketika proses produksi diganggu dengan malfungsi sementara atau ketika mesin menganggur. *Minor Stop* dicerminkan dalam *Net Operating Rate*.
- b. *Speed Loss* mengacu pada perbedaan di antara desain kecepatan mesin dan kecepatan operasional sesungguhnya. *Speed Loss* dicerminkan dalam *Speed Ratio*.

2.1.7 Logistic Efficiency

Logistic Efficiency dipengaruhi oleh *Waiting Time*. *Waiting time* adalah waktu yang hilang karena masalah-masalah yang berhubungan dengan persediaan seperti menunggu material, operator, bahan kimia dan lain-lain.

2.1.8 OEE (Overall Equipment Effectiveness)

Overall Equipment Effectiveness (OEE) merupakan total indikator kerja yang berupa *Availability*, *Direct Yield* dan *Performance Rate*. OEE dapat diukur dengan menggunakan rumus :

$$\text{OEE} = \text{Availability} \times \text{Direct Yield} \times \text{Performance Rate}$$

2.1.9 OPE (Overall Production Effectiveness)

Sedangkan *Overall Production Effectiveness (OPE)* yang merupakan hasil perhitungan OEE dengan memasukkan unsur *Logistic Efficiency*. OPE dapat diukur dengan menggunakan rumus :

$$OPE = OEE \times \text{Logistic Efficiency}$$

2.1 Teknologi Web

2.1.1 HTML (Hypertext Markup Language)

HTML (*Hypertext Markup Language*) merupakan suatu *script* dimana kita bisa menampilkan informasi dan daya kreasi kita lewat internet. HTML sendiri adalah suatu dokumen teks biasa yang mudah untuk dimengerti dibandingkan bahasa pemrograman lainnya, dan karena bentuknya itu maka HTML bisa dibaca oleh platform yang berlainan seperti *Windows*, *UNIX*, dan lain-lain. Walaupun berbentuk dokumen teks biasa, HTML memiliki perbedaan dengan dokumen lain seperti dokumen *Word*, misalnya. Perbedaan yang paling mencolok adalah, pada dokumen *Word* banyaknya karakter akan terbatas oleh besarnya kertas sehingga jika teks yang ada didalamnya banyak maka ia akan terdiri dari banyak halaman pula. Sedangkan HTML tak memiliki batasan teks, sehingga tidak ada pemisah halaman 1, 2 dan seterusnya [AG99].

HTML merupakan bahasa pemrograman fleksibel dimana kita bisa meletakkan *script* dari bahasa pemrograman lain seperti *Java*, *Visual Basic*, *C* dan lain-lain, jika HTML tersebut tidak dapat mendukung suatu perintah pemrograman tertentu. Browser tidak akan menampilkan kotak dialog '*syntax error*' jika terdapat penulisan kode yang keliru pada *script* HTML sepanjang

kode-kode yang kita tuliskan merupakan kode-kode HTML tanpa penambahan kode-kode dari luar seperti *Java*. Oleh karena itu jika terdapat ‘*syntax error*’ pada *script* HTML, efek yang paling jelas adalah HTML tersebut tidak akan ditampilkan pada jendela *browser*.

Hypertext dalam HTML berarti bahwa kita dapat menuju ke suatu tempat, misal website atau halaman *homepage* lain, dengan cara suatu *link* yang biasanya digarisbawahi atau diwakili oleh suatu gambar. Selain *link* ke *website* atau *homepage* halaman lain, *hypertext* ini juga mengijinkan kita untuk menuju salah satu bagian dalam teks itu sendiri.

Sedangkan *markup language* menunjukkan suatu fasilitas yang berupa tanda tertentu dalam *script* HTML dimana kita bisa men-set judul, garis, gambar dan lain-lain dengan perintah khusus. Secara umum dokumen HTML mempunyai beberapa bagian, yaitu :

2.1.1.1 Bagian HEAD

Digunakan untuk mengidentifikasi isi keseluruhan dari dokumen. Bagian ini diawali dengan *tag* `<HEAD>` dan diakhiri dengan *tag* `</HEAD>`. Informasi ini tidak ditampilkan sebagai bagian dari dokumen, tetapi berisi tentang dokumen itu sendiri.

2.1.1.2 Bagian BODY

Digunakan untuk menandai bahwa daerah dalam lingkup bagian ini merupakan isi dokumen. Elemen-elemen dalam *BODY* biasanya menggambarkan ekspresi secara visual di dalam *browser*. Cara penulisannya diawali dengan `<BODY>` dan diakhiri `</BODY>`.

2.1.1.3 Bagian FORM

Pada dasarnya *Form* digunakan untuk meminta masukan dari pengguna aplikasi untuk kemudian dikirim ke server. *Form* ini dapat berupa formulir isian, daftar atau jenis-jenis masukan standar lainnya. Nilai-nilai atau data yang diisikan disimpan dalam form, untuk kemudian disisipkan bersamaan dengan form yang dikirim dalam proses interaksi dengan server. Dengan kata lain salah satu fungsi dari *Form* adalah sebagai salah satu sarana komunikasi dengan program eksternal menggunakan sebuah aplikasi antar muka. Aplikasi tersebut bisa dikatakan bertugas sebagai *pipeline* yang dapat menghubungkan data antara dokumen web dengan program tersebut. Proses interaksi yang terjadi disini menggunakan aturan-aturan yang berlaku protocol *HTTP* [KU03].

Dalam terdapat atribut *METHOD* yang mempunyai dua nilai yaitu : *POST* dan *GET*. Metode *GET* mengirimkan data pada server dengan cara meletakkan data pada bagian akhir *URL* (*Uniform Resource Locator*). Sedangkan metode *POST* mengirimkan datanya secara terpisah. Jika data masukan banyak, metode *POST* akan lebih efektif [PY01].

2.1.2 DHTML (Dynamic Hypertext Markup Language)

Pengertian DHTML pada dasarnya adalah penggunaan objek model dan *CSS* (*Cascading Style Sheet*) atribut yang dimanipulasi secara dinamis melalui bahasa *scripting* [EF00]. DHTML menyediakan fasilitas bagi pengembang untuk membangun dokumen HTML yang sangat bagus secara visual dan dapat digunakan untuk berinteraksi dengan pengguna tanpa harus mengandalkan

program yang berjalan di server atau sekumpulan kode-kode rumit HTML untuk menghasilkan efek khusus dalam suatu halaman web [KU03].

2.1.3 VBScript

VBScript adalah bagian terbaru dari keluarga Visual Basic yang membawa penulisan *script* yang aktif (*active scripting*) ke sebagian besar lingkungan, mencakup penulisan pada sisi *client* maupun pada sisi *server*. Bahasa ini pada dasarnya merupakan bagian atau subset dari bahasa Basic.

VBScript mempunyai kemampuan dalam menambah unjuk kerja suatu halaman web antara lain validasi dan kalkulasi data, penyimpanan data atau memberi respon terhadap input yang diberikan oleh pengguna aplikasi. Selain itu bahasa pemrograman ini juga mendukung penggunaan OLE dan *ActiveX Control*. Dimana *ActiveX* merupakan suatu teknologi yang dikenalkan oleh *microsoft* yang berbasiskan COM agar dapat menambah *feature-feature* pada suatu halaman HTML dan juga sekaligus untuk menandingi teknologi *Java Applet* dan *Sun Microsystem* [EF00]. Sedangkan OLE (*Object Linking and Embedding*) mempunyai kemampuan dalam berinteraksi dengan sebuah file misalnya file teks, file sound dalam sebuah dokumen [KU03].

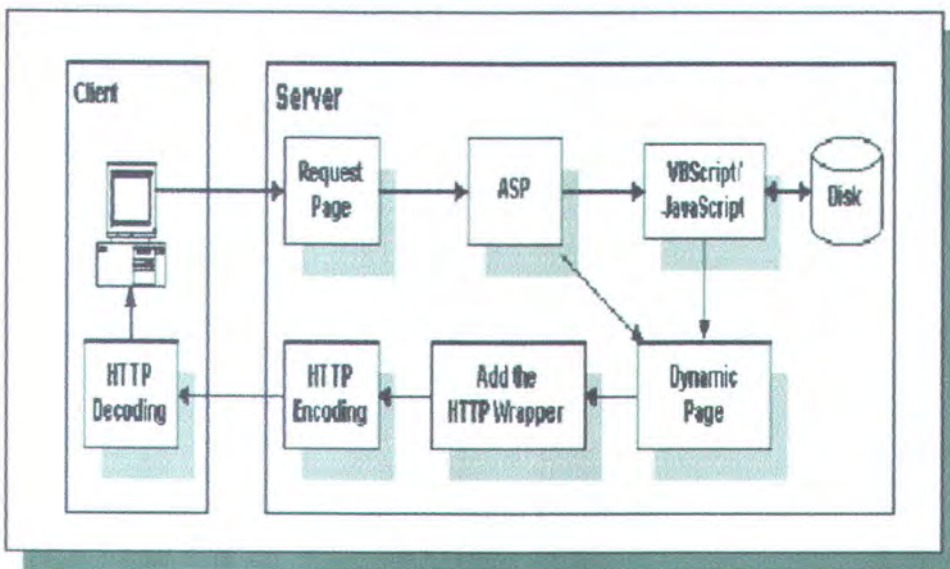
2.1.4 JavaScript

Pada awalnya *JavaScript* dibuat untuk membantu menciptakan suatu aplikasi web yang interaktif yang didukung oleh *Browser Netscape*. *JavaScript* sangat mudah digunakan, ringan dan dinamis. Pada dasarnya *JavaScript* sama dengan *VBScript* hanya *JavaScript* dijalankan di sisi server. Yang membedakan

adalah tag pembuka yaitu : `<Script Language= JavaScript >` sedangkan `VBScript <Script Language=VBScript>`.

2.1.5 ASP (Active Server Page)

Microsoft ® Active Server Page (ASP) adalah *server-side scripting environment* yang akan mempermudah pembuatan aplikasi berbasis web yang dinamis dan interaktif [KU03]. Disamping menggunakan *browser* untuk mengambil halaman, ASP juga menggunakan mesin yang lain –*web server*–sebelum mengembalikan hasilnya ke user dalam bentuk HTML [FA98].



Gambar 2-3 : Web dinamis dengan ASP

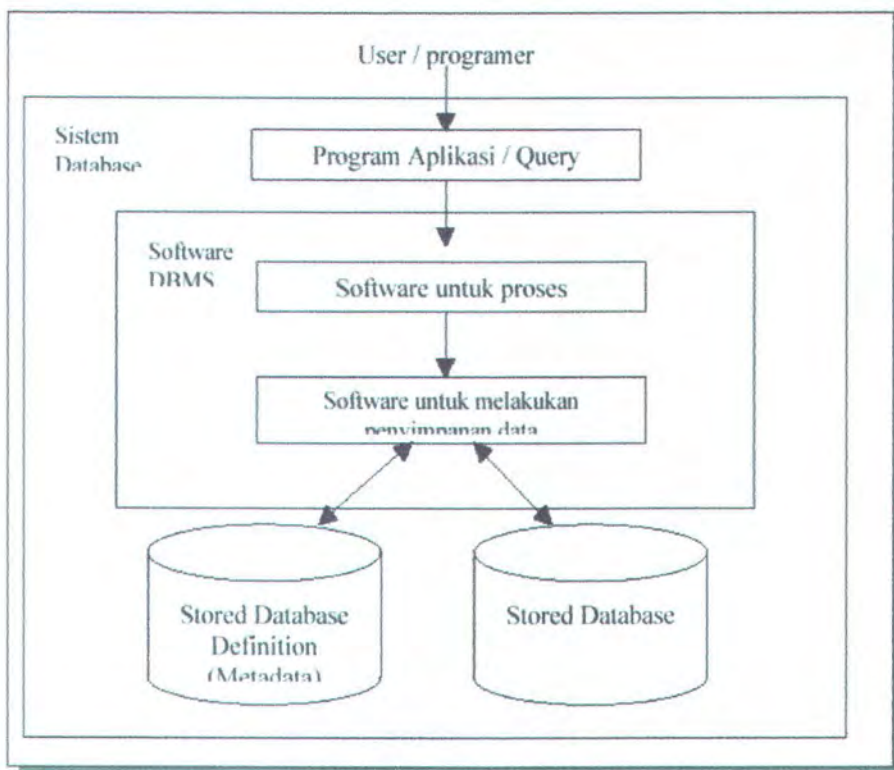
Tidak seperti HTML, ASP bukanlah sebuah bahasa. Bukan juga merupakan sebuah aplikasi. Pada dasarnya ASP menggabungkan halaman-halaman HTML, perintah-perintah *script*, serta komponen lainnya sehingga memudahkan dalam membuat dan memodifikasi web [KU01]. Asp didesain untuk digunakan secara bersama-sama dengan HTML untuk menciptakan halaman web yang dinamis. Bahkan sebenarnya ASP membentuk kode-kode HTML.

Untuk membedakan antara kode HTML dengan ASP tidaklah sulit. Karena *script* ASP biasanya diawali dengan “<%” dan diakhir dengan “%>”.

2.2 Data Base

2.2.1 Pengertian Secara Umum

Pada dasarnya pengertian database bersifat umum. El Masri mendefinisikan database sebagai sekumpulan dari data-data yang saling berhubungan [EM00]. Sementara DBMS (*Data Base Management System*) adalah kumpulan program yang digunakan user untuk membuat dan mengelola database. Sistem basis data adalah basis data dan sistem manajemen basis data [KU03].



Gambar 2-4 : Skema sistem database sederhana

2.2.2 SQL (Structured Query Language)

Bahasa SQL adalah sarana dasar yang dibutuhkan untuk mengakses dan memanipulasi data dari sebuah database relasional [RJ01]. Menguasai bahasa SQL adalah langkah pertama dan juga penting dalam memahami sebuah data base relasional. Dengan banyaknya perintah dan fungsi dalam SQL , kita akan dapat dengan mudah melakukan sebagian besar aktifitas yang berhubungan dengan manipulasi database.

Berdasarkan kepada fungsinya deklarasi atau perintah dalam SQL dibagi dalam dua kategori yaitu DDL dan DML [RJ01].

2.2.2.1 DDL (Data Definition Language)

DDL adalah bagian dari SQL yang digunakan untuk mendefinisikan data dan objek database [RJ01]. Apabila perintah ini digunakan, enrti akan dibuat ke dalam kamus data dari database server. Beberapa contoh perintahnya adalah :

- *Create Table* : Membuat table
- *Create Index* : Membuat Indeks
- *Alter Table* : Mengubah atau menyisipkan kolom ke dalam table
- *Drop Table* : Menghapus table dari database
- *Drop Index* : Menghapus indeks
- *Grant* : Memberikan izin akses kepada user.

2.2.2.2 DML (Data Manipulation Language)

DML adalah bagian SQL yang digunakan untuk memulihkan dan memanipulasi data. Perintah-perintah ini bertanggung jawab melakukan *query* dan

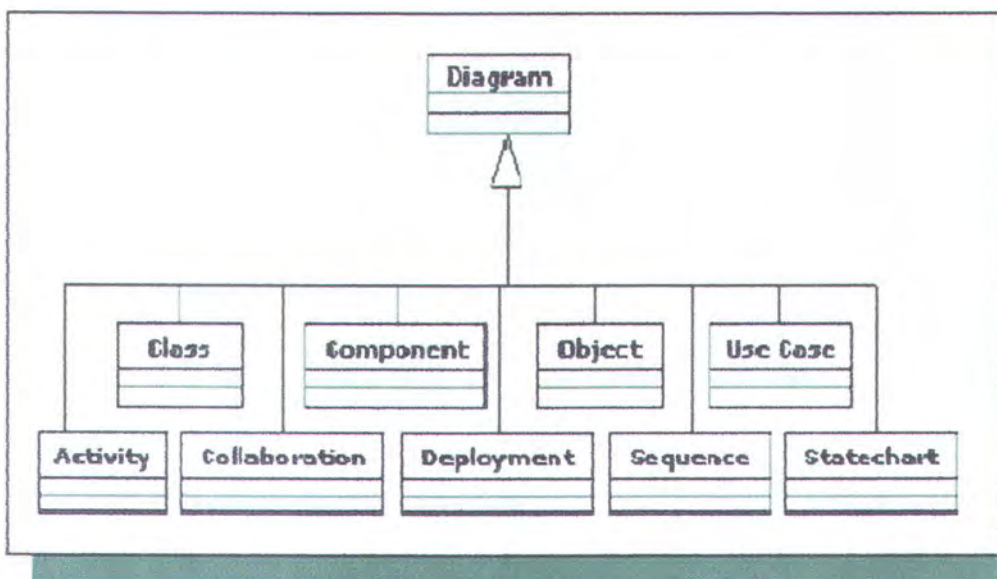
perubahan yang dilakukan dalam table [RJ01]. Beberapa perintah penting dalam kategori ini :

- *Select* : memilih data dari table atau view
- *Insert* : menyisipkan baris dari table
- *Delete* : menghapus baris dari table
- *Update* : mengubah isi kolom dari table

2.3 UML (Unified Modelling Language)

2.3.1 Definisi

UML (*Unified Modelling Language*) adalah bahasa untuk menspesifikasikan, memvisualisasikan dan membangun produk sistem perangkat lunak. UML sendiri merupakan notasi pemodelan sistem dengan menggunakan pendekatan yang berorientasi objek [UM99].



Gambar 2-5 : Diagram-diagram yang didefinisikan dalam UML

Selain itu UML juga merupakan standar industri untuk pemodelan yang berorientasi objek yang dibangun dari tiga metoda pendahulunya, yaitu metoda Booch (oleh Grady Booch), metoda OMT (*Object Modelling Technique* oleh James Rumbaugh) dan metoda OOSE (*Object Oriented Software Engineering* oleh Ivar Jacobson).

2.3.2 Diagram UML

Pada dasarnya UML mempunyai sembilan macam diagram sistem, yaitu :

- *Activity Diagram*

Diagram yang menggambarkan sifat / kelakuan suatu operasi sebagai rangkaian dari aktifitas yang dilakukan.

- *Class*

Diagram yang menggambarkan struktur dan kelakuan suatu *class* (*attribute / property* dan *operation method*) beserta hubungan antar *class*.

- *Collaboration*

Diagram yang berisi objek-objek beserta relasi antar objek yang menggambarkan interaksi / operasi antar objek.

- *Componen*

Diagram yang menggambarkan komponen-komponen beserta relasi ketergantungan antar komponen dalam suatu aplikasi.

- *Deployment*

Diagram yang menggambarkan penempatan komponen suatu aplikasi pada perangkat keras dalam suatu lingkungan sistem dimana aplikasi tersebut digunakan.

- *Object*

Merupakan bentuk lain dari *collaboration diagram* yang telah disederhanakan dengan tidak mengikutsertakan interaksi objek dengan sistem eksternal.

- *Use Case*

Diagram yang menggambarkan fungsi-fungsi yang disediakan oleh sistem dari sudut pandang lain.

- *Sequence*

Merupakan bentuk lain dari *collaboration diagram* dengan sudut pandang yang dilakukan berdasarkan urutan waktu.

- *State*

Diagram yang menggambarkan kelakuan suatu operasi sebagai serangkaian perubahan kondisi / situasi (*state*).

Pada dasarnya kesembilan diagram diatas dapat dikelompokkan ke dalam tiga kategori yaitu : statis, dinamis dan arsitektural.

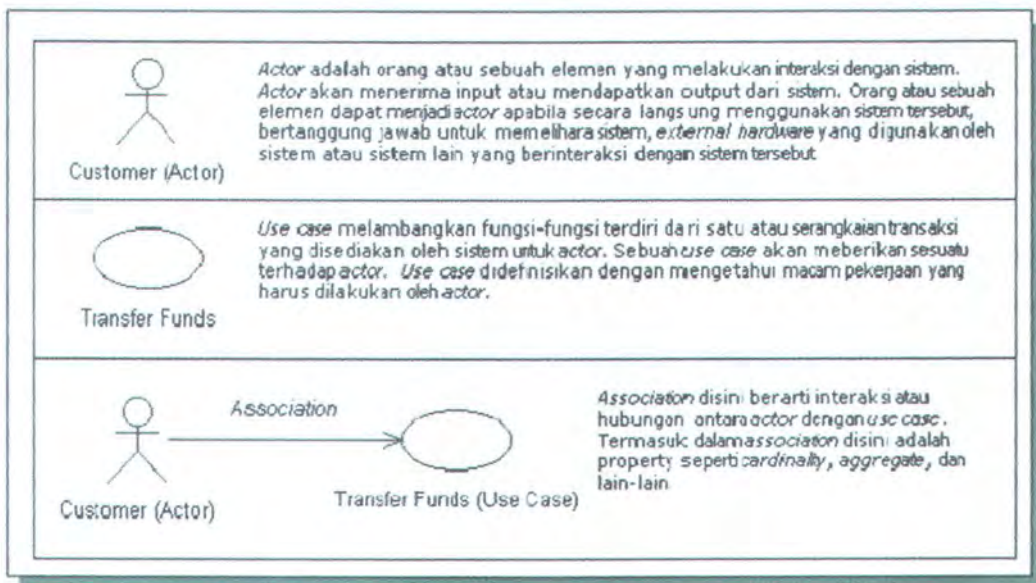
Sebuah diagram statis (*Static Diagram*) menggambarkan struktur dari suatu sistem dan fungsi serta tanggungjawabnya. Yang termasuk ke dalam diagram statis adalah *User Case*, *Class* dan *Object Diagram*.

Diagram dinamis (*Dynamic Diagram*) menggambarkan interaksi atau hubungan-hubungan yang didukung oleh sistem tersebut. Yang termasuk di dalam kategori diagram dinamis ini diantaranya *Activity*, *Collaboration*, *Sequence*, *State* dan *Use Case*.


Sedangkan diagram arsitektural (*Architectural Diagram*) menggambarkan realisasi dari sistem tersebut menjadi komponen-komponen yang dapat dijalankan dan dieksekusi. Yang termasuk didalamnya adalah *Component*, dan *Deployment Diagram*.

Pada kenyataannya, setiap proyek yang ingin digambarkan paling tidak akan memiliki minimum tiga diagram, yaitu : *Use Case*, *Class* dan *Sequence Diagram*.

Dalam suatu UML, diagram akan digambarkan sebagian atau secara keseluruhan karakteristik dari objek-objek sesuai dengan konteks diagram. Dokumentasi dari desain-desain yang dihasilkan pada saat pengembangan aplikasi tidak harus menggunakan semua tipe diagram UML diatas.



Gambar 2-6 : Notasi pada *use case diagram*



BAB 3
DESAIN DAN IMPLEMENTASI SISTEM

BAB 3

DESAIN DAN IMPLEMENTASI SISTEM

Pada bab ini berisi mengenai desain sistem yang akan dibuat. Untuk desain sistemnya menggunakan UML (*Unified Modelling Language*). Sedangkan *Case Tool* yang digunakan adalah *Rational Rose 2000 Enterprise Edition*. Dari permodelan ini diharapkan dapat menggambarkan dan menjelaskan proses-proses yang terjadi.

3.1 Definisi Sistem

Pada sub bab ini akan dijelaskan mengenai sistem yang telah ada dan rancangan sistem baru yang akan dibuat.

3.1.1 Sistem Yang Sudah Ada

Saat ini sistem yang digunakan untuk melakukan analisa Kerugian Produksi dan Indikator Kinerja masih dilakukan secara manual dan terpisah antara masing-masing departemen. Hal ini tentunya menyulitkan pihak *Management* untuk menggunakan hasil laporan-laporan tersebut untuk menentukan kebijakan perusahaan. Selain itu data-data yang tersebar pada masing-masing departemen akan memperlambat proses analisa dan pihak *Management* akan kesulitan untuk memperolehnya dibandingkan apa bila data tersebut diletakan secara terpusat. Secara umum pihak-pihak yang terlibat dalam sistem ini adalah sebagai berikut :

3.1.1.1 Management

Pihak *Management* terdiri atas para Direksi dan Menejer. Mereka adalah para pimpinan dan kepala dari masing-masing departemen. Pihak *management*

inilah yang memerlukan informasi-informasi yang diolah dari sistem ini agar dapat mempergunakannya untuk pengambilan keputusan terhadap kebijakan-kebijakan perusahaan, khususnya yang menyangkut masalah produksi dan *maintenance*.

3.1.1.2 Maintenance Department

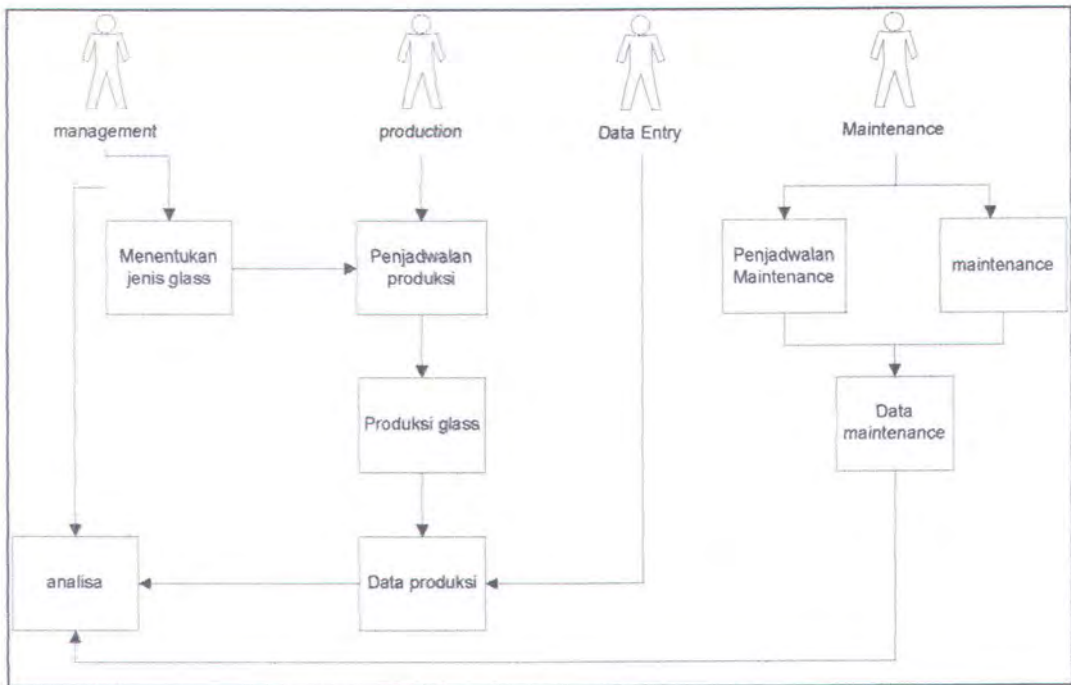
Departemen *Maintenance* adalah bagian yang mengurus masalah-masalah perawatan dan perbaikan mesin dan peralatan yang meliputi penjadwalan untuk melakukan perawatan dan perbaikan mesin dan peralatan yang rusak atau mengalami gangguan, pengecekan terhadap peralatan dan mesin, permintaan *spare part* dan lain-lain yang berhubungan dengan proses perawatan dan perbaikan.

3.1.1.3 Production Department

Sedangkan Departemen Produksi adalah bagian yang mengurus masalah-masalah produksi yang meliputi penjadwalan proses produksi, target-target produksi dan hal-hal lain yang berhubungan dengan proses produksi.

3.2 Desain Sistem

Dari sistem lama yang sudah ada dikembangkan sistem baru dengan mengintegrasikan seluruh bagian dari pihak-pihak yang berhubungan dengan proses produksi, proses *maintenance* dan pengambil kebijakan perusahaan sehingga diperoleh aliran sistem yang baru.



Gambar 3-1 : Aliran kerja dari proses TPM

3.2.1 Pembagian Sistem

Sistem yang akan dibuat terbagi menjadi dua bagian yaitu bagian sistem informasi produksi dan maintenance dan yang satunya lagi adalah proses hasil analisa dari proses produksi dan *maintenance*.

3.2.1.1 Sistem Informasi Produksi dan Maintenance

Sistem informasi produksi dan *maintenance* ini hanya terbatas dipergunakan di lingkungan PT. Philips Ralins Electronins Surabaya. Didalam sistem informasi ini akan melibatkan beberapa pihak, yaitu Administrator, *Management*, *Production Department*, dan *Maintenance Department*.

3.2.1.1.1 Administrator

Administrator adalah orang yang mempunyai hak akses terhadap semua bagian. Pada dasarnya administrator disini hanya ditujukan untuk mengatur

pengguna yang akan terhubung dengan sistem, seperti menambah pengguna baru, mengubah data pengguna yang sudah ada dan menghapus data pengguna.

3.2.1.1.2 Management

Pengguna ini mempunyai hak untuk menentukan tipe *glass* baru untuk diproduksi dan melihat hasil analisa dan perhitungan dari proses produksi dan *maintenance*. Penentuan jenis *glass* ini akan dipergunakan oleh departemen produksi untuk penjadwalan proses produksi sesuai dengan pesanan.

3.2.1.1.3 Maintenance Planner

Maintenance Planner adalah jenis pengguna yang bertugas untuk melakukan penjadwalan proses *maintenance* dan memasukkan data-data hasil proses *maintenance*.

3.2.1.1.4 Production Planner

Production Planner adalah pengguna yang mempunyai tugas untuk penjadwalan produksi, penjadwalan mesin, dan menentukan target dari proses produksi tersebut. Untuk target dari proses produksi tersebut dibagi menjadi dua, yaitu target produksi umum dan target produksi khusus. Target produksi khusus ini menyangkut performa dari mesin untuk memproduksi tipe *glass* tertentu.

3.2.1.1.5 Data Entry

Data Entry adalah pengguna yang bertugas untuk memasuk data produksi harian. Data produksi harian ini juga terbagi menjadi dua, yaitu data produksi umum dan data produksi khusus. Data produksi umum adalah data produksi menyangkut sistem produksi secara keseluruhan, sedangkan data produksi khusus

adalah data performa dari masing-masing mesin untuk memproduksi tipe glass yang berbeda-beda.

3.2.1.2 Analisa TPM

Analisa TPM akan menggunakan data-data dari proses produksi dan *maintenance* untuk memperoleh hasil OEE. Analisa TPM akan menghitung OEE, *Availability*, *Direct Yield*, *Performance Rate* dan *Logistic Efficiency*. Hasil dari analisa TPM ini berupa data perhitungan dan grafik.

3.2.2 Use Case View

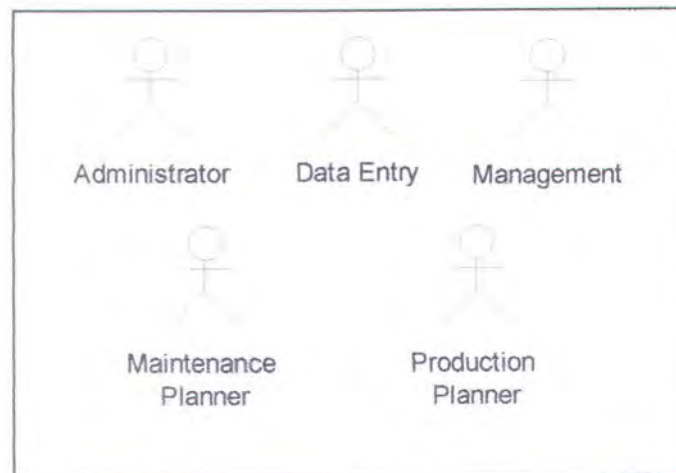
3.2.2.1 Actor

Actor bukan merupakan bagian dari sebuah sistem, tetapi mereka merepresentasikan seseorang atau sesuatu yang harus berinteraksi dengan sebuah sistem [QT02].

Seorang *actor* biasanya :

- Memberi input ke sistem
- Menerima input ke sistem
- Memberi dan menerima input dari dan ke sistem

Dalam pemodelan yang dilakukan terhadap sistem ini, yang pertama kali dilakukan adalah menentukan *actor* yang terlibat. Actor yang terlibat dalam sistem ini adalah Administrator, Management, Production, Maintenance dan Data Entry. Dalam sistem ini *actor* yang terlibat adalah :



Gambar 3-2 : Actor yang terlibat dalam sistem

3.2.2.2 Use Case Diagram

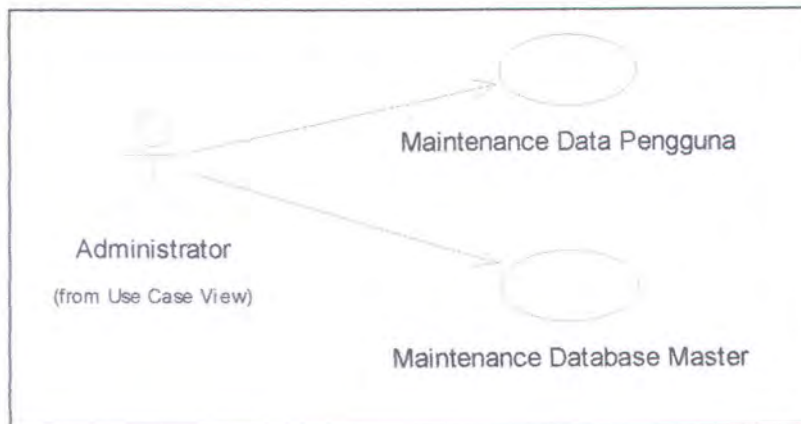
Sebuah *use case diagram* merupakan sebuah penampakan grafis dari beberapa atau keseluruhan *actors*, *use case*, dan juga interaksi mereka yang teridentifikasi untuk sebuah sistem [QT02]. Berikut ini akan dijelaskan beberapa interaksi yang terjadi antara *actors* dengan *use case* yang ada.

3.2.2.2.1 Administrator

Actor administrator mempunyai dua aktivitas utama, yang pertama adalah *maintanance* data pengguna dan *maintenance* database master. Untuk lebih jelasnya adalah sebagai berikut :

a. Maintenance data pengguna

Dalam aktivitas ini administrator mempunyai tanggung jawab untuk mengelola data pangguna. Mulai dari menambahkan data pengguna baru, mengubah data pengguna yang sudah ada sampai dengan menghapus data pengguna.



Gambar 3-3 : Use Case Diagram untuk Actor Administrator

b. Maintenance database master

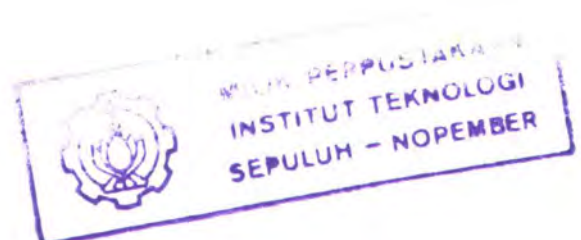
Sedangkan untuk aktivitas pada maintenance database master, administrator bertanggung jawab untuk mengelola semua database master meliputi master pengguna, master produksi, master *glass*, master *maintenance* dan master target produksi.

3.2.2.2.2 Management

Dalam *Use Case* untuk *actor management* terdapat dua buah aktivitas utama yaitu melihat hasil analisa dan menentukan tipe *glass*.

a. Melihat Hasil analisa

Dalam *use cse* melihat analisa ini *actor management* dapat melihat hasil-hasil perhitungan yang telah ada, hasil-hasil perhitungan ini berupa grafik dan kertas kerja. Hasil analisa ini ada empat macam yaitu TPM, *Technical Statistic Machine*, *Data Flag* dan *Production*. Sedangkan untuk grafiknya ada tiga macam yaitu grafik TPM, grafik *Maintenance* dan grafik *Production*.





Gambar 3-4 : Use Case Diagram untuk Actor Management

b. Menentukan tipe *glass*

Sedangkan untuk menentukan tipe *glass*, *actor maintenance* bertugas untuk mengelola data tipe *glass*. Termasuk didalamnya memasukkan data tipe *glass* baru, mengubah tipe *glass* yang udah ada dan menghapus tipe *glass* yang sudah ada.

3.2.2.2.3 Maintenance

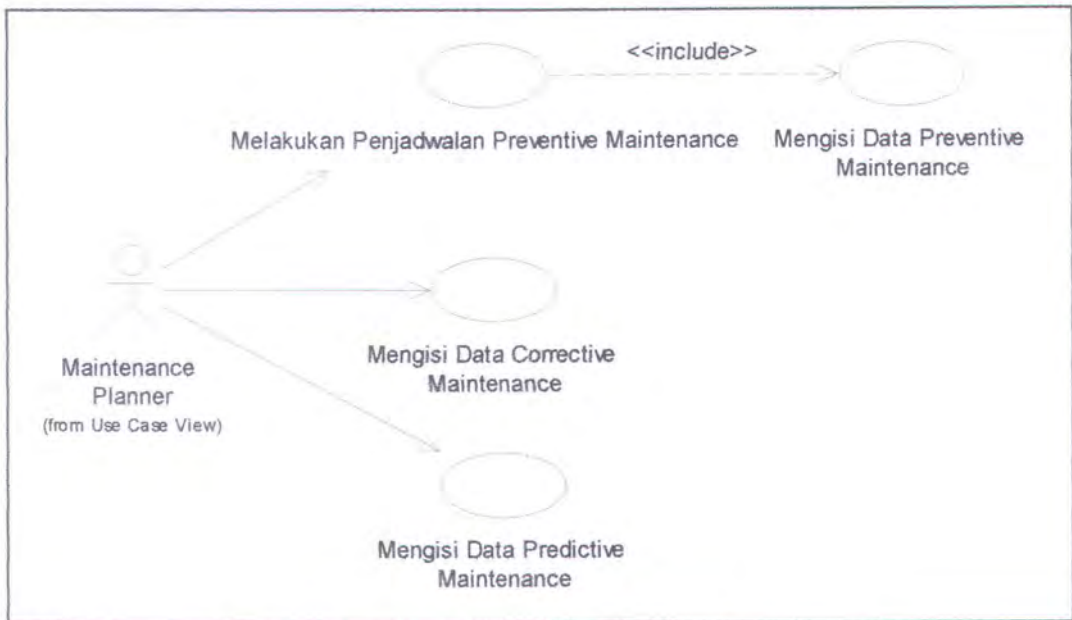
Dalam *use case* untuk *actor maintenance* terdapat tiga aktifitas utama yaitu sebagai berikut :

a. Menentukan penjadwalan PM (*Preventive Maintenance*).

Disini *actor maintenance* bertugas untuk menentukan penjadwalan pelaksanaan PM. Kemudian mengisi data pelaksanaan PM. Termasuk didalamnya memasukkan data baru, mengubah data yang sudah ada dan menghapus data.

b. Mengisi data CM (*Corrective Maintenance*)

Dalam mengisi data CM ini *actor maintenance* bertugas untuk mengelola data CM, termasuk memasukkan data baru, mengubah data yang sudah ada dan menghapus data.



Gambar 3-5 : Use Case Diagram untuk Actor Maintenance

c. Mengisi data PDM (*Predictive Maintenance*)

Actor maintenance dalam *use case* ini bertugas untuk mengelola data PDM, termasuk memasukkan data baru, mengubah data dan menghapus data.

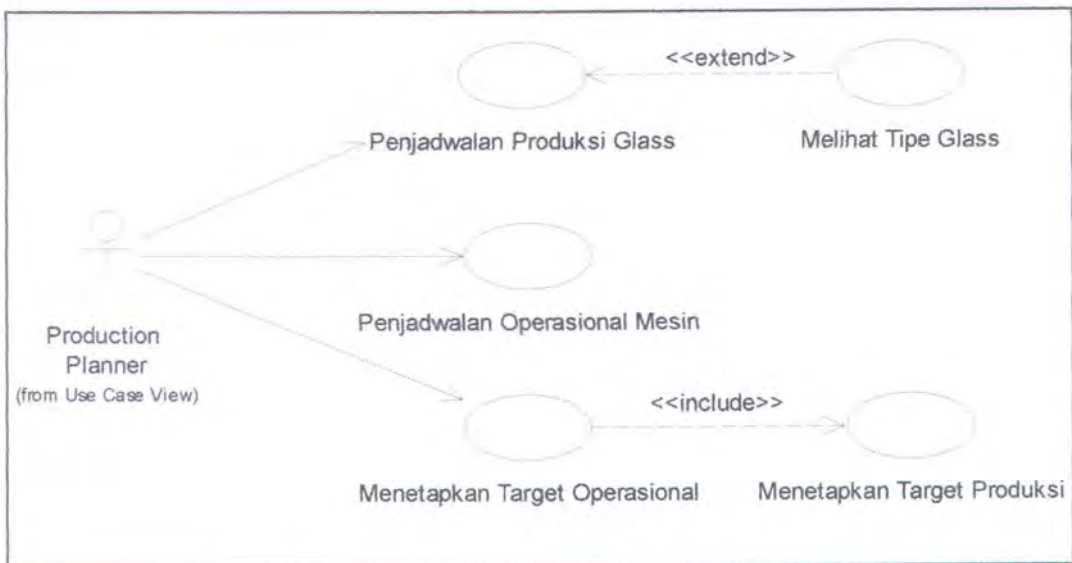
3.2.2.2.4 Production

Untuk *use case actor production* terdiri atas tiga aktivitas utama, yaitu adalah sebagai berikut :

a. Penjadwalan Produksi *Glass*

Untuk *use case* penjadwalan produksi *glass* ini terdapat *extend use case* melihat tipe *glass*, jadi sebelum melakukan penjadwalan produksi *glass*, *actor production* melihat daftar *glass* yang telah ditentukan oleh pihak *management*.

Kemudian *actor production* baru dapat melakukan penjadwalan produksi *glass*, menambah penjadwalan produksi *glass*, mengubah penjadwalan produksi *glass* dan menghapus penjadwalan produksi *glass*.



Gambar 3-6 : Use Case Diagram untuk Actor Production

b. Penjadwalan Operasional Mesin

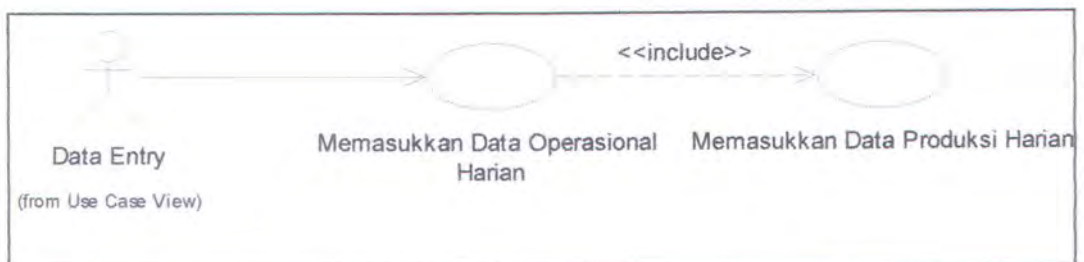
Pada *use case* penjadwalan operasional mesin, *actor production* bertugas untuk melakukan penjadwalan produksi, termasuk didalamnya adalah memasukkan data jadwal produksi baru, mengubah jadwal produksi yang sudah ada dan menghapus jadwal produksi.

c. Menetapkan Target

Untuk *use case* menetapkan target, *actor production* mempunyai tugas untuk menambahkan target produksi baru, mengubah data produksi yang sudah ada dan menghapus data target produksi. Selain itu *actor production* juga melakukan pengisian data target produksi untuk tiap tipe produksi, meliputi menambah data target baru, mengubah data target yang sudah ada dan menghapus data.

3.2.2.2.5 Data Entry

Pada *use case data entry* hanya mempunyai satu aktivitas, yaitu memasukkan data operasional harian. *Actor Data Entry* setiap hari harus memasukkan data ini, selain itu dapat mengubah data yang sudah dimasukkan dan menghapus data. Setelah memasukkan data produksi *Actor Data Entry* juga harus mengisi data produksi untuk tiap tipe glass yang diproduksi pada hari itu. Pada *use case* ini *actor Data Entry* juga mempunyai tugas untuk mengisi data baru, mengubah data dan menghapus data.



Gambar 3-7 : Use CaseDiagram untuk Actor Data Entry

3.2.2.3 Activity Diagram

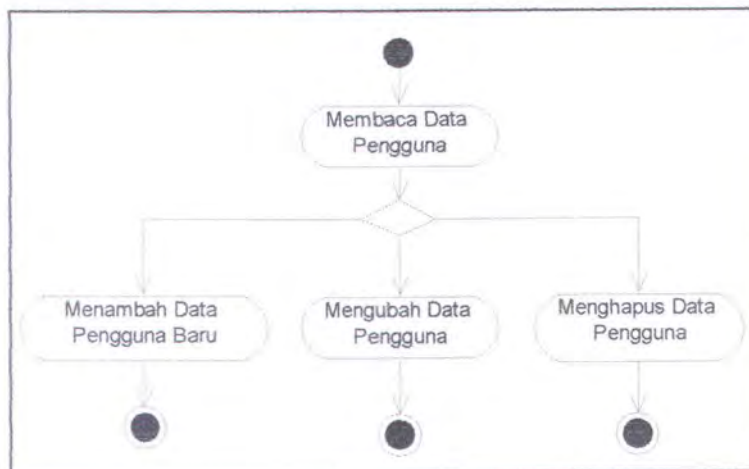
Activity diagram merepresentasikan dinamika operasional sebuah sistem. Diagram seperti ini merupakan *flow chart* yang digunakan untuk menunjukkan aliran kerja sebuah sistem yang mana mereka menunjukkan aliran control dari aktifitas dalam sebuah sistem [QT02].

3.2.2.3.1 Administrator

a. Activity Diagram Maintenance Data Pengguna

Activity diagram untuk *maintenance* data pengguna yang dilakukan oleh *actor administrator* adalah menambah data pengguna baru, mengubah data pengguna dan menghapus data pengguna.

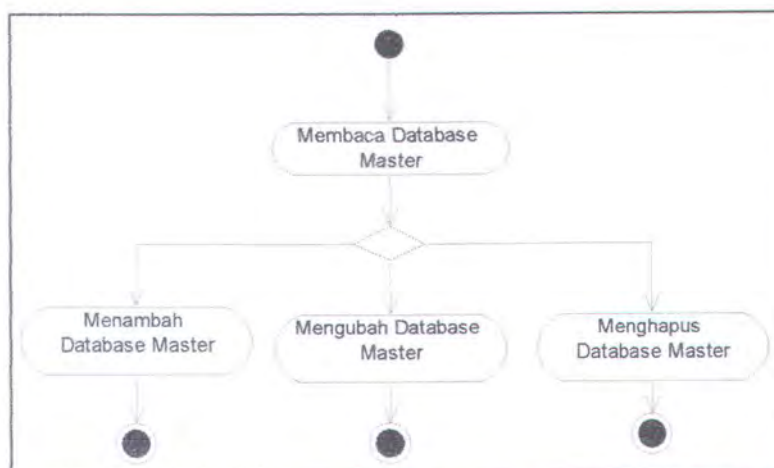
Berdasarkan *activity diagram* diatas maka bisa ditentukan form-form yang nanti akan dibuat dalam aplikasi. *form-form* tersebut adalah *form entry*, *form update* dan *form delete*.



Gambar 3-8 : Activity Diagram untuk maintenance data pengguna

b. *Activity Diagram Maintenance database Master*

Diagram berikutnya adalah *activity diagram maintenance database master*. Diagram tersebut menunjukkan aktivitas yang terjadi dalam *maintenance database master*.



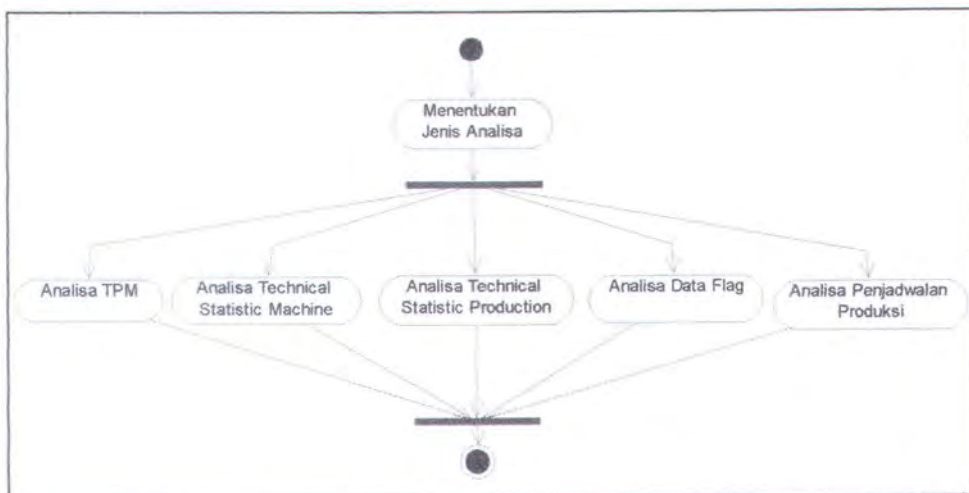
Gambar 3-9 : Activity Diagram untuk maintenance database master

Aktivitasnya adalah menambah database master, mengubah database master dan menghapus database master. Dari aktivitas diatas dapat diperoleh *form-form* yang akan dibuat dalam aplikasi. *Form-form* tersebut adalah *form entry*, *form edit* dan *form delete*.

3.2.2.3.2 Management

a. Activity Diagram Melihat Hasil Analisa

Activity diagram untuk melihat hasil analisa menunjukkan aktivitas yang dilakukan oleh *actor management*. Aktivitasnya adalah melihat hasil analisa TPM, *Technical Statistic Machine*, *Techninacal Statistic Production*, *Data Flag* dan *Production Schedule*.

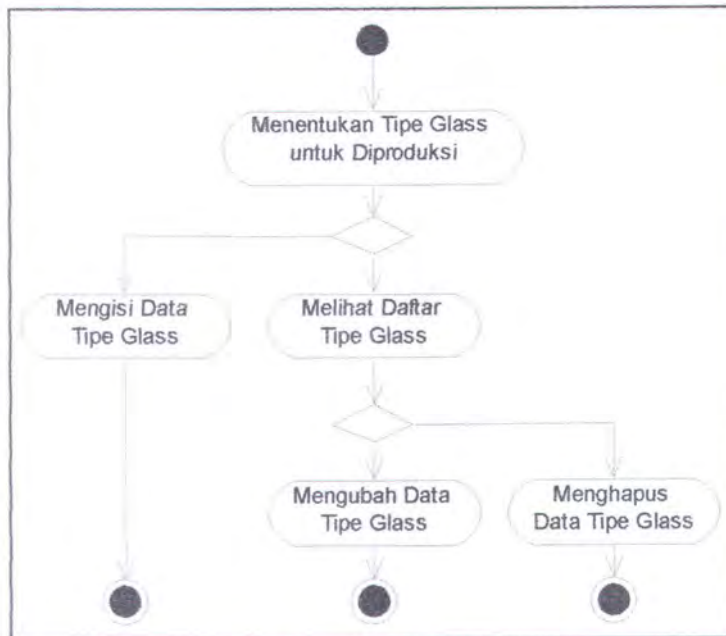


Gambar 3-10 : Activity Diagram untuk melihat hasil analisa

Dari *activity diagram* tersebut dapat diperoleh *form-form* yang akan dibuat, yaitu analisa TPM, analisa *technical statistic machine*, analisa *technical statistic production*, analisa *data flag* dan analisa penjawalan produksi.

b. *Activity Diagram Menentukan Tipe Glass*

Activity diagram untuk menentukan tipe *glass* untuk diproduksi adalah mengisi data tipe *glass*, mengubah data tipe *glass*, menghapus data dan melihat daftar tipe *glass*.



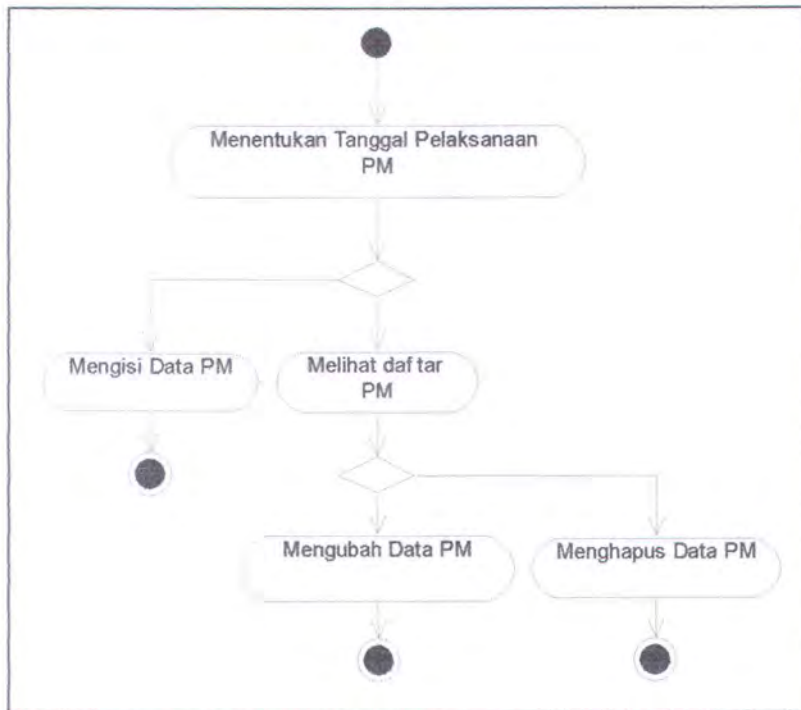
Gambar 3-11 : *Activity Diagram* untuk menentukan tipe *glass*

Dari *activity diagram* diatas diperoleh *form-form* yang akan dibuat, yaitu *form* mengisi data tipe *glass*, mengubah data tipe *glass*, menghapus data tipe *glass* dan melihat daftar tipe *glass*.

3.2.2.3.3 Maintenance

a. *Activity Diagram* Penjadwalan PM

Activity diagram penjadwalan PM menunjukkan aktivitas yang dilakukan oleh *aktor maintenance*. Aktivasnya meliputi menentukan tanggal pelaksanaan PM, mengisi data PM, mengubah data PM dan menghapus data PM.

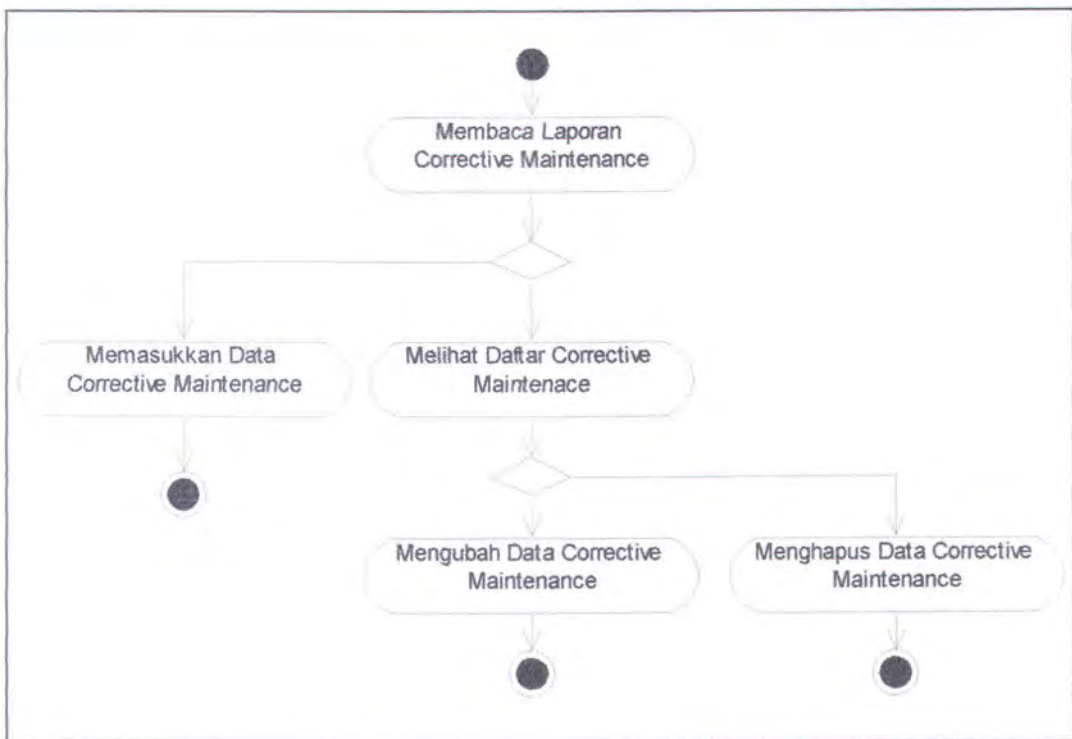


Gambar 3-12 : Activity Diagram untuk penjadwalan PM

Dari aktivitas diatas dapat diperoleh *form-form* yang akan dibuat yaitu *form* menentukan tanggal pelaksanaan PM, mengisi data PM, melihat daftar PM, mengubah data PM dan menghapus data PM.

b. *Activity Diagram* Mengisi Data CM

Activity diagram mengisi data CM menunjukkan aktivitas yang dilakukan oleh aktor *maintenance*. Aktivitas meliputi membaca laporan hasil pelaksanaan CM, memasukkan data CM, mengubah data CM yang sudah ada, menghapus data CM dan melihat daftar CM. Dari aktifitas dapat diperoleh *form-form* yang akan dibuat yaitu *form* memasukkan data CM, mengubah data CM, menghapus data CM dan melihat data CM.

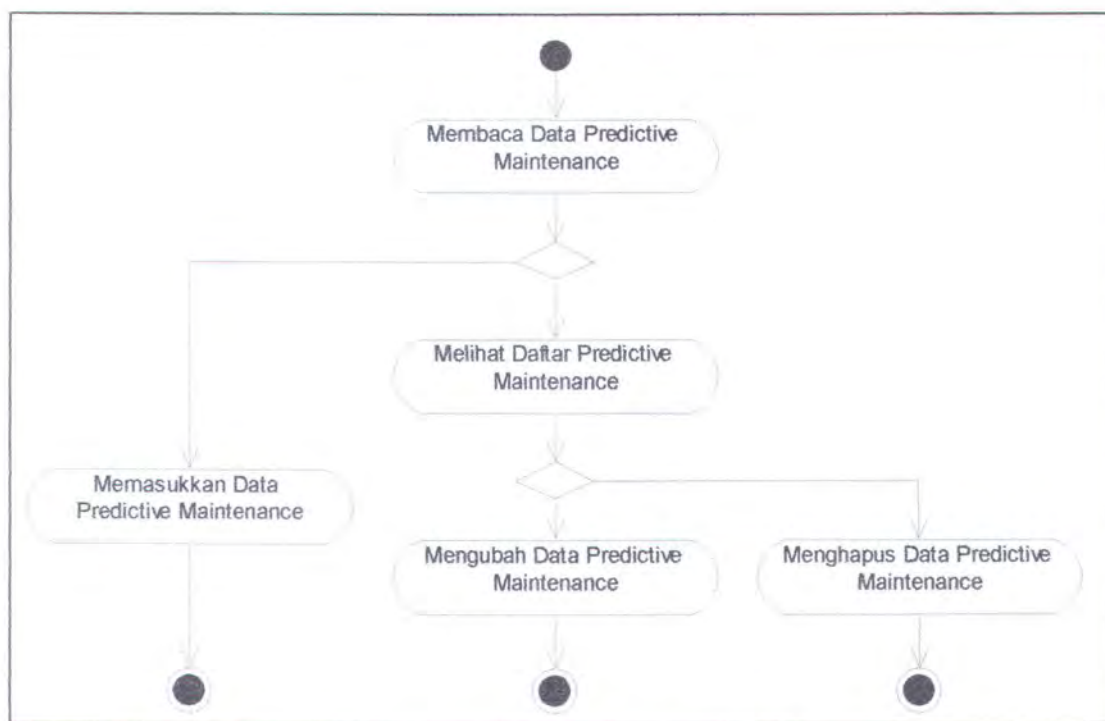


Gambar 3-13 : *Activity Diagram* untuk mengisi data CM

c. *Activity Diagram* Mengisi Data *Predictive Maintenance*

Activity diagram mengisi data PDM menunjukkan aktivitas yang dilakukan oleh *actor maintenance*. Aktivitasnya adalah membaca hasil laporan PDM, menambahkan data PDM, mengubah data PDM, menghapus data PDM dan melihat data PDM. Pada dasarnya untuk mengisi data PDM ini hampir sama dengan mengisi data pada CM yang membedakan adalah jenisnya saja. Untuk PDM dilakukan atas pembacaan parameter tertentu sedangkan CM lebih ditekankan pada *breakdown*.

Dari aktivitas diatas dapat diperoleh *form-form* yang akan dibuat, yaitu *form* menambah data PDM, mengubah data PDM, menghapus data PDM dan melihat data PDM.



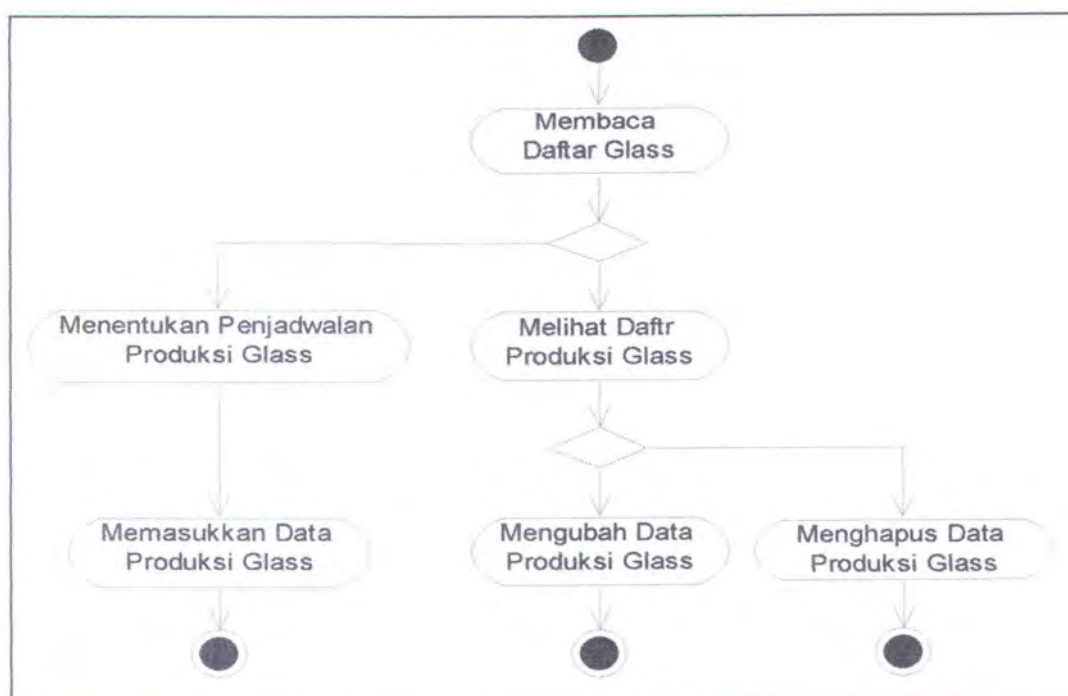
Gambar 3-14 : *Activity Diagram* untuk mengisi data PDM

3.2.2.3.4 Production

a. *Activity Diagram* Penjadwalan Produksi *Glass*

Activity diagram penjadwalan produksi *glass* menunjukkan aktivitas yang dilakukan oleh aktor production. Aktivitas yang dilakukan adalah membaca daftar *glass* yang telah dibuat oleh pihak *management*, melakukan penjadwalan produksi *glass*, mengisi data produksi *glass*, mengubah data produksi *glass*, menghapus data produksi *glass* dan melihat daftar produksi *glass*.

Dari aktivitas tersebut dapat diperoleh *form-form* yang akan dibuat, yaitu *form* penjadwalan produksi *glass*, mengisi data produksi *glass*, mengubah data produksi *glass*, menghapus data produksi *glass* dan melihat daftar produksi *glass*.

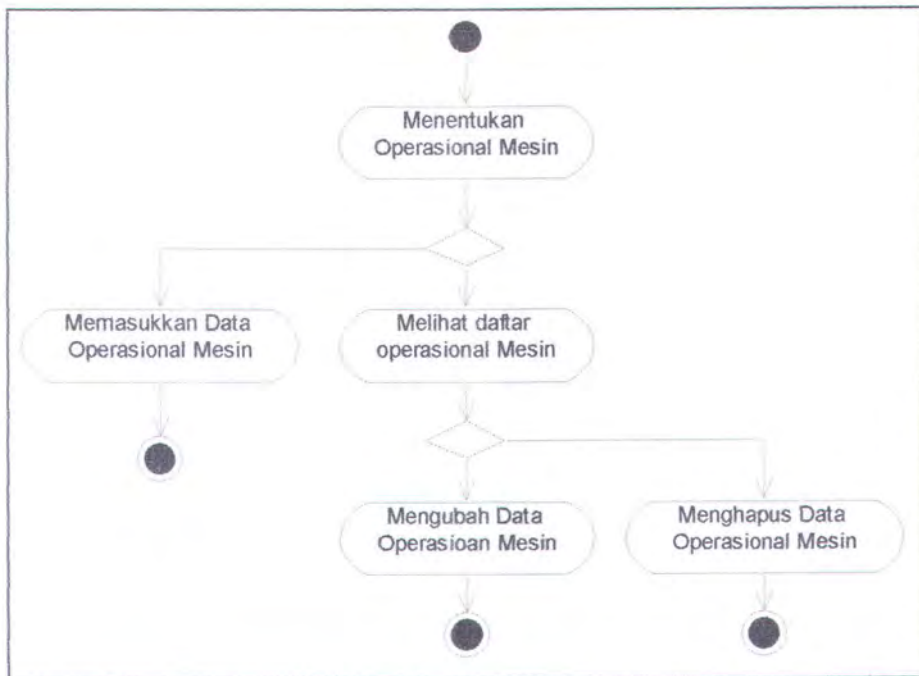


Gambar 3-15 : *Activity Diagram* penjadwalan produksi glass

b. *Activity Diagram* Penjadwalan Operasional Mesin

Activity diagram penjadwalan operasional mesin menunjukkan aktivitas yang dilakukan oleh *actor production*. Aktivitas yang dilakukan adalah menentukan operasional mesin, memasukkan data operasional mesin, mengubah data operasional mesin, menghapus data operasional mesin dan melihat daftar operasional mesin.

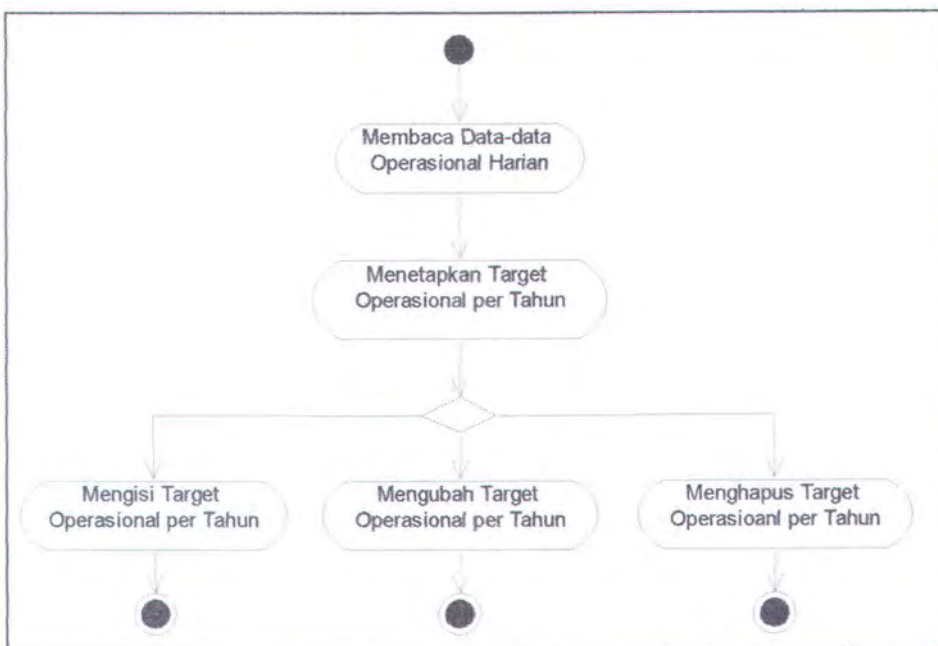
Dari aktivitas tersebut dapat diperoleh *form-form* yang akan dibuat, yaitu *form* penjadwalan operasional mesin, mengisi data operasional mesin, mengubah data operasional mesin, menghapus data operasional mesin dan melihat daftar operasional mesin.



Gambar 3-16 : *Activity Diagram* mengisi operasional mesin

c. *Activity Diagram* Menetapkan Target Operasional

Activity diagram menetapkan target operasinal menunjukkan aktivitas yang dilakukan oleh *actor production*.



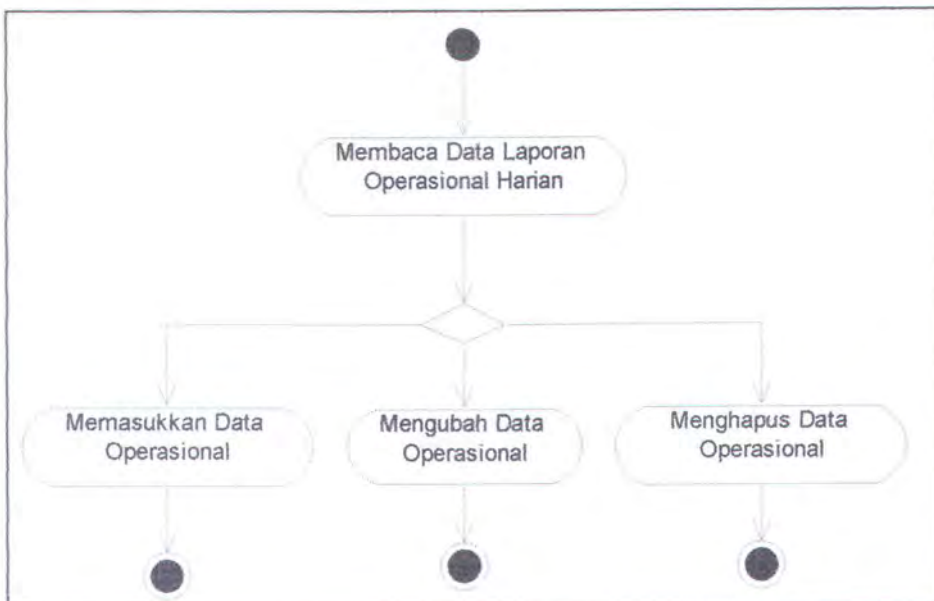
Gambar 3-17 : *Activity Diagram* menetapkan target operasional

Aktivitas yang dilakukan adalah membaca data operasional harian, menetapkan target operasional, mengisi target operasional, mengubah target operasional dan menghapus data target operasional. Dari aktivitas tersebut dapat diperoleh *form-form* yang akan dibuat, yaitu *form* menentukan target operasional, mengisi target operasional, mengubah target operasional dan menghapus data operasional.

3.2.2.3.5 Data Entry

a. Activity Diagram Memasukkan Data Operasional Harian

Pada *activity diagram* memasukkan data operasional harian yang melakukannya adalah *actor Data Entry*. Aktivitas yang dilakukan adalah membaca data laporan operasional harian, memasukkan data operasional, mengubah data operasional dan menghapus data operasional.



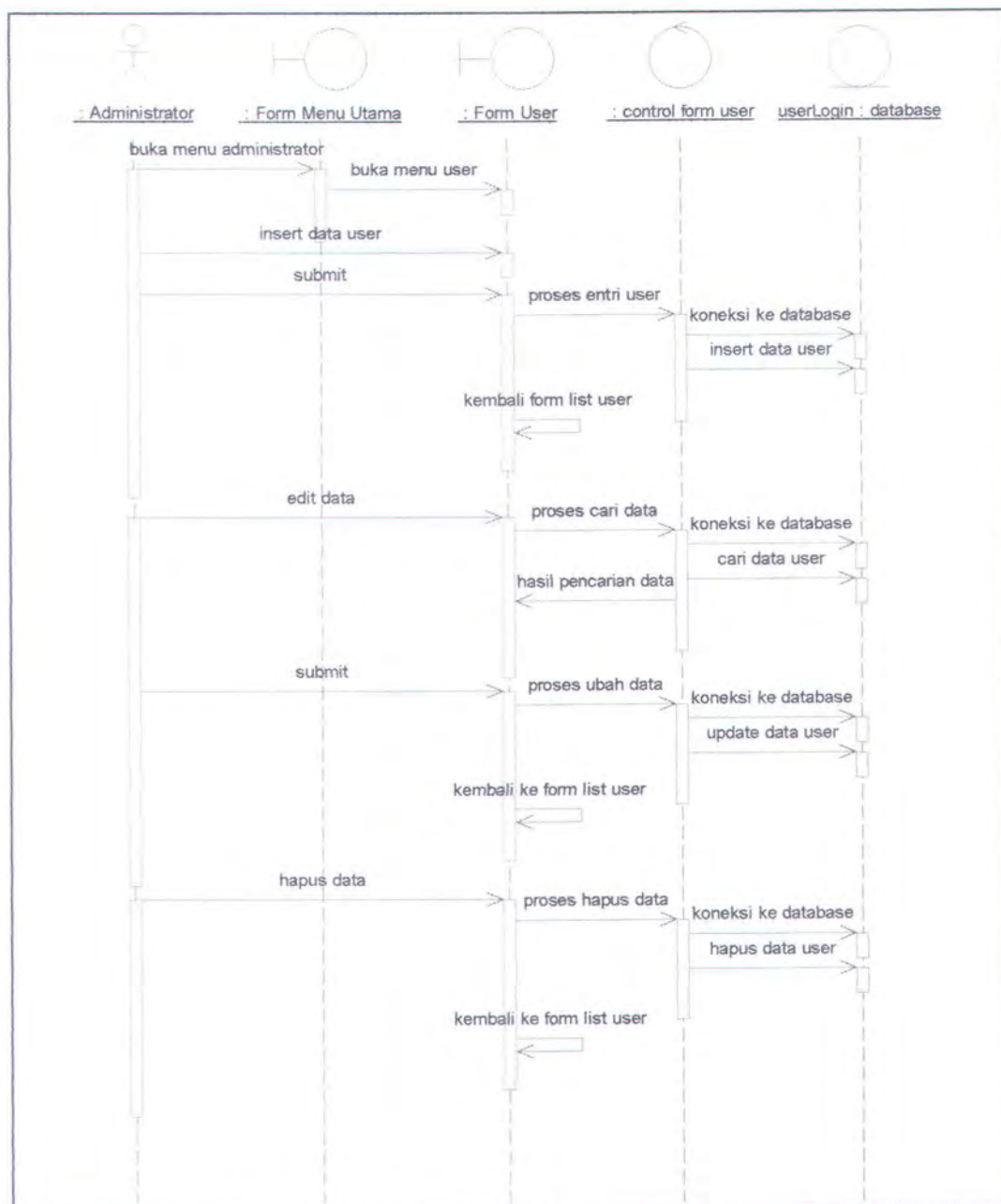
Gambar 3-18 : Activity Diagram memasukkan data operasional harian

3.2.2.4 Sequence Diagram

Pada dasarnya *sequence diagram* menunjukkan interaksi objek yang disusun dalam waktu yang berurutan [QT02]. *Sequence diagram* juga menggambarkan kumpulan objek dan *class* yang terlibat dalam sebuah scenario dan adanya pesan yang secara terurut berubah diantara objek guna menunjukkan fungsionalitas dari scenario yang ada. Dalam bagian ini akan dibahas mengenai operasional yang terjadi pada masing-masing bagian secara umum.

3.2.2.4.1 Administrator

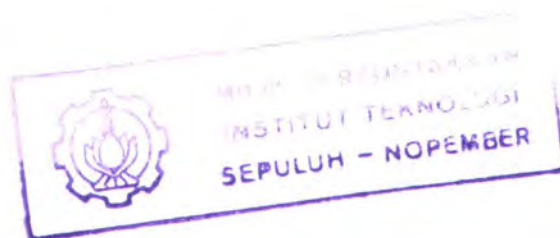
Sequence diagram untuk proses *maintenance* data user. Untuk melakukan *maintenance* data user, administrator harus melakukan login terlebih dahulu ke dalam sistem setelah melakukan login administrator baru bisa melakukan tambah data, ubah data maupun menghapus data. Untuk tambah data dari menu utama administrator dapat memilih *link* ke *add new user*. Setelah itu akan dibawa ke halaman tambah user baru. Setelah selesai melakukan pengisian data user baru kemudian menekan tombol *submit* agar data tersebut dimasukkan ke dalam database. Sedangkan untuk melakukan ubah data user, harus melalui halaman *list user* terlebih dahulu. Di situ akan ditampilkan semua user yang ada di dalam data base. Dari halaman *view* ini administrator dapat melakukan ubah data user maupun menghapus user tadi. Untuk ubah data user administrator akan dibawa ke halaman yang mirip dengan halaman *add new user* tetapi sudah ada data dari user tersebut, administrator tinggal melakukan perubahan data apabila diperlukan.

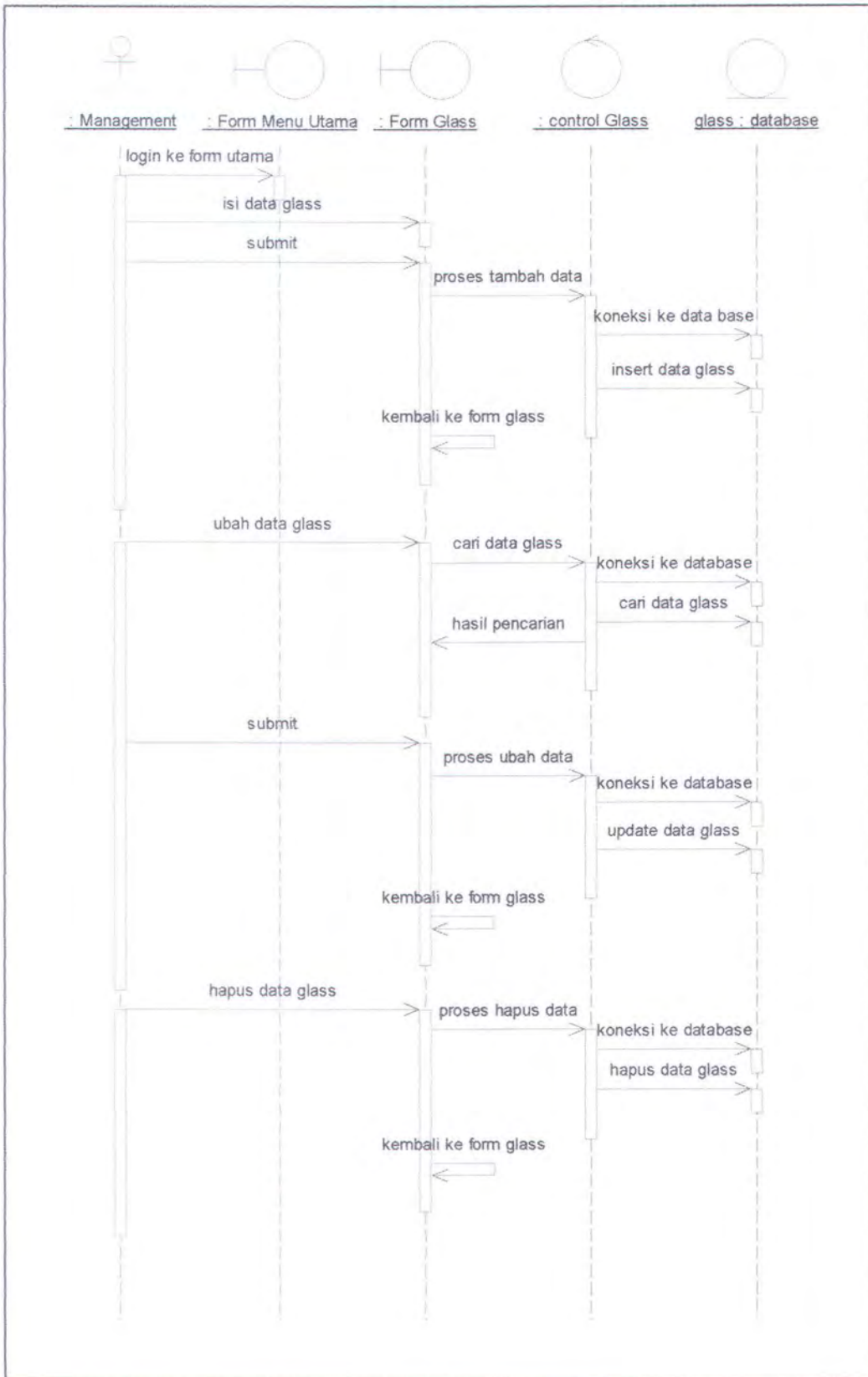


Gambar 3-19 : *Sequence Diagram* untuk *maintenance data user*

3.2.2.4.2 Management

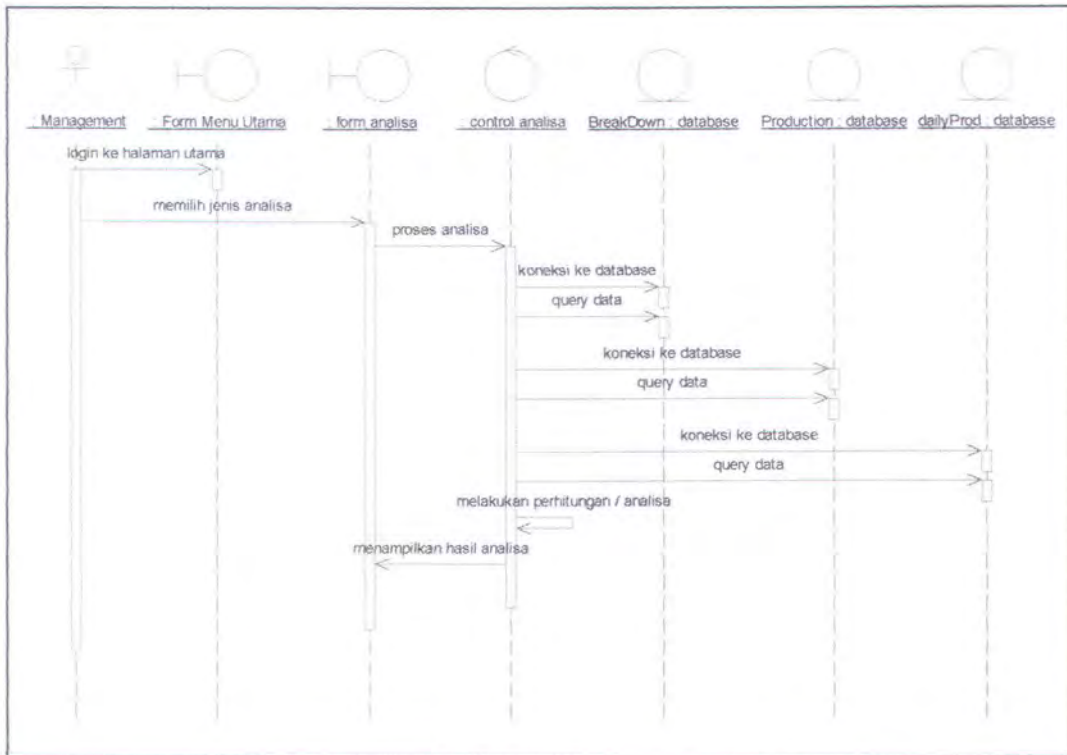
Management mempunyai dua buah *sequence diagram*, yaitu untuk menambah data *glass* dan untuk melakukan analisa perhitungan TPM. Untuk *sequence diagram* menambah data *glass* ditunjukkan seperti pada gambar berikut ini:





Gambar 3-20 : Sequence Diagram untuk maintenance data glass

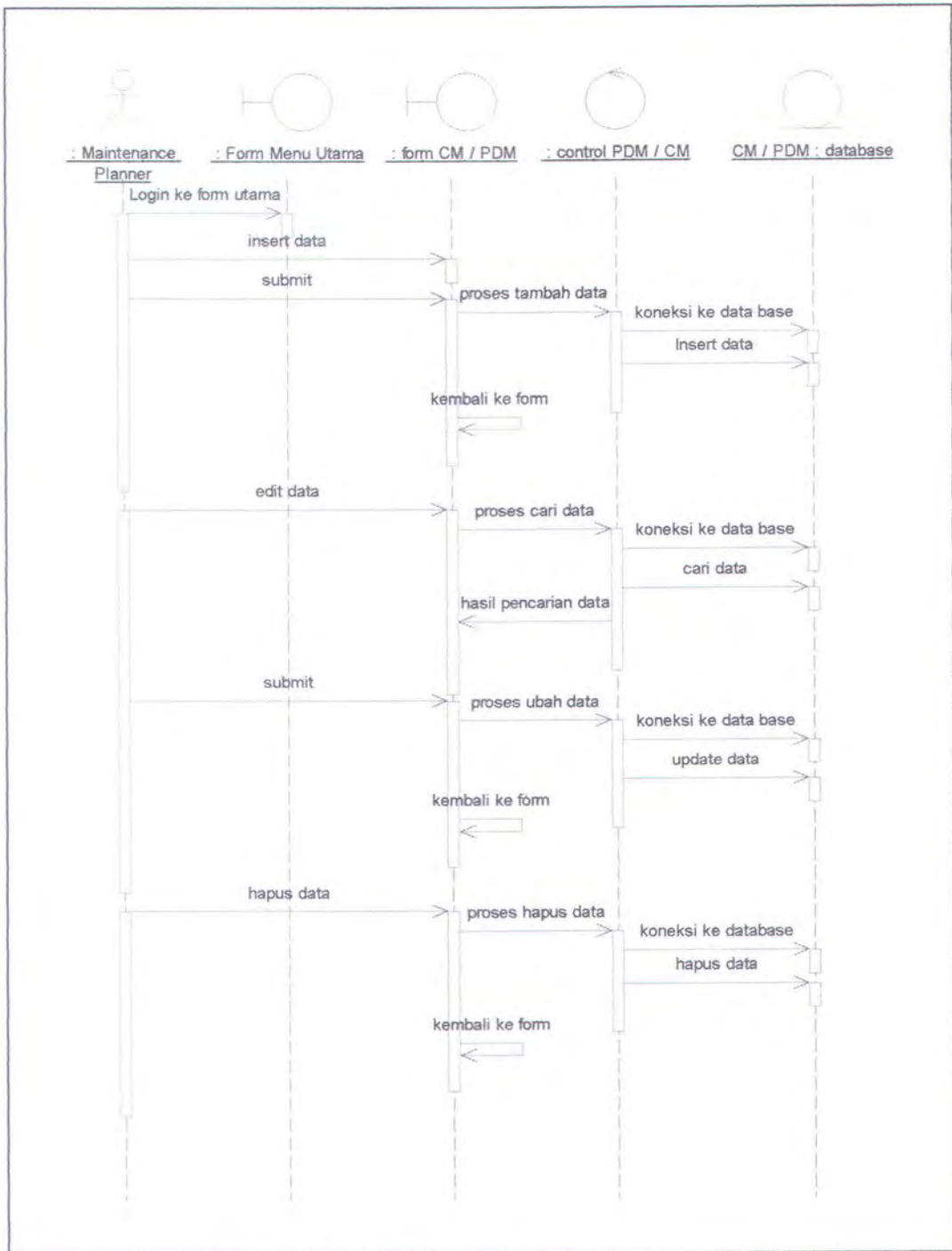
Sedangkan sequence diagram untuk melihat hasil analisa adalah sebagai berikut:



Gambar 3-21 : *Sequence Diagram* melihat hasil analisa

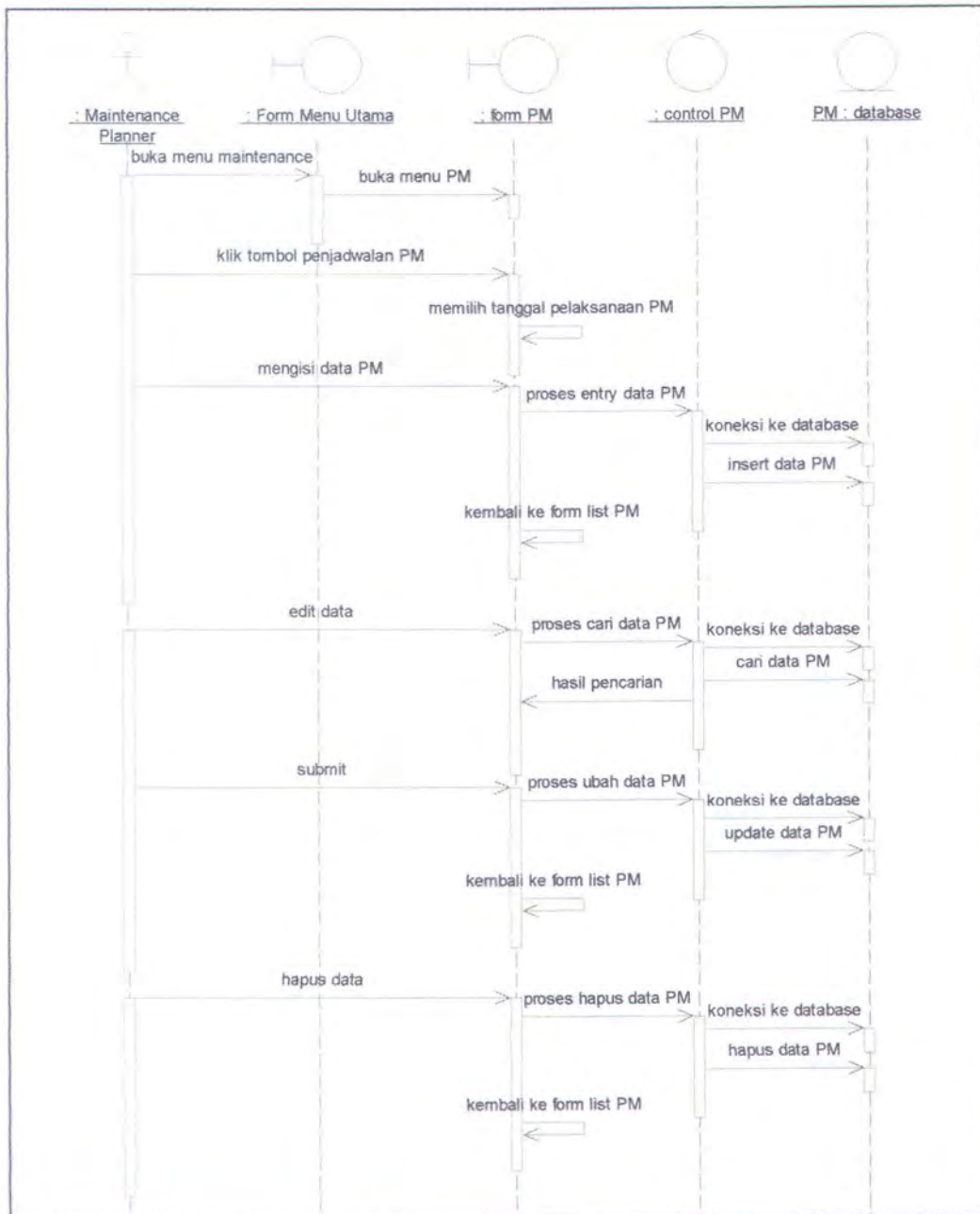
3.2.2.4.3 Maintenance

Pada *sequence diagram maintenance* untuk PM dan CM serta PDM memiliki perbedaan. Karena untuk *sequence diagram* PM diperlukan perencanaan terlebih dahulu sedangkan untuk *sequence diagram* CM dan PDM sama karena keduanya hanya melakukan pengisian data sesuai dengan tanggal dari sistem sekarang. Untuk PM setelah melakukan login dari menu utama *maintenance planer* akan dibawa ke halaman penanggalan, disitu *maintenance planner* tinggal memilih tanggal berapa akan dilakukan proses *maintenance*.



Gambar 3-22 : Sequence Diagram untuk CM / PDM

Sedangkan *sequence diagram* untuk PM adalah dengan menambahkan proses penentuan tanggal, gambarnya adalah sebagai berikut :



Gambar 3-23 : *Sequence Diagram* untuk PM

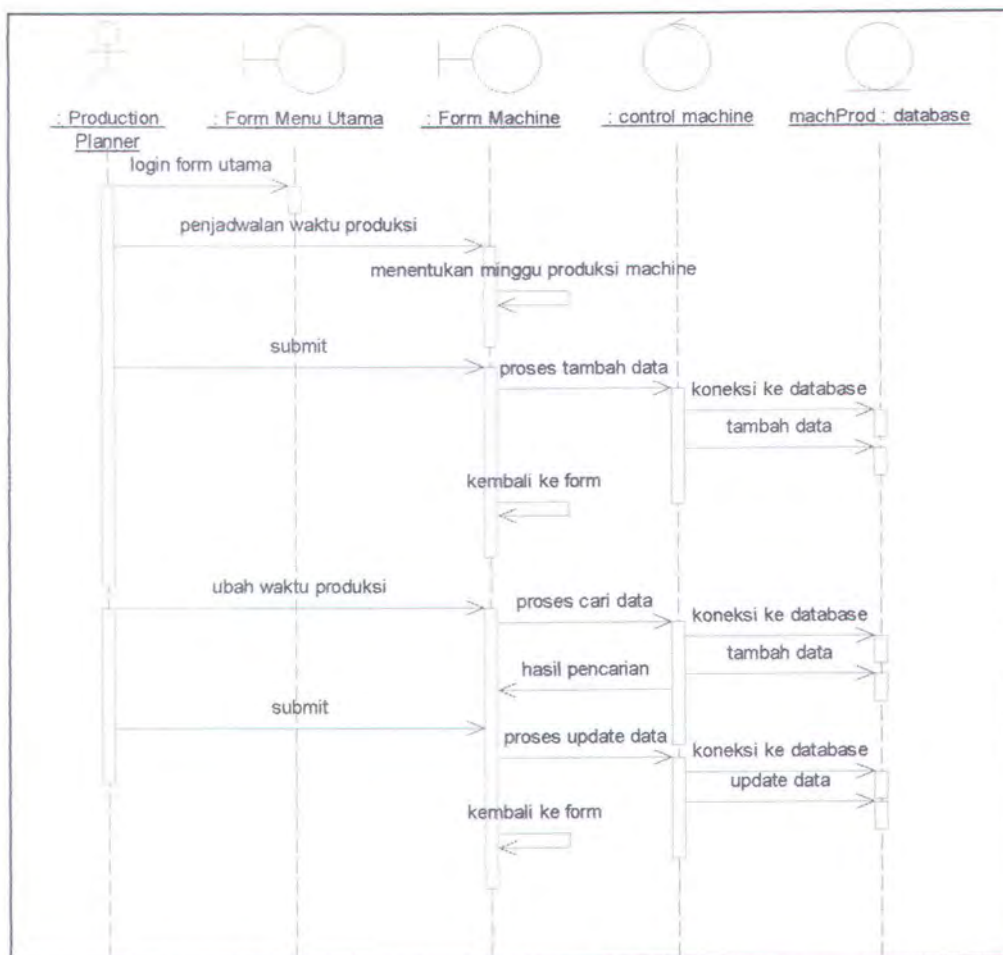
3.2.2.4.4 Production

Untuk produksi mempunyai *sequence diagram* untuk penjadwalan produksi dan menentukan target. *Sequece diagram* untuk untuk penjadwalan produksi *glass* dan mesin sama karena keduanya sama-sama dijadwalkan untuk melakukan produksi selama satu minggu bedanya untuk penjadwalan produksi *glass* yang

dijadwalkan adalah jumlah produksi *glass*-nya sedangkan untuk mesin menjadwalkan waktunya. Sedangkan *sequence diagram* untuk menentukan target produksi hampir sama dengan *sequence diagram* yang lain karena prosesnya hanya menambah, mengubah dan menghapus data yang ada.



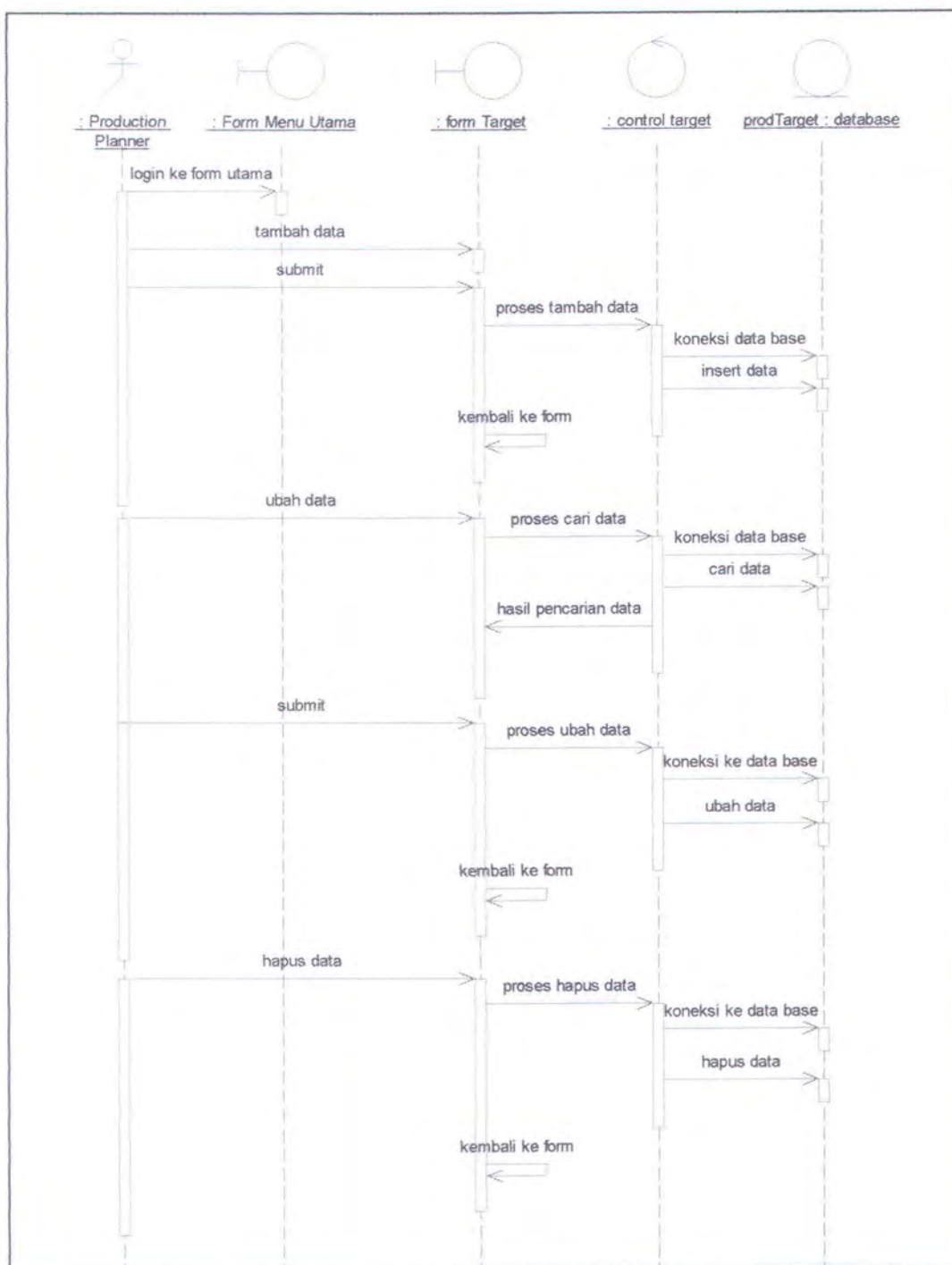
Gambar 3-24 : *Sequence Diagram* untuk penjadwalan produksi *glass*



Gambar 3-25 : *Sequence Diagram* untuk mengisi operasional mesin

Gambar diatas menunjukkan proses untuk mengisi data operasional mesin. Dalam mengisi data operasional mesin secara mingguan akan memasukkan jumlah jam kerja dari mesin beserta kemungkinan adanya pengecualian seperti hari libur nasional, cuti bersama dan yang lainnya.

Sedangkan untuk proses mengisi target produksi ditunjukkan dalam sequence digram berikut ini. Target produksi ini menggunakan data pada tahun sebelumnya untuk digunakan sebagai acuan untuk target pada tahun berikutnya.

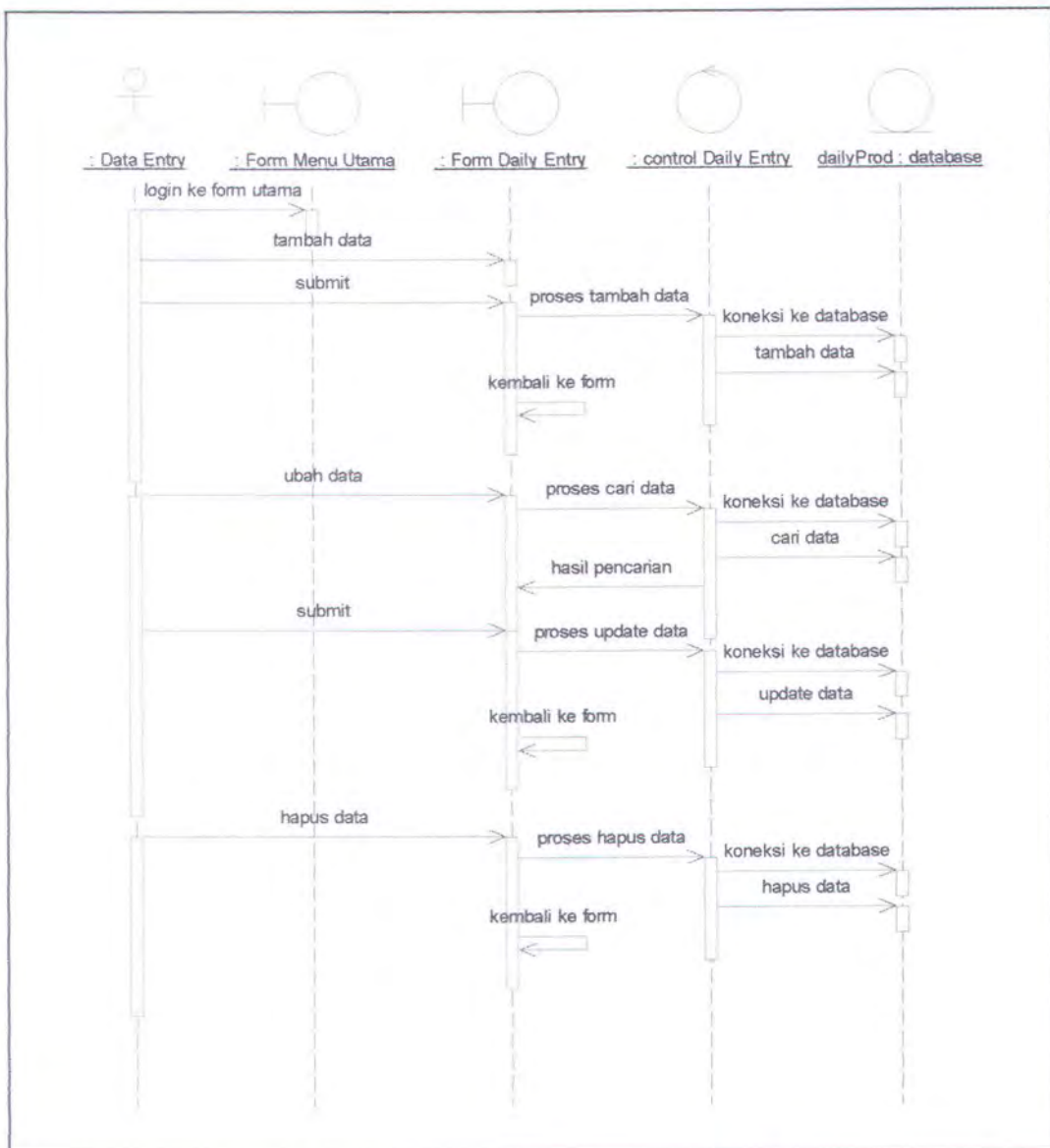


Gambar 3-26 : *Sequence Diagram* untuk menetapkan target operasional

3.2.2.4.5 Data Entry

Sequence diagram untuk data entry sama dengan *sequence diagram* sebelumnya.

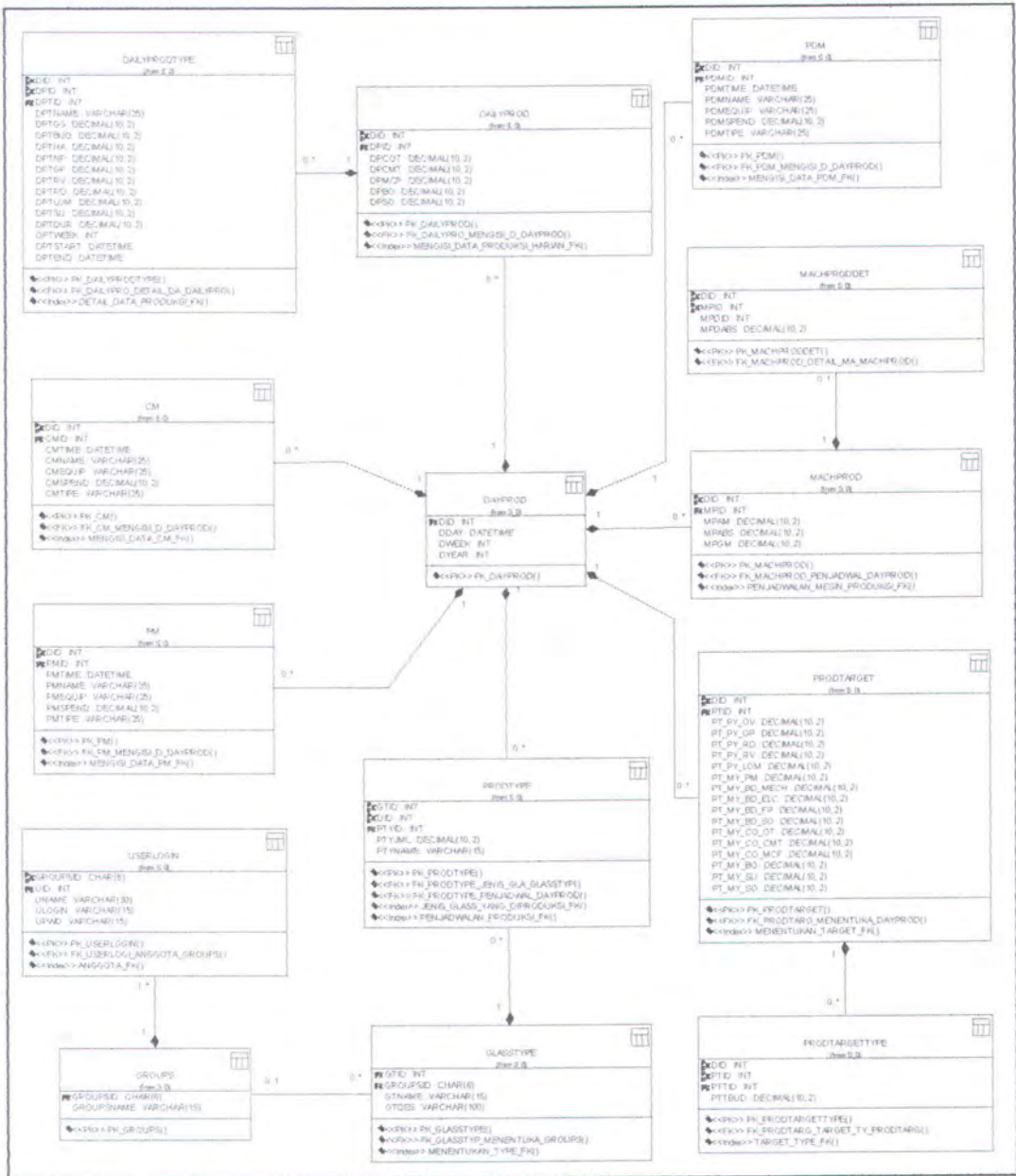
Pada *sequence diagram* data akan melakukan proses pengisian data produksi setiap hari.



Gambar 3-27 : *Sequence Diagram* data produksi harian

3.2.3 Desain Data Base

Dalam sub bab ini akan dijelaskan mengenai perancangan dan pembuatan database yang melibatkan entitas-entitas yang ada dan juga relasi yang terjadi antara entitas-entitas tersebut. Hal ini akan ditunjukkan oleh gambar berikut ini :



Gambar 3-28 : Desain database aplikasi

Pada dasarnya ada beberapa kelompok tabel yang terlibat di dalam perancangan sistem, yaitu : tabel *User*, tabel *Maintenance*, tabel *Produksi*, tabel *Data Harian*, tabel *Target* dan tabel *Jenis Glass*. Masing-masing kelompok tabel tersebut merepresentasikan operasional yang terjadi secara umum pada masing-masing kelompok.

3.2.3.1.1 Kelompok Tabel User

Tabel yang masuk kedalam kelompok tabel *user* adalah tabel UserLogin dan tabel Groups.

Tabel 3-1 : Tabel groups

No	Nama Item Data	Tipe Data	Keterangan	Asal Tabel
1	groupsID	Varchar(6)	Primary Key	
2	groupsName	Varchar(15)		

Tabel Groups digunakan untuk menyimpan data dari groups yang akan dibuat. Masing-masing groups nanti akan memiliki hak dan tugas sendiri-sendiri.

Fields yang digunakan adalah :

- groupsID : menyimpan no ID groups.
- groupsName : menyimpan nama dari groups tersebut

Tabel 3-2 : Tabel userLogin

No	Nama Item Data	Tipe Data	Keterangan	Asal Tabel
1	uID	Varchar(6)	Primary Key	
2	groupsID	Varchar(6)	Foreign Key	Groups
3	uName	Varchar(25)		
4	uLogin	Varchar(10)		
5	uPwd	Varchar(10)		

Tabel User Login digunakan untuk untuk menyimpan daftar user yang mempunyai hak untuk masuk kedalam sistem. Fields yang digunakan adalah :

- uID : menyimpan no induk karyawan.
- groupsID : menyimpan jenis group dari karyawan.
- uName : menyimpan nama dari karyawan.
- uLogin : menyimpan nama login karyawan.



- uPwd : menyimpan password login karyawan.

3.2.3.1.2 *Kelompok Tabel Tanggal*

Tabel yang masuk kedalam kelompok ini adalah tabel dayProd.

Tabel 3-3 : Tabel dayProd

No	Nama Item Data	Tipe Data	Keterangan	Asal Tabel
1	dID	Int	Primary Key	
2	dDay	DateTime		
3	dWeek	Int		
4	dYear	Int		

Tabel dayProd dipergunakan untuk menyimpan data mengenai tanggal dan minggu, data ini akan digunakan untuk keterangan waktu pada data produksi dan maintenance.

- dID : menyimpan no ID untuk dayProd.
- dDay : menyimpan tanggal.
- dWeek : menyimpan minggu ke berapa dalam tahun tersebut.
- dYear : menyimpan tahun.

3.2.3.1.3 *Kelompok Tabel Maintenance*

Tabel yang masuk kedalam kelompok ini adalah tabel PDM, tabel CM, tabel PM.

Tabel 3-4 : Tabel PDM

No	Nama Item Data	Tipe Data	Keterangan	Asal Tabel
1	pdmID	Int	Primary Key	
2	dID	Int	Foreign Key	dayProd
3	pdmTime	Datetime		
4	pdmName	Varchar(25)		
5	pdmEquipt	Varchar(25)		
6	pdmSpend	Decimal(6,2)		
7	pdmTipe	Varchar(6)		

Tabel PDM digunakan untuk menyimpan data dari proses *predictive maintenance* yang terjadi pada hari itu. *Fields* yang dipergunakan :

- pdmID : menyimpan no ID dari pelaksanaan PDM.
- dID : menyimpan day ID yang diperoleh dari tabel dayProd.
- pdmTime : menyimpan jam pelaksanaan PDM.
- pdmName : menyimpan nama dari PDM.
- pdmEquipt : menyimpan nama *equipment* yang diperbaiki.
- pdmSpend : menyimpan durasi waktu yang diperlukan untuk pelaksanaan PDM
- pdmTipe : menyimpan jenis PDM yang dilakukan.

Tabel PM digunakan untuk menyimpan data preventive maintenance yang dilakukan pada hari tertentu (telah dijadwalkan). *Fields* yang dipergunakan :

Tabel 3-5 : Tabel PM

No	Nama Item Data	Tipe Data	Keterangan	Asal Tabel
1	pmID	Int	Primary Key	dayProd
2	dID	Int	Foreign Key	
3	pmTime	Datetime		
4	pmName	Varchar(25)		
5	pmEquipt	Varchar(25)		
6	pmSpend	Decimal(6,2)		
7	pmTipe	Varchar(6)		

Tabel PMdet digunakan untuk menyimpan detail dari proses maintenance yang dilakukan pada hari tertentu. *Fields* yang dipergunakan :

- pmID : menyimpan no ID dari pelaksanaan PM.

- dID : menyimpan day ID yang diperoleh dari tabel dayProd.
- pmTime : menyimpan jam pelaksanaan PM
- pmName : menyimpan nama dari PM.
- pmEquipt : menyimpan nama *equipment* yang diperbaiki.
- pmSpend : menyimpan durasi waktu yang diperlukan untuk pelaksanaan PM.
- pmdTipe : menyimpan jenis PM yang dilakukan.

Tabel 3-6 : Tabel CM

No	Nama Item Data	Tipe Data	Keterangan	Asal Tabel
1	cmID	Int	Primary Key	dayProd
2	dID	Int	Foreign Key	
3	cmTime	Datetime		
4	cmName	Varchar(25)		
5	cmEquipt	Varchar(25)		
6	cmSpend	Decimal(6,2)		
7	cmTipe	Varchar(6)		

Tabel CM digunakan untuk menyimpan detail dari proses *corrective maintenance* yang dilakukan pada hari itu. Fields yang dipergunakan adalah :

- cmID : menyimpan no ID dari pelaksanaan CM.
- dID : menyimpan day ID yang diperoleh dari tabel dayProd.
- cmTime : menyimpan jam pelaksanaan CM
- cmName : menyimpan nama dari CM.
- cmEquipt : menyimpan nama *equipment* yang diperbaiki.
- cmSpend : menyimpan durasi waktu yang diperlukan untuk pelaksanaan CM.
- cmTipe : menyimpan jenis CM yang dilakukan.

3.2.3.1.4 Kelompok Tabel Produksi

Tabel yang masuk kedalam kelompok ini adalah tabel prodType.

Tabel 3-7 : Tabel prodType

No	Nama Item Data	Type Data	Keterangan	Asal Tabel
1	ptyID	Int	Primary Key	
2	dID	Int	Foreign Key	dayProd
3	gtID	Int	Foreign Key	glassType
4	ptyName	Varchar(6)		
5	ptyJumlah	Decimal(6,2)		

Tabel prodType digunakan untuk menyimpan detail dari produksi yang telah dijadwalkan terlebih dahulu, disini juga menyimpan data jenis glass yang akan diproduksi dan jumlah produksinya. Fields yang dipergunakan adalah :

- ptyID : menyimpan no ID dari detail tipe produksi.
- dID : menyimpan day ID dari tabel dayProd.
- gtID : menyimpan glass ID dari tabel glassType.
- ptyName : menyimpan jenis glass yang diproduksi.
- ptyJumlah: menyimpan jumlah glass yang diproduksi.

3.2.3.1.5 Kelompok Tabel Data Harian

Tabel 3-8 : Tabel dailyProd

No	Nama Item Data	Type Data	Keterangan	Asal Tabel
1	DPID	Int	Primary Key	dayProd
2	DID	Int	Foreign Key	
8	DPCOT	Decimal(6,2)		
9	DPCMT	Decimal(6,2)		
10	DPMCF	Decimal(6,2)		
11	DPBO	Decimal(6,2)		
12	DPSU	Decimal(6,2)		
13	DPSO	Decimal(6,2)		

Tabel yang masuk dalam kelompok data harian adalah `dailyProd` dan `dailyProdType`.

Tabel `dailyProd` digunakan untuk menyimpan data harian produksi. Fields yang dipergunakan adalah :

- `dpID` : menyimpan no ID dari data harian.
- `dID` : menyimpan day ID yang diperoleh dari tabel `dayProd`.
- `dpCOT` : menyimpan data change over overtime.
- `dpCMT` : menyimpan data change mould time.
- `dpMCF` : menyimpan data mould change frequency.
- `dpBo` : menyimpan data Black Out.
- `dpSo` : menyimpan data Stock Out.

Tabel 3-9 : Tabel `dailyProdType`

No	Nama Item Data	Tipe Data	Keterangan	Asal Tabel
1	DPTID	Int	Primary Key	
2	DPID	Int	Foreign Key	<code>dailyProd</code>
3	DID	Int	Foreign Key	<code>dayProd</code>
4	DPTNAME	Varchar(10)		
5	DPTGP	Decimal(6,2)		
6	DPTRD	Decimal(6,2)		
7	DPTRV	Decimal(6,2)		
8	DPTLOM	Decimal(6,2)		
9	DPTGS	Decimal(6,2)		
10	DPTBUD	Decimal(6,2)		
11	DPTHA	Decimal(6,2)		
12	DPTNP	Decimal(6,2)		
13	DPTSU	Decimal(6,2)		
14	DPTDUR	Decime(6,2)		
15	DPTSTART	DATETIME		
16	DPTEND	DATETIME		

Tabel *dailyProdType* digunakan untuk menyimpan detail dari tiap tipe *glass* yang diproduksi pada hari itu. Fields yang dipergunakan adalah :

- *dptID* : menyimpan no ID dari detail data harian untuk tiap tipe *glass*.
- *dpID* : menyimpan *day production* ID dari tabel *dailyProd*.
- *dID* : menyimpan day ID dari tabel *dayProd*.
- *dptName* : menyimpan nama tipe *glass*.
- *dptGP* : menyimpan data *glass problem*.
- *dptRV* : menyimpan data *reject visual*.
- *dptRD* : menyimpan data *reject dimensioan*.
- *dptLom* : menyimpan data *loss on machine*.
- *dptGS* : menyimpan data *gross speed*.
- *dptBud* : menyimpan data *budget*.
- *dptHA* : menyimpan data *hours available*.
- *dptNP* : menyimpan *net production*.
- *dptDur* : menyimpan data lama waktu produksi.
- *dptStart* : menyimpan data waktu mulai produksi.
- *dptEnd* : menyimpan data waktu berakhirnya produksi.

Tabel 3-10 : Tabel *machProd*

No	Nama Item Data	Tipe Data	Keterangan	Asal Tabel
1	MPID	Int	Primary Key	dayProd
2	DID	Int	Foreign Key	
3	MPAM	Decimal(6,2)		
4	MPABS	Decimal(6,2)		
5	MPGM	Decimal(6,2)		

Tabel machProd digunakan untuk menyimpan data produksi dari mesin tiap minggu. Fields yang digunakan adalah :

- mpID : menyimpan no ID machine.
- dID : menyimpan day ID dari tabel dayProd.
- mpAM : menyimpan data *available machine*.
- mpAbs : menyimpan data total *absence machine*.
- mpGm : menyimpan data *gross machine*.

Tabel 3-11 : Tabel machProdDet

No	Nama Item Data	Tipe Data	Keterangan	Asal Tabel
1	MPDID	Int	Primary Key	
2	DID	Int	Foreign Key	dayProd
3	MPID	Int	Foreign Key	machProd
4	MPDABS	Decimal(10,2)		

Tabel machProdDet menyimpan data detail absent program dari tiap hari selama satu minggu penjadwalan produksi mesin. Fields yang digunakan adalah :

mpdID : menyimpan no ID dari machProdDet.

dID : menyimpan day ID dari tabel dayProd.

mpID : menyimpan machine production ID dari tabel machProd.

mpdAbs : menyimpan data dari absent program untuk hari itu.

3.2.3.1.6 Kelompok Tabel Target

Tabel yang masuk kedalam kelompok ini adalah tabel prodTarget, tabel prodTypeTarget.

Tabel ProdTarget digunakan untuk menyimpan daftar target produksi dan daftar target ini diisikan untuk tiap tahun. Fields yang dipergunakan adalah :

- ptID : menyimpan no ID dari target produksi yang berupa tahun.
- did : menyimpan day ID dari tabel dayProd.

Tabel 3-12 : Tabel prodTarget

No	Nama Item Data	Type Data	Keterangan	Asal Tabel
1	PTID	Int	Primary Key	dayProd
2	DID	Int	ForeignKey	
3	PT_PY_OV	Decimal(6,2)		
4	PT_PY_GP	Decimal(6,2)		
5	PT_PY_RD	Decimal(6,2)		
6	PT_PY_RV	Decimal(6,2)		
7	PT_PY_LOM	Decimal(6,2)		
8	PT_MY_PM	Decimal(6,2)		
9	PT_MY_BD_MECH	Decimal(6,2)		
10	PT_MY_BD_ELC	Decimal(6,2)		
11	PT_MY_BD_FP	Decimal(6,2)		
12	PT_MY_BD_SO	Decimal(6,2)		
13	PT_CO_OT	Decimal(6,2)		
14	PT_CO_CMT	Decimal(6,2)		
15	PT_CO_MCF	Decimal(6,2)		
16	PT_MY_BO	Decimal(6,2)		
17	PT_MY_SU	Decimal(6,2)		
18	PT_MY_SO	Decimal(6,2)		

- Pt_py_ov : menyimpan data target over filling.
- Pt_py_gp : menyimpan data target glass problem.
- Pt_py_rd : menyimpan data target reject dimensi.
- Pt_py_rv : menyimpan data target reject visual.
- Pt_py_lom : menyimpan data target loss on machine.
- pt_my_pm : menyimpan data target PM.
- Pt_my_bd_mech : menyimpan data breakdown mechanic.
- Pt_my_bd_elc : menyimpan data breakdown electrical.
- Pt_my_bd_fp : menyimpan data breakdown feeder pin.
- Pt_my_bd_so : menyimpan data breakdown stock out.

- Pt_co_ot : menyimpan data target change over over time.
- Pt_co_cmt : menyimpan data target change mould time.
- Pt_co_mcf : menyimpan data target mould change frequency dalam tms/
week
- Pt_my_bo : menyimpan data target black out.
- Pt_my_su : menyimpan data target start up.
- Pt_my_so : menyimpan data target stock out.

Tabel 3-13 : Tabel ProdTypeTarget

No	Nama Item Data	Tipe Data	Keterangan	Asal Tabel
1	PTTID	Int	Primary Key	
2	PTID	Int	Foreign Key	prodTarget
3	DID	Int	Foreign Key	dayProd
4	PTT_BUD	Decimal(10,2)		

Tabel prodTypeTarget digunakan untuk menyimpan data target untuk tiap tipe produksi. Fields yang dipergunakan adalah :

- pttID : menyimpan no ID dari target tiap tipe glass.
- ptID : menyimpan ptID dari tabel prodTarget.
- did : menyimpan day ID dari tabel dayProd.
- Ptt_bud : menyimpan data budget.

3.2.3.1.7 Kelompok Tabel Jenis Glass

Tabel 3-14 : Tabel GlassType

No	Nama Item Data	Tipe Data	Keterangan	Asal Tabel
1	GTID	Int	Primary Key	
2	GROUPSID	Int	Foreign Key	Groups
3	GTNAME	Varchar(15)		
4	GTDES	Varchar(100)		

Tabel yang termasuk dalam kelompok ini adalah tabel *glassType*. Tabel *glassType* ini digunakan untuk menyimpan daftar tipe glass yang akan diproduksi.

Fields yang dipergunakan :

- *gtID* : menyimpan no ID dari tipe glass.
- *groupsID* : menyimpan data groups ID dari tabel groups.
- *gtName* : menyimpan data nama tipe glass.
- *gtDes* : menyimpan data description dari tipe glass.

3.2.4 Desain Antar Muka

Pada sub bab ini akan dijelaskan mengenai perancangan antar muka dari aplikasi yang akan dibuat nanti. Disini ada beberapa menu utama yang akan digunakan untuk mengakses sesuai dengan groups dari user login tersebut dan untuk mengakses menu-menu tersebut harus login terlebih dahulu. Menu-menu utama tersebut adalah :

- Menu Administrator
- Menu *Management*
- Menu *Maintenance*
- Menu *Production*
- Menu *Data Entry*

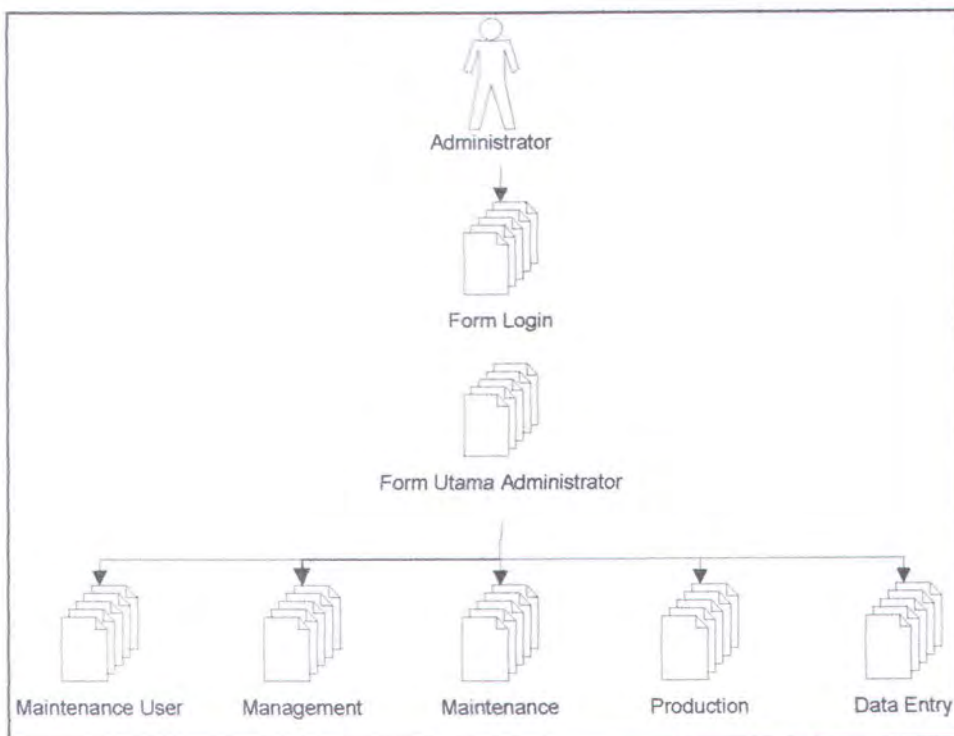


3.2.4.1.1 Menu Administrator

Dari menu administrator dapat melakukan akses ke semua menu yang ada dalam sistem ini.

Dari menu administrator terdapat lima buah sub menu yaitu :

- *Maintenance user* : meliputi menambah, mengubah, menghapus dan melihat daftar user.
- *Management* : dapat mengakses semua sub menu *management*.
- *Maintenance* : dapat mengakses semua sub menu *maintenance*.
- *Production* : dapat mengakses semua sub menu *production*.
- *Data Entry* : dapat mengakses semua sub menu *data entry*.



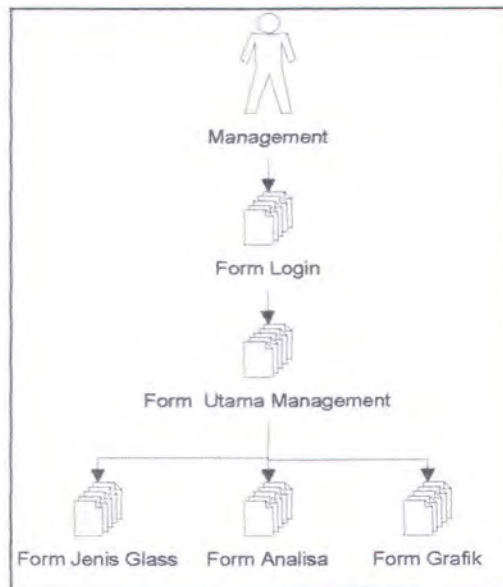
Gambar 3-29 : Diagram aliran sistem untuk administrator

3.2.4.1.2 Menu Management

Menu *management* dapat diakses oleh *user* yang mempunyai login dengan hak administrator maupun *management*.

Dari menu *management* terdapat tiga sub menu yaitu :

- Form jenis *glass* : meliputi menambah jenis *glass* baru, mengubah data jenis *glass* yang sudah ada, menghapus data *glass* dan melihat daftar *glass*.



Gambar 3-30 : Diagram aliran sistem untuk *management*

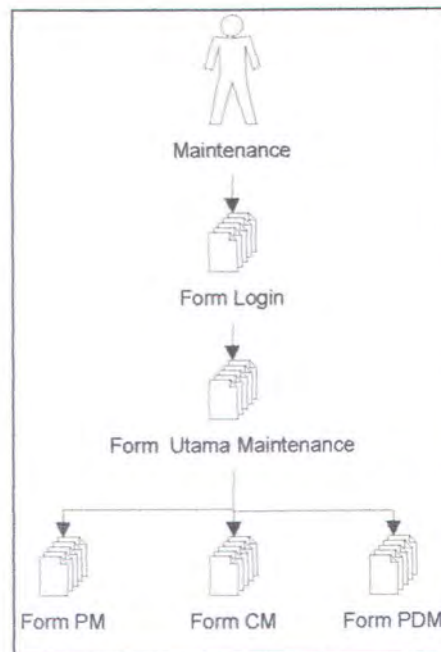
- Form analisa : meliputi melihat hasil perhitungan yang berupa *Data Flag*, *technical statistic*, *TPM*, *production performance* dan *production schedule*.
- Form grafik : menampilkan hasil perhitungan analisa kedalam bentuk grafik.

3.2.4.1.3 Menu Maintenance

Menu *maintenance* dapat diakses oleh *user* yang mempunyai hak login administrator dan *maintenance*.

Dari menu *maintenance* terdapat tiga sub menu yaitu :

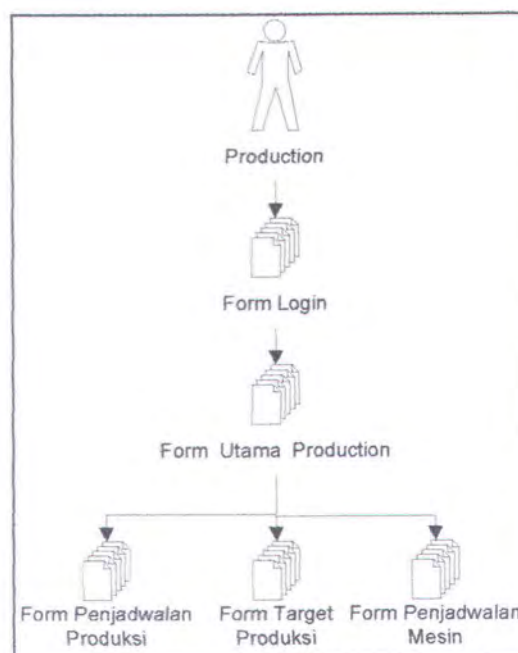
- *Form PM* : meliputi merencanakan pelaksanaan PM pada tanggal tertentu, mengubah data PM, menghapus data PM dan melihat daftar PM.
- *Form CM* : meliputi menambah data CM baru, mengubah data CM, menghapus data CM dan melihat daftar CM



Gambar 3-31 : Diagram aliran sistem untuk *maintenance*

- *Form PDM* : meliputi menambah data PDM baru, mengubah data PDM, menghapus data PDM dan melihat daftar PDM.

3.2.4.1.4 Menu Production



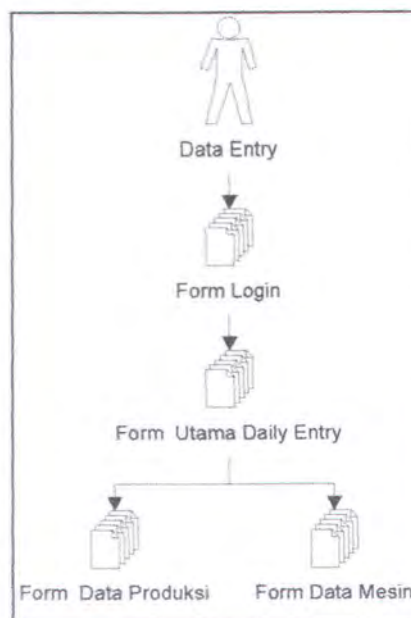
Gambar 3-32 : Diagram aliran sistem untuk *production*

Menu production dapat diakses oleh *user* yang mempunyai login administrator dan *production*.

Dari menu *production* terdapat tiga sub menu, yaitu :

- *Production* : meliputi perencanaan untuk memproduksi glass tipe tertentu pada hari tertentu, mengubah data produksi glass, menghapus data produksi *glass* dan melihat daftar seluruh produksi *glass*.
- *Target* : meliputi mengisi data target untuk tahun tertentu dan kinerja mesin untuk tipe tertentu, mengubah data target, menghapus data target dan melihat keseluruhan data target.
- *Machine* : meliputi penjadwalan jam produksi dari mesin, mengubah data produksi mesin, menghapus data produksi mesin dan melihat daftar produksi mesin.

3.2.4.1.5 *Menu Daily Data Entry*



Gambar 3-33 : Diagram aliran sistem untuk *data entry*

Menu daily data entry dapat diakses oleh *user* yang mempunyai login administrator dan *data entry*.

Dari menu daily data entry terdapat dua sub menu, yaitu :

- *Production data* : meliputi mengisi data produksi harian, mengubah data produksi, menghapus data dan melihat daftar data produksi harian.
- *Performance data* : meliputi mengisi data *performance* untuk tiap tipe *glass* yang diproduksi pada hari itu, mengubah data *performance*, menghapus data *performance* dan melihat daftar data *performance*.

3.3 Implementasi Sistem

Berdasarkan desain yang telah dibuat pada bab sebelumnya akan dibuatkan aplikasinya. Pada bab ini desain itu akan diwujudkan menjadi sebuah aplikasi yang berbasis web.

3.3.1 Implementasi Data Base

Dalam perancangan basisdata sebagaimana dijelaskan pada bab III, tipe basis data yang digunakan adalah DBMS SQL Server. Setelah selesai dengan perancangannya yang mengacu pada model basis data relasional, maka dilakukan suatu proses untuk melakukan *generating schema*. Dari proses tersebut diperoleh script yang disimpan dalam *file philips.ddl* sebagaimana dapat dilihat pada Lampiran.

3.3.1.1 Pembuatan Koneksi dengan Data Base

Masing-masing halaman web dalam aplikasi ini, khususnya yang menyediakan fasilitas pengolahan data, berhubungan dengan basis data yang telah dibuat. Untuk mengakses basisdata digunakan sebuah file koneksi, yaitu

dbconn.inc dengan *script* seperti yang ditunjukkan dalam gambar dibawah ini. Variabel *server_name*, *db_name* dan *user_id* akan disesuaikan dengan komputer server basis data. *File* ini akan di-*include*-kan ke setiap halaman web yang mengakses basis data tersebut.

```
<%
    var_strConnect ← "Driver={SQL Server}; Server ← server_name; Database ← db_name; UID ← user_id"
%>
```

Gambar 3-34 : *Script* koneksi ke database

3.3.2 Sistem Informasi Produksi dan Maintenance

3.3.2.1 Pembuatan Fasilitas Login

Untuk dapat mengakses menu-menu yang ada dalam sistem ini setiap pengguna harus login dulu. Berdasarkan login ini pengguna dapat dipilah-pilah sesuai dengan haknya masing-masing. Apabila pengguna akan mengakses halaman tertentu dan belum melakukan login maka dia akan diarahkan ke halaman login

The screenshot shows a web application interface. At the top, there is a header with the title "Performance Indicator & Production Loss Analysis" and a Philips logo on the right. Below the title is a navigation menu with four items: "MANAGEMENT", "MAINTENANCE", "PRODUCTION", and "DATA ENTRY". A "Welcome" message is displayed. The main content area contains a login form with three fields: "Login" (text input), "Password" (text input), and "Groups" (a dropdown menu currently showing "Data Entry"). An "Enter" button is located below the form. At the bottom of the page, there is a footer with the text: "copyright © 2003 Dedit. all right reserved" and "best viewed in Internet Explorer at 1024x768, webmaster contact : [webmaster](#)".

Gambar 3-35 : Tampilan halaman login

Fungsi untuk pemeriksaan login pengguna diletakkan dalam satu *file*, yaitu *login.asp* dengan *script* seperti ditunjukkan pada gambar dibawah ini. Fungsi ini akan mencari kesesuaian login yang dimasukkan pengguna dengan isi tabel UserLogin di basis data. Apabila pengguna akan mengakses suatu halaman yang bukan menjadi haknya, maka akan dibawa kembali ke menu utama yang sesuai dengan kelompoknya.

```

<%
    dim query
    session("login")="false"
    session("nama")=""
    session("groups")=""
    set rs = con.execute ("select * from userLogin where UserLogin='"&login&'" and UserPwd='"&pass&'" and
groupsID ='"&groups&'"")
    if (rs.eof) then
        Response.Write "<script>alert('Password / Login salah');window.location='index.asp';</script>"
    else
        session("login")="true"
        session("nama")=rs("UserLogin")
        session("groups")=rs("groupsID")
        Response.Write "Password Match"
        Response.Write "<br>"
        Response.Write "Wait you being redirected :)"
        Response.Write "<br>"
        select case groups
            case "001"
                response.redirect "dataEntry.asp"
            case "002"
                response.redirect "admin.asp"
            case "003"
                response.redirect "management.asp"
            case "004"
                response.redirect "maintenance.asp"
            case "005"
                response.redirect "production.asp"
        end select
    end if
%>

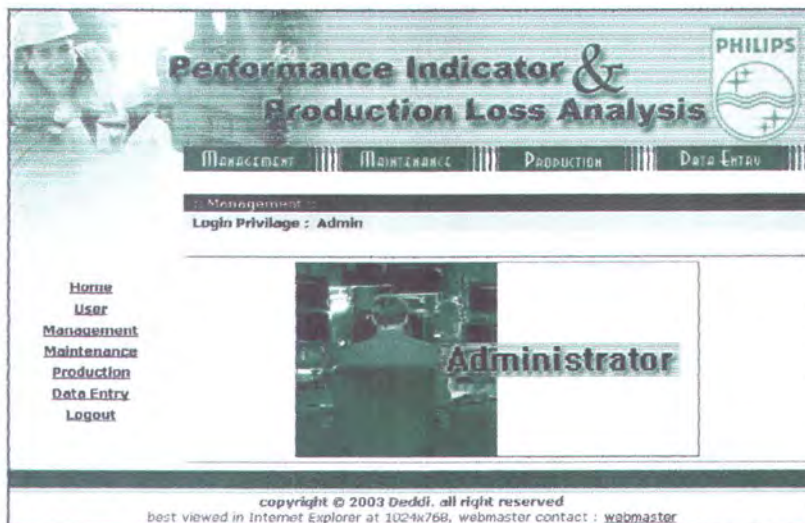
```

Gambar 3-36 : Script proses login

3.3.2.2 Pembuatan Antar Muka Aplikasi

Setiap halaman web dalam aplikasi ini terbagi menjadi empat bagian. Keempat bagian tersebut adalah bagian atas (*header*), bagian menu di samping

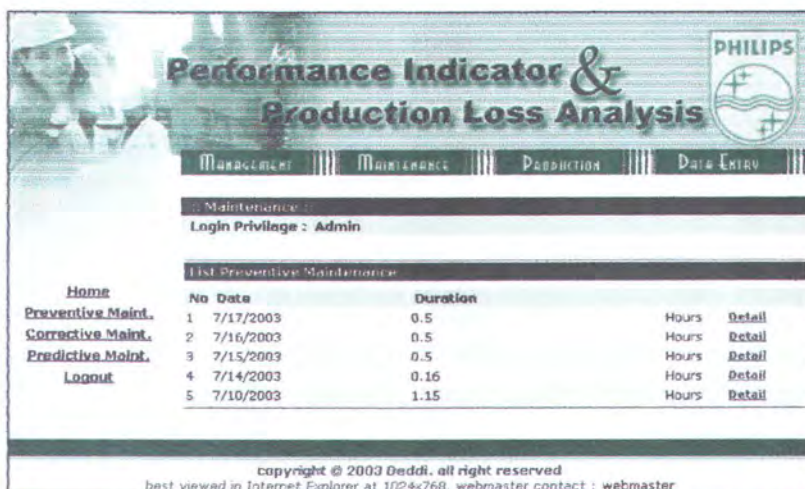
kiri, bagian isi di tengah halaman serta bagian bawah (*footer*). Pada bagian isi akan ditampilkan pengolahan data sesuai dengan fasilitas yang disediakan oleh aplikasi ini.



Gambar 3-37 : Tampilan antar muka pengguna

3.3.2.3 Pembuatan Fasilitas Menampilkan Data

Sesuai dengan perancangan antar muka halaman informasi, ada dua jenis cara menampilkan data pada halaman web, yaitu data ditampilkan dalam tabel dan data ditampilkan secara detail.



Gambar 3-38 : Tampilan untuk menampilkan data dalam tabel

Data yang ditampilkan dalam tabel mungkin tidak lengkap, keterbatasan ukuran tabel pada halaman web. Untuk mengatasinya, salah satu *field* data dibuat sebagai *link* ke halaman web yang akan menampilkan data secara detail. Contoh *script* untuk menampilkan data dalam tabel dapat dilihat pada gambar dibawah ini.

```

<%
Set Recordset ← Server.CreateObject("ADODB.Recordset")
Recordset.ActiveConnection ← var_strConnect
Recordset.Source ← SELECT field_yg_ditampilkan FROM tabel
Recordset.Open() %>
<table>

<%While NOT Recordset.EOF %>
<tr><td><%= (Recordset.Fields.Item(field_key).Value)%></td>
<td><%= (Recordset.Fields.Item(field2).Value)%></td>
<td><a href="form_detail.asp?key=<%= (Recordset.Fields.Item(field_key).Value)%>">Detail</a></td>
</tr></table>
<% Recordset.MoveNext()
Wend
%>

```

Gambar 3-39 : *Script* menampilkan data dalam tabel

Pada *script* tersebut *field* yang merupakan kunci utama (*primary key*) dibuat menjadi *link* untuk menampilkan detail dari data yang bersangkutan.

Performance Indicator & Production Loss Analysis

MANAGEMENT MAINTENANCE PRODUCTION DATA ENTRY

Maintenance - Corrective Maintenance

Login Privilege : Admin

Preventive Maintenance - Total of the Day

Date	:	7/10/2003	
Week	:	200327	
Maintenance Type	- Mechanical PM	:	0.75 Hours
	- Electrical PM	:	0.16 Hours
	- Change Feeder Expandable	:	0.16 Hours
	- Chain Preparation	:	0.08 Hours
Total Maintenance Time	:	1.15	Hours

Add Detail

copyright © 2003 Deddi, all right reserved
best viewed in Internet Explorer at 1024x768, webmaster contact : webmaster

Gambar 3-40 : Tampilan untuk menampilkan data secara detail

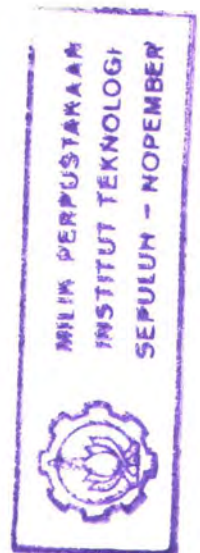
Sedangkan script untuk menampilkan data secara detail dapat dilihat pada gambar dibawah ini.

```
<%
Set Recordset ← Server.CreateObject("ADODB.Recordset")
Recordset.ActiveConnection ← var_strConnect
Recordset.Source ← SELECT semua_field FROM table
                    WHERE field_key ← request.querystring(key)
Recordset.Open()
Response.write("field1 : " & (Recordset.Fields.Item(field1).Value) & "<br>")
Response.write("field2 : " & (Recordset.Fields.Item(field2).Value) & "<br>")
%>
```

Gambar 3-41 : Script menampilkan data secara detail

3.3.2.4 Pembuatan Fasilitas Menambah Data

Pada halaman web untuk menambah data, disediakan kotak teks isian sesuai dengan data yang perlu disimpan.



Gambar 3-42 : Tampilan halaman untuk menambah data

Setelah pengguna mengisikan dan menyimpan data, maka aplikasi akan menuju ke *file* yang menangani penambahan data. *Script*-nya adalah seperti dibawah ini :

```
<%
Set objCommand ← Server.CreateObject("ADODB.Command")
objCommand.ActiveConnection ← var_strConnect
objCommand.CommandText ← INSERT INTO tabel (field1,field2) VALUES (request.form(field1),
```



```

request.form(field2))
objCommand.Execute
objCommand.ActiveConnection.Close
%>

```

Gambar 3-43 : Script menambah data baru

3.3.2.5 Pembuatan Fasilitas Mengubah Data

Pada halaman web untuk mengubah data, disediakan kotak teks isian sesuai dengan data yang akan diubah, yaitu *field* yang bukan sebagai kunci utama dari baris data yang bersangkutan.

Gambar 3-44 : Tampilan halaman untuk mengubah data

Setelah pengguna mengubah dan menyimpan data, maka aplikasi akan menuju ke *file* yang menangani pengubahan data. *Pseudocode*-nya adalah seperti gambar dibawah ini.

```

<%
Set objCommand ← Server.CreateObject("ADODB.Command")
objCommand.ActiveConnection ← var_strConnect
objCommand.CommandText ← UPDATE tabel SET
    field1 ← request.form(field1),
    field2 ← request.form(field2)
    WHERE field_key ← request.form(key)

objCommand.Execute

```

```
objCommand.ActiveConnection.Close
%>
```

Gambar 3-45 : Script untuk meng-update data

3.3.2.6 Pembuatan Fasilitas Menghapus Data

Pengguna aplikasi dapat menghapus data melalui halaman web untuk menampilkan data, baik yang berupa data dalam tabel maupun data secara detil. Setelah pengguna menghapus data, maka aplikasi akan menuju ke *file* yang menangani penghapusan data. *script*-nya adalah seperti gambar berikut.

```
<%
Set objCommand ← Server.CreateObject("ADODB.Command")
objCommand.ActiveConnection ← var_strConnect
objCommand.CommandText ← DELETE FROM tabel WHERE field_key ← request.querystring(key)
objCommand.Execute
objCommand.ActiveConnection.Close
%>
```

Gambar 3-46 : Script untuk menghapus data

3.3.3 Analisa TPM

Pada analisa TPM ini memerlukan data produksi dan data *maintenance*., data produksi akan digunakan untuk melakukan perhitungan analisa *availability* dan analisa *performance rate*. Sedangkan data *maintenance* akan digunakan untuk analisa *direct yield*.

3.3.3.1 Analisa Availability, Direct Yield, Performance Rate dan OEE

Analisa *availability* memerlukan data-data dari tabel-tabel basis data. Untuk ini dilakukan query untuk memperoleh data *Total Time*, *Planned Uptime*, *Break Down Time* dan *Setup dan Retooling Time*. Untuk *query* akan dibuatkan *view* untuk memperoleh total dari masing-masing tabel.

- View untuk memperoleh Total Machine Hours

```
SELECT SUM(dbo.MachProdDet.MPDABS) AS abs, dbo.MachProd.MPID
FROM   dbo.MachProd INNER JOIN
       dbo.MachProdDet ON dbo.MachProd.MPID = dbo.MachProdDet.MPID
GROUP BY dbo.MachProd.MPID
```

Gambar 3-47 : Script untuk view Total Machine Hours

View untuk memperoleh Total Break Down

- PM break down

```
SELECT SUM(dbo.PMDET.PMDSPEND) AS Expr1, dbo.PMDET.PMTIPE, dbo.PM.PMWEEK
FROM   dbo.PM INNER JOIN
       dbo.PMDET ON dbo.PM.PMID = dbo.PMDET.PMID
GROUP BY dbo.PMDET.PMTIPE, dbo.PM.PMWEEK
```

Gambar 3-48 : Script untuk view Total PM break down

- CM break down

```
SELECT SUM(dbo.CMDET.CMDSPEND) AS Expr1, dbo.CMDET.CMTIPE, dbo.CM.CMWEEK
FROM   dbo.CM INNER JOIN
       dbo.CMDET ON dbo.CM.CMID = dbo.CMDET.CMID
GROUP BY dbo.CMDET.CMTIPE, dbo.CM.CMWEEK
```

Gambar 3-49 : Script untuk view Total CM break down

- PDM break down

```
SELECT SUM(dbo.PDMDET.PDMSPEND) AS Expr1, dbo.PDMDET.PDMTIPE, dbo.PDM.PDMWEEK
FROM   dbo.PDM INNER JOIN
       dbo.PDMDET ON dbo.PDM.PDMID = dbo.PDMDET.PDMID
GROUP BY dbo.PDMDET.PDMTIPE, dbo.PDM.PDMWEEK
```

Gambar 3-50 : Script untuk view Total PDM break down

Sedangkan untuk *direct yield* diperlukan data *defect* dan *total production*.

Untuk itu dibuat *view* untuk memperoleh jumlah total *defect* dan produksi *glass*.

```
SELECT  dbo.DAILYPROD.DPWEEK, SUM(dbo.DAILYPROD.DP_MY_CO_OT) AS ot,
        SUM(dbo.DAILYPROD.DP_MY_CO_CMT) AS cm,

        SUM(dbo.DAILYPROD.DP_MY_BO) AS bo, SUM(dbo.DAILYPROD.DP_MY_SO) AS so,
        SUM(dbo.DAILYPROD.DP_MY_CO_MCF) AS mcf,

        SUM(dbo.DAILYPRODTYPE.dpt_gp) AS gp, SUM(dbo.DAILYPRODTYPE.dpt_rv) AS rv,
        SUM(dbo.DAILYPRODTYPE.dpt_rd) AS rd,
        SUM(dbo.DAILYPRODTYPE.dpt_lom) AS lom, SUM(dbo.DAILYPRODTYPE.dpt_su) AS su

FROM    dbo.DAILYPROD INNER JOIN
```



```

dbo.DAILYPRODTYPE ON dbo.DAILYPROD.DPID = dbo.DAILYPRODTYPE.DPID
GROUP BY dbo.DAILYPROD.DP WEEK

```

Gambar 3-51 : Script total data harian produksi

:: Management ::	
Login Privilage : Admin	
Total Productive Maintenance	
Total Productive Maintenance on week : 20031 <input type="button" value="Refresh"/>	
Week : 200330	
Avilability	
Breakdown Time	2.19 %
Set-up, Adjust and Retooling Time	10.67 %
A V	87.14 %
Direct Yield	
Start-up defects	0.90 %
In-process defect	1.54 %
D Y	97.55 %
Performance Rate	
Net Operating Rate	0.78
Speed Ratio	90.52 %
P R	96.60 %
Overall Equipment Effectiveness	
OEE	82.12 %
Overall Production Effectiveness	
Logistic Efficiency	100.00 %
OPE	82.12 %

Gambar 3-52 : Tampilan halaman hasil analisa TPM

3.3.3.2 Data Flag

Data flag digunakan untuk menampilkan detail dari *production yield*, *machine yield*, *change over*, *preventive maintenance* dan *break down*.

View untuk yang digunakan sama dengan yang dipergunakan dalam analisa TPM yang membedakan adalah hasil ditampilkan satu-satu. Hasil ini akan dibandingkan dengan target yang telah ditentukan.

- View untuk menampilkan target produksi :

```

SELECT
dbo.PRODTARGET.PTID AS ID, dbo.PRODTARGET.PT_TY AS ty, dbo.PRODTARGET.PT_MY AS my,
dbo.PRODTARGET.PT_PY AS py,
dbo.PRODTARGET.PT_PY_OV AS ov, dbo.PRODTARGET.PT_PY_GP AS gp,
dbo.PRODTARGET.PT_PY_RD, dbo.PRODTARGET.PT_PY_RV,
dbo.PRODTARGET.PT_PY_LOM, dbo.PRODTARGET.PT_MY_DT, dbo.PRODTARGET.PT_MY_PM,
dbo.PRODTARGET.PT_MY_BD,
dbo.PRODTARGET.PT_MY_NU, dbo.PRODTARGET.PT_MY_CO, dbo.PRODTARGET.PT_MY_STS,
dbo.PRODTARGET.PT_MY_STOP,
dbo.PRODTARGET.PT_MY_BDFREQ, dbo.PRODTARGET.PT_CO_OT, dbo.PRODTARGET.PT_CO_OF,
dbo.PRODTARGET.PT_CO_CMT,
dbo.PRODTARGET.PT_CO_MCF, dbo.PRODTARGET.PT_CO_MLT, dbo.PRODTARGET.PT_PM_CF,
dbo.PRODTARGET.PT_PM_CP,
dbo.PRODTARGET.PT_BD_M, dbo.PRODTARGET.PT_BD_E, dbo.PRODTARGET.PT_BD_F,
dbo.PRODTARGET.PT_BD_SO, dbo.PRODTARGET.PT_MTBFB,
dbo.PRODTARGET.PT_MTR, dbo.PRODTARGET.PT_COM, dbo.PRODTARGET.PT_COT,
dbo.PRODTYPETARGET.PTTID, dbo.PRODTYPETARGET.PTID,
dbo.PRODTYPETARGET.PTT_NAME, dbo.PRODTYPETARGET.PTT_GS,
dbo.PRODTYPETARGET.PTT_BUD, dbo.PRODTYPETARGET.PTT_HA,
dbo.PRODTYPETARGET.PTT_NP
FROM
dbo.PRODTARGET INNER JOIN
dbo.PRODTYPETARGET ON dbo.PRODTARGET.PTID = dbo.PRODTYPETARGET.PTID

```

Gambar 3-53 : Script menampilkan Data Target

Week : 200330	Current	Target
Production Yield		
Over Filling	2 %	2 %
Glass Problem	0.00 %	0.16 %
Reject Dimension	0.02 %	0.6 %
Reject Visual	0.00 %	0.6 %
Los on Machine	0.01 %	0.72 %
Machine Yield		
Preventive Maintenance	1.79 %	1.49 %
Break Down	0.40 %	1.21 %
No Utilities	0.00 %	0.1 %
Change Over	3.33 %	1.67 %
Start Up	10.00 %	3.47 %
Down Time	15.52 %	7.94 %
Change Over		
Change Over Time	48.00 min/CO	33 min/CO
Change Over freq	9 tms/wks	3 tms/wks
Change Mould Time	19.80 min/CO	23 min/CO
Mould Change freq	3 tms/wks	3 tms/wks
Preventive Maintenance		
Prev. Mitenance & Overhaul	1.79 %	1.49 %
Change Feeder	0 %	0.00 %
Chain Preparation	0 %	0 %
Break Down		
Mech. Break Down	0.40 %	0.79 %
Elect. Break Down	0.00 %	0.32 %
Feeder Pin Mechanism	0.00 %	0.1 %
Stock out Spare Part	0.00 %	0 %

Gambar 3-54 : Tampilan halaman untuk Data Flag

3.3.3.3 Machine Technical Statistics

Machine technical statistics menampilkan data mingguan kinerja dari mesin tersebut. Data yang diperlukan dalam menampilkan *Machine Technical Statistics* adalah data *Machine Production Plan* dan data *Maintenance*. Data *maintenance* di-*query* berdasarkan tipe dan jenisnya.

Machine Technical Statistics		
Technical Statistic Machine on week :	200329	<input type="button" value="Refresh"/>
Week	: 200329	
A. Gross Machine Hour	: 168	Hours
1. Absence of program	: 0	Hours
B. Available Machine Hour	: 168	Hours
C. Normal Allowance		
1. Preventive Maintenance	: 0	Hours
2. Break Down - Mechanical	: 0	Hours
- Electrical	: 0	Hours
- Feeder Pin	: 0.16	Hours
3. Change Moulds/Type	: 0.49	Hours
4. Start up until Approval	: 0.25	Hours
5. Black Out / No Utility	: 0	Hours
6. Stock Out	: 0	Hours
	Total : 0.90	Hours
D. Net Machine Hour	: 167.10	Hours
E. Machine Yield	: 99.46 %	

Gambar 3-55 : Tampilan untuk halaman *Machine Technical Statistics*

View untuk mendapatkan total proses *maintenance* dari masing-masing jenis:

```
SELECT SUM(CMDSPEND) AS Expr1, CMTIPE, CMDWEEK
FROM dbo.CMDET
GROUP BY CMTIPE, CMDWEEK
```

Gambar 3-56 : *Script* untuk total CM berdasarkan jenisnya

```
SELECT SUM(PMDSPEND) AS Expr1, PMTIPE, PMDWEEK
FROM dbo.PMDET
GROUP BY PMTIPE, PMDWEEK
```

Gambar 3-57 : *Script* untuk total PM berdasarkan jenisnya


```

SELECT SUM(PDMDSPEND) AS Expr1, PDMDTIPE, PDMDWEEK
FROM dbo.PDMDDET
GROUP BY PDMDTIPE, PDMDWEEK

```

Gambar 3-58 : Script untuk total PDMD berdasarkan jenisnya

3.3.3.4 Production Performance

Production performance akan digunakan untuk menampilkan *production yield* untuk memproduksi tiap jenis *glass*. Analisa ini akan menampilkan seluruh tipe *glass* yang diproduksi selama satu minggu.

Production Performance		
Production Performance on week :	200329	Refresh
Week : 200329		
NO : 1		7/13/2003
Glass Type	: G40C	
Gross Speed	: 1450	pcs/min
Budget	: 1500	pcs/min
Hours Available	: 18.00	hours
Net Production	: 1545000	pcs
Gross Production	: 1566000	pcs
Production Yield	: 98.66 %	

Gambar 3-59 : Tampilan untuk halaman *Production Performance*

3.3.3.5 Grafik

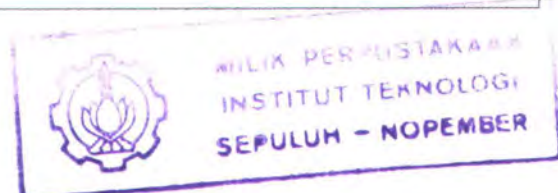
Halaman grafik menampilkan semua hasil perhitungan kedalam bentuk grafik selama kurun waktu satu tahun. Untuk menampilkan hasil grafik pada halaman web dipergunakan OWC (*Office web component*) oleh karena itu untuk menampilkan halaman ini maka komputer *client* harus sudah diinstal program Microsoft Office.

```

<object id=ChartSpace1 classid=CLSID:0002E556-0000-0000-C000-000000000046
style="width:100%;height:350"></object>
<object id=ADOCConnection1 classid=CLSID:00000514-0000-0010-8000-00AA006D2EA4></object>

<script language=vbs>

```



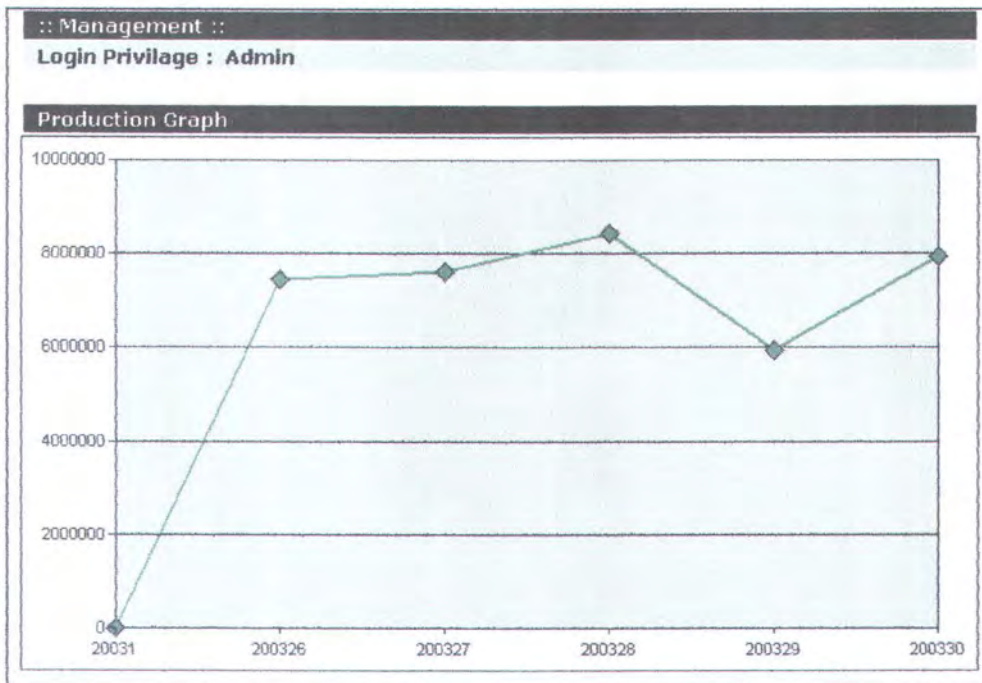
```

Sub Window_OnLoad()
    Dim rs, categories, values, val, c

    categories = ""
    values = ""
    ADOConnection1.Open "DRIVER={SQL Server};Server=BLUESTARORION; Database=philips;
UID=sa; PWD=deddi"
    Set rs = ADOConnection1.Execute("SELECT * FROM [grpProd]")
    rs.MoveFirst
    Do While Not rs.EOF
        categories = categories & rs.Fields(0).Value & Chr(9)
        values = values & rs.Fields(1).Value & Chr(9)
        rs.MoveNext
    Loop
    rs.Close
    ADOConnection1.Close
    categories = Left(categories, Len(categories) - 1)
    values = Left(values, Len(values) - 1)
    ChartSpace1.Clear
    ChartSpace1.Charts.Add
    ChartSpace1.Charts(0).SeriesCollection.Add
    ChartSpace1.Charts(0).SeriesCollection(0).Caption = "Glass Production"
    Set c = ChartSpace1.Constants
    ChartSpace1.Charts(0).SeriesCollection(0).SetData c.chDimCategories, c.chDataLiteral, categories
    ChartSpace1.Charts(0).SeriesCollection(0).SetData c.chDimValues, c.chDataLiteral, values
    ChartSpace1.Charts(0).Type = c.chChartTypeLineMarkers
End Sub

```

Gambar 3-60 : Script membuat grafik dengan office web component



Gambar 3-61 : Tampilan halaman grafik.

Dari *script* diatas akan diperoleh grafik dalam bentuk grafik garis. Dengan kategorinya adalah minggu produksi dan nilainya adalah sesuai dengan yang ingin ditampilkan. Nilai ini diperoleh dari hasil *query* kedalam database sesuai dengan kriteria yang ingin ditampilkan.

3.3.4 Pendukung Keputusan

Untuk pendukung keputusannya digunakan sistem pohon keputusan. Data dari masing-masing analisa akan dibandingkan dengan tabel yang sudah ditentukan.

3.3.4.1 Overall Equipment Effectiveness

Data OEE diperoleh dari hasil perhitungan pada TPM.

```

<%
Set Rec1 ← Server.CreateObject("ADODB.Recordset")
query←SELECT OEE FROM TotalTPM
      GROUP BY week
Rec1.Open query, var_strConnect
While (not Rec1.EOF)
  If oee > 80 then
    If Dee > 85 then
      If oee > 90 then
        If oee > 95 then
          Score = 10
        else
          Score = 8
        End if
      Else
        Score = 6
    End if
  Else
    Score = 4
  End if
Else
  Score = 2
End if
Wend
%>

```

Gambar 3-62 : *Script* untuk menentukan nilai OEE

3.3.4.2 Overall Utilisation

Sedangkan untuk *overall utilisation* diperoleh dari OPE.

```

<%
Set Rec1 ← Server.CreateObject("ADODB.Recordset")
query←SELECT OEE FROM TotalTPM
      GROUP BY week
Rec1.Open query, var_strConnect
While (not Rec1.EOF)
  If oee > 50 then
    If Oee > 60 then
      If oee > 70 then
        If oee > 80 then
          Score = 10
        else
          Score = 8
        End if
      Else
        Score = 6
      End if
    Else
      Score = 4
    End if
  Else
    Score = 2
  End if
Wend
%>

```

Gambar 3-63 : Script untuk menentukan nilai Overall Utilisation

3.3.4.3 Maintenance

Maintenance akan menunjukkan sudah sejauh mana perusahaan mencapai tingkatan yang efektif dalam melaksanakan proses maintenance.

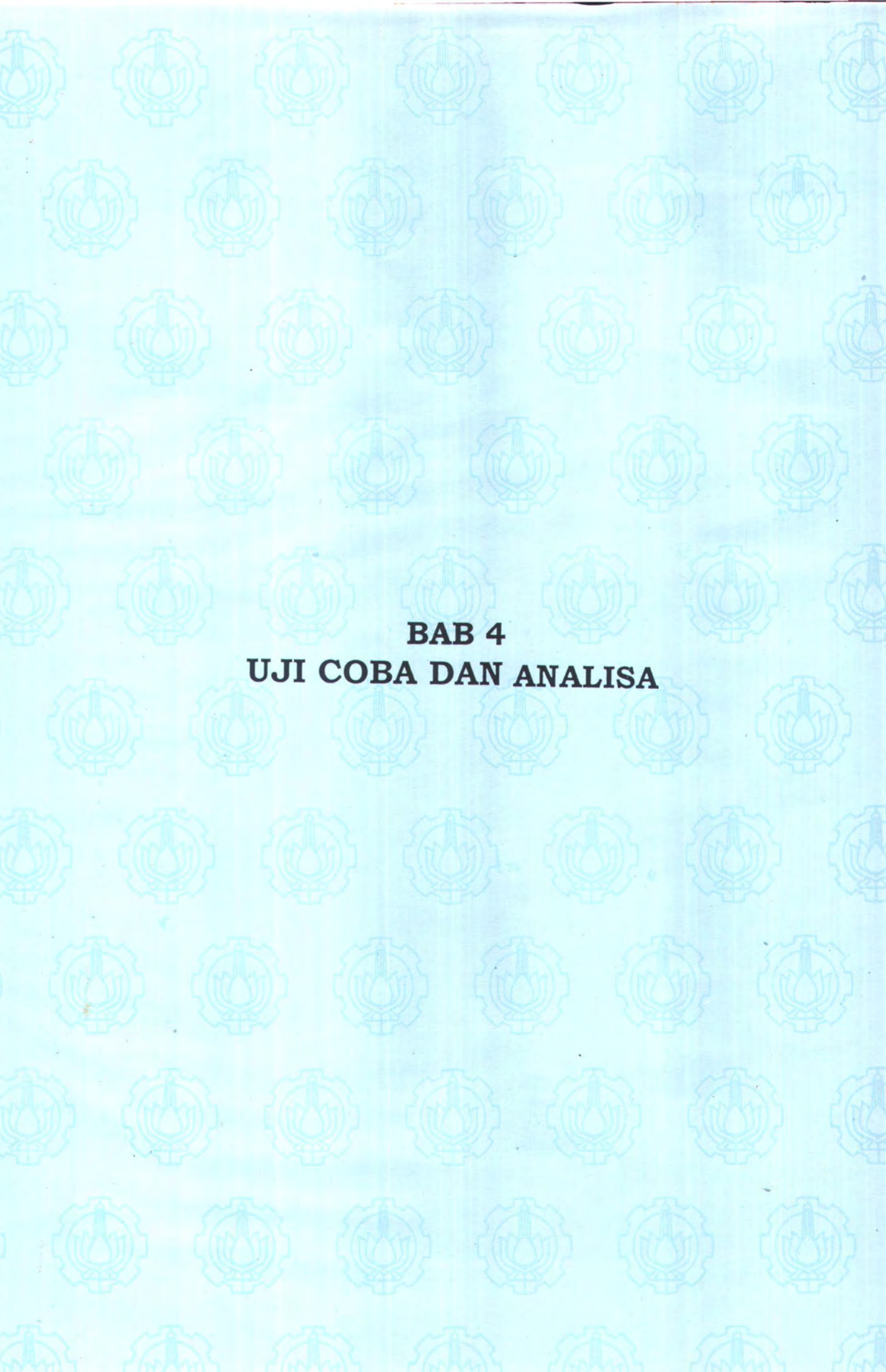
```

<%
Set Rec1 ← Server.CreateObject("ADODB.Recordset")
Query 1 ← SELECT CM FROM CM
Query 2 ← SELECT PM FROM PM
Query 3 ← SELECT PDM FORM PDM
      GROUP BY week
Rec1.Open query, var_strConnect
While (not Rec1.EOF)
  If CM < 90 AND PM < 10 AND PDM <= 0 then
    If CM < 60 AND PM < 40 AND PDM <= 0 then
      If CM < 10 AND PM < 70 AND PDM <= 10 then
        If CM < 5 AND PM < 60 AND PDM <= 35 then
          Score = 10
        else
          Score = 8
        End if
      End if
    End if
  End if
Wend
%>

```

```
End if
Else
  Score = 6
End if
Else
  Score = 4
End if
Else
  Score = 2
End if
Wend
%>
```

Gambar 3-64 : *Script* untuk menentukan nilai *maintenance*



BAB 4
UJI COBA DAN ANALISA

BAB 4

UJI COBA DAN ANALISA

Pada bab ini akan dilakukan uji coba terhadap aplikasi analisa indikator kinerja dan kerugian produksi melalui TPM. Uji coba ini bertujuan untuk mengetahui apakah pengelolaan data pada aplikasi ini telah berhasil dan menghasilkan keluaran sesuai dengan yang diharapkan. Proses uji coba dilakukan pada lingkungan tertentu dan sesuai dengan skenario uji coba yang telah dibuat.

4.1 Lingkungan Uji Coba

Uji coba aplikasi ini dilakukan dalam suatu jaringan lokal (*intranet*) dengan komputer berikut:

- *Server* basis data dan *server* web

Komputer yang dijadikan sebagai *server* basis data dan *server* web mempunyai spesifikasi sebagai berikut:

Nama komputer	: LP-07
IP address	: 10.126.11.107
Prosesor	: Intel Pentium II 450 MHz
Memori fisik	: 256 MB

Sistem operasi yang diinstall adalah *Windows XP Professional* dengan *Internet Information Service 5.0* dan *RDBMS SQL Server 2000*.

- *Komputer client*

Komputer yang digunakan sebagai *client* untuk menjalankan aplikasi ini adalah beberapa komputer yang ada di Laboratorium Pemrograman dengan dilengkapi dengan *browser Internet Explorer 5.0* atau di atasnya dan sudah diinstal program *Microsoft Office*.

4.2 Parameter Yang Diamati

Ektifitas dan nilai guna dari perangkat lunak yang dibuat dan dikembangkan tergantung dari beberapa parameter yang ada, sebagaimana dibawah ini :

- Kemampuan multi user

Pada dasarnya perangkat lunak yang dikembangkan dirancang untuk bisa digunakan oleh banyak pengguna, maka parameter utama yang harus mampu dipenuhi oleh perangkat lunak tersebut haruslah mampu digunakan oleh banyak pengguna sekaligus.

- Fungsionalitas

Fungsionalitas juga merupakan salah satu hal yang harus dipertimbangkan. Maksudnya apakah perangkat lunak yang telah dibuat bisa berfungsi sesuai yang diinginkan. Dan juga apakah aplikasi yang telah dibuat bisa berfungsi sepenuhnya tanpa adanya kesalahan misalnya dalam memasukkan data dan perhitungan analisa.

- Kecepatan aliran data

Penggunaan perangkat lunak oleh banyak pengguna dan kemampuan akses tanpa adanya keterbatasan satu tempat tidak boleh mengurangi kecepatan aliran data antar masing-masing pengguna.



- Efisiensi

Yang terakhir adalah efisiensi dimana perangkat lunak yang dirancang dan dibuat harus benar-benar mampu mengurangi beban pengguna dalam hal ini kecepatan kerja.

4.3 Skenario Uji Coba

4.3.1 Kemampuan Multi User

Untuk kemampuan multi user uji coba dilakukan di Lab Pemrograman – Teknik Informatika. Komputer yang digunakan sebagai web server adalah komputer LP-07 dengan IP 10.126.11.107. Sedangkan untuk Ms SQL server terletak pada komputer LP-14 dengan IP 10.126.11.114. Kedua komputer memiliki spesifikasi yang sama yaitu menggunakan processor Intel Pentium II 450 MHz dan memory sebesar 256 MB. Sedangkan untuk komputer client menggunakan komputer lain yang terdapat di Lab Pemrograman. Spesifikasi untuk komputer client adalah sama, yaitu menggunakan processor Intel Pentium II 450 MHz dan memory sebesar 256 MB. Uji coba dilakukan dari 5 buah computer client yaitu LP-08, LP-09, LP-10 ,LP-11 dan LP-12. Proses uji coba dilakukan dari masing-masing komputer dengan login sebagai administrator, *management*, *maintenance planner*, *production planner* dan *data entry*.

4.3.2 Kemampuan Fungsionalitas

Untuk kemampuan fungsionalitas pengujian dilakukan untuk menambah penjadwalan produksi, penjadwalan maintenance, memasukkan data produksi harian, memasukkan data *breakdown* dan melihat hasil analisa

- Penjadwalan produksi

Untuk uji coba penjadwalan produksi dilakukan untuk penjadwalan produksi minggu ke 30. Penjadwalan produksi untuk minggu ke 30 tidak ada hari libur, sehingga proses produksi memakai waktu penuh. Dan penjadwalan produksi glass-nya adalah sebagai berikut:

Tabel 4-1 : Tabel rencana produksi untuk minggu ke 30

Week 2003/30						
20	21	22	23	24	25	26
July	July	July	July	July	July	July
A60 1204600						
	A60 1204600					
		A55 1152500				
			A55 1152500			
				P45 1167000		
					B35 1045000	
						E60 756025

- Penjadwalan PM

Pada minggu ke 30 ini PM yang dijadwalkan dilakukan pada hari kamis tanggal 24 juli, dengan spesifikasi seperti berikut

Tabel 4-2 : Tabel penjadwalan PM

Task No.	DANNER1.3-PM-3W-E&M
Description	Electrical & Mechanical Preventive Maintenance Danner D1.3 - 3 Weekly
Equipment No.	Danner_13_Crusher_&_Conveyor
Duration	60 mnt

- Data Breakdown

Breakdown yang terjadi selama minggu ke 30 adalah *Corrective maintenance* pada hari selasa tanggal 22 juli:

Tabel 4-3 : Tabel data *breakdown*

WO No.	WO-0001391
Description	Repair Motor Head Shoulder
Equipment	Ribbon Machine
WO Type	CM(2)
Date	2003-07-22
Duration	40 mnt

- Data produksi harian

Data produksi harian dimasukkan setiap hari, untuk ujicoba yang ditampilkan hanya untuk data satu hari produksi, yaitu pada tanggal 20 juli 2003 :

Tabel 4-4 : Tabel data produksi

Date	: 7/20/2003	
Week	: 200330	
Glass Production Type A60C		
Change Over Overtime	: 0.8	hour
Change Mould Time	: 0.33	hour
Mould Change Freq	: 1	tms
Black Out / No Utility	: 0	hour
Stock Out (Gas, Air, Box, Palette)	: 0	hour

Sedangkan untuk data spesifik untuk produksi glass dengan tipe A60 adalah sebagai berikut :

Tabel 4-5 : Tabel data produksi untuk *glass* tipe A60

Production Code	: 7/20/2003	
Glass Type	: A60C	
Production time	: 1301	mnt
Gross Speed	: 1050	pcs/min
Net Production	: 1331000	pcs
Start up until Approval	: 126	
Defect		
Glass Problem	: 0	pcs
Reject Dimension	: 15340	pcs
Reject Visual	: 0	pcs
Loss on Machine	: 8970	pcs

4.3.3 Kecepatan Aliran Data

Uji coba dilakukan dengan menggunakan kelima komputer tersebut secara bersama-sama misalnya untuk memasukkan data-data sesuai fungsionalitas masing-masing pengguna.

4.3.4 Efisiensi

Uji coba dilakukan dengan membandingkan antara sistem yang lama dengan sistem baru yang memanfaatkan perangkat lunak yang telah dibuat.

4.4 Hasil Pengujian dan Analisa

Setelah dilakukan uji coba berdasarkan parameter-parameter yang ada maka dilakukan analisa terhadap apa yang telah dilakukan.

4.4.1 Terhadap Kemampuan Multi User

Penggunaan lima komputer secara bersama-sama yang berada dalam satu jaringan local (*intranet*) tidak mengalami hambatan yang berarti. Aplikasi bisa berjalan dengan lancar tanpa mengalami *delay* yang berlebihan (komputer macet). Hal ini dimungkinkan karena server web yang ada mampu melayani permintaan

dari pengguna secara bersamaan. Juga karena adanya teknologi jaringan yang memungkinkan satu aplikasi bisa diakses dari beberapa tempat yang berbeda.

4.4.2 Terhadap Fungsionalitas

Setelah memasukkan data kedalam aplikasi, data yang digunakan adalah selama minggu ke 30 untuk proses memasukkan data masing-masing *user* tidak mengalami kesulitan dan hambatan. Dan hasil perhitungan sudah sesuai dengan yang diharapkan dan SOP. Dari hasil tadi dapat diperoleh hasil perhitungan sebagai berikut :

- Data Flag

Tabel 4-6 : Tabel *data flag* minggu ke 30

Week : 200330	Current		Target	
Production Yield				
Over Filling	2	%	2	%
Glass Problem	0.00	%	0.16	%
Reject Dimension	0.02	%	0.6	%
Reject Visual	0.00	%	0.6	%
Los on Machine	0.01	%	0.72	%
Machine Yield				
Preventive Maintenance	1.79	%	1.49	%
Break Down	0.40	%	1.21	%
No Utilities	0.00	%	0.1	%
Change Over	3.33	%	1.67	%
Start Up	10.00	%	3.47	%
Down Time	15.52	%	7.94	%
Change Over				
Change Over Time	48.00	min/CO	33	min/CO
Change Over freq	9	tms/wks	3	tms/wks
Change Mould Time	19.80	min/CO	23	min/CO

- Technical Statistic Machine

Tabel 4-7 : Tabel *technical statistic machine* minggu ke 30

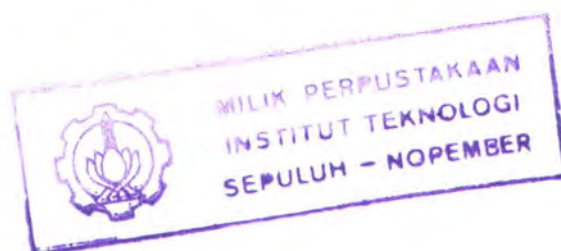
Week	: 200330	
A. Gross Machine Hour	: 168	Hours
1. Absence of program	: 0	Hours
B. Available Machine Hour	: 168	Hours
C. Normal Allowance		
1. Preventive Maintenance	: 3	Hours
2. Break Down - Mechanical	: 0.68	Hours
- Electrical	: 0	Hours
- Feeder Pin	: 0	Hours
3. Change Moulds/Type	: 5.6	Hours
4. Start up until Approval	: 16.80	Hours
5. Black Out / No Utility	: 0	Hours
6. Stock Out	: 0	Hours
Total	: 26.08	Hours
D. Net Machine Hour	: 141.92	Hours
E. Machine Yield	: 84.48 %	

- Production Performance

Tabel 4-8 : Tabel *production performance* minggu ke 30

Week : 200330		
NO : 1	7/20/2003	
Glass Type	: A60C	
Gross Speed	: 1050	pcs/min
Budget	: 1200	pcs/min
Hours Available	: 21.68	hours
Net Production	: 1331000	pcs
Gross Production	: 1366050	pcs
Production Yield	: 97.43 %	
<hr/>		
NO : 2	7/21/2003	
Glass Type	: A60C	
Gross Speed	: 1050	pcs/min
Budget	: 1200	pcs/min
Hours Available	: 21.68	Hours
Net Production	: 1331000	Pcs
Gross Production	: 1366050	Pcs
Production Yield	: 97.43 %	

NO : 3	7/22/2003	
Glass Type	: A55C	
Gross Speed	: 1075	pcs/min
Budget	: 1200	pcs/min
Hours Available	: 21.37	Hours
Net Production	: 1320000	Pcs
Gross Production	: 1378150	Pcs
Production Yield	: 95.78 %	
NO : 4	7/24/2003	
Glass Type	: P45C	
Gross Speed	: 1175	pcs/min
Budget	: 1200	pcs/min
Hours Available	: 17.75	hours
Net Production	: 1220000	Pcs
Gross Production	: 1251375	Pcs
Production Yield	: 97.49 %	
NO : 5	7/25/2003	
Glass Type	: B35C	
Gross Speed	: 1075	Pcs/min
Budget	: 1200	Pcs/min
Hours Available	: 18.62	hours
Net Production	: 1173000	Pcs
Gross Production	: 1200775	Pcs
Production Yield	: 97.69 %	
NO : 6	7/26/2003	
Glass Type	: E60C	
Gross Speed	: 1090	Pcs/min
Budget	: 1200	Pcs/min
Hours Available	: 13.67	hours
Net Production	: 868000	Pcs
Gross Production	: 893800	Pcs
Production Yield	: 97.11 %	
NO : 7	7/26/2003	
Glass Type	: TP45	
Gross Speed	: 1100	Pcs/min
Budget	: 1200	Pcs/min
Hours Available	: 5.70	hours
Net Production	: 367000	Pcs
Gross Production	: 376200	Pcs
Production Yield	: 97.55 %	



- TPM

Tabel 4-9 : Tabel TPM untuk minggu ke 30

Week : 200330		
Avilability		
Breakdown Time	2.19	%
Set-up,Adjust and Retooling Time	10.67	%
A V	87.14	%
Direct Yield		
Start-up defects	0.90	%
In-process defect	1.54	%
D Y	97.55	%
Performance Rate		
Net Operating Rate	0.78	
Speed Ratio	90.65	%
P R	96.78	%
Overall Equipment Effectiveness		
OEE	82.27	%
Overall Production Effectiveness		
Logistic Efficiency	100.00	%
OPE	82.27	%


4.4.3 Terhadap Aliran Data

Penggunaan lima komputer secara bersama-sama tidak membuat aliran data menjadi lambat. Proses penjadwalan produksi, penjadwalan PM dan mengisi data produksi dan breakdown harian dapat dilakukan tanpa adanya hambatan. Hal ini dimungkinkan karena server web maupun database yang ada mampu menangani lima pengguna secara bersama-sama.

4.4.4 Terhadap Efisiensi

Dibandingkan dengan sistem lama yang ada, aplikasi yang dibuat ini dapat meningkatkan kecepatan update data dan mempermudah pihak management untuk

memperoleh data yang diperlukan berkaitan dengan proses produksi dan proses maintenance. Biasanya untuk laporan produksi dan *maintenance* diberikan ke pihak management dalam kurun waktu satu minggu sekali atau satu bulan sekali. Tetapi dengan adanya aplikasi ini pihak management dapat melihat data produksi dan *maintenance* secara *real time*. Dan untuk analisa TMP yang meliputi *Availability*, *Direct Yield* dan *Performance Rate* pihak *management* dapat langsung memperolehnya dan ditambah lagi bisa mengetahui tingkatan efisiensi dari perusahaan.



BAB 5
KESIMPULAN DAN SARAN

BAB 5

KESIMPULAN DAN SARAN

Bab ini berisi beberapa kesimpulan dari tugas akhir dan kemungkinan pengembangan lebih lanjut yang dapat dilakukan dari tugas akhir ini

5.1 Kesimpulan

Beberapa kesimpulan yang bisa ditarik dari hasil pengamatan selama perancangan, pembuatan, implementasi hingga uji coba aplikasi ini adalah sebagai berikut :

1. Kerugian produksi dan indikator kinerja dapat ditentukan dari hasil perhitungan dari *Overall Equipment Effectiveness, Availability, Production Yield* dan *Performance Rate*
2. Proses integrasi *Corrective Maintenance, Preventive Maintenance* dan *Predictive Maintenance* dengan cara memasukkan data *maintenance* tersebut ke dalam proses perhitungan, sehingga dapat diketahui besarnya pengaruh dari proses *maintenance* tersebut kedalam proses produksi.
3. Dari hasil perhitungan *Overall Equipment Effectiveness, Availability, Production Yield* dan *Performance Rate*, akan diperoleh tingkatan dari kinerja dan kerugian produksi. Sehingga pihak *management* dapat mengambil kebijaksanaan baru dalam proses produksi dan *maintenance*.

4. Proses integrasi program *maintenance* dan produksi dapat memudahkan pihak *management* untuk memantau kinerja dari masing-masing departemen dan pihak *management* dapat memperoleh data produksi dan *maintenance* secara cepat dan *on-line*.

5.2 Saran

Beberapa pengembangan yang bisa dilakukan dalam tugas akhir ini adalah sebagai berikut :

1. Memberikan pilihan kastomisasi pada aplikasi sehingga aplikasi ini dapat dipergunakan oleh setiap departemen dan bidang kerja yang berbeda-beda.
2. Mengikut sertakan seluruh departemen produksi dan *maintenance* yang ada di PT Philips Ralins Electronics Surabaya, sehingga dapat diketahui kinerja dan kerugian produksi dari perusahaan secara keseluruhan.





DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [VJ98] Venne, J.L.M. Van de, *Total Productive Maintenance, Total Productivity Management, Self Assessment tool, version 1.0*, Nederlandse Philips Bedrijven B.V, 1998
- [NS89] Nakajime, Seiichi, *TPM Development Program – Implementing Total Productive Maintenance*, Productivity Press, Inc, 1989
- [AG99] Agung, Gregorius, *Membuat Homepage Interaktif dengan CGI / Perl*, PT Elex Media Komputindo, 1999
- [PY01] Purwanto, Yudhi, *Pemrograman Web dengan PHP*, PT Elex Media Komputindo, 2001
- [EF00] Effendi, Handaya, *Pemrograman Dynamic HTML*, PT Elex Media Komputindo, 2000
- [RJ01] Ramalho, Jose, *SQL Server 7.0*, PT Elex Media Komputindo, 2001
- [EM00] Elmasri, Ramez A, Navathe, Shamkant B, *Fundamental of Database Systems (3rd Edition)*, Prentice Hall, 2000
- [FA98] Fedorov A, Francis B, Harrison R, Homer A, Murphy S, Smith R, Sussman D, Wood S, *Professional Active Server Pages 2.0*, Wrox Press, 1998
- [QT02] Quatrani, Terry, *Visual Modelling With Rational Rose 2002 And UML*, Addison-wesley, 2002

- [KU03] Kurniawan, Ujik, *Perancangan dan Pembuatan Perangkat Lunak Otomatisasi Aliran Kerja Aircraft Maintenance Facility di PT. Merpati Nusantara Airlines*, Informatika ITS, 2003



LAMPIRAN

LAMPIRAN I

Script DDL yang digunakan untuk membuat database.

```
/*=====*/
/* Table: CM */
/*=====*/
create table CM (
  DID          int          not null,
  CMID         int          not null,
  CMTIME       datetime    null,
  CMNAME       varchar(25)  null,
  CMEQUIP      varchar(25)  null,
  CMSPEND      decimal(10,2) null,
  CMTIPE       varchar(25)  null,
  constraint PK_CM primary key (DID, CMID)
)
```

```
/*=====*/
/* Table: DAILYPROD */
/*=====*/
create table DAILYPROD (
  DID          int          not null,
  DPID         int          not null,
  DPCOT        decimal(10,2) null,
  DPCMT        decimal(10,2) null,
  DPMCF        decimal(10,2) null,
  DPBO         decimal(10,2) null,
  DPSO         decimal(10,2) null,
  constraint PK_DAILYPROD primary key (DID, DPID)
)
```

```
/*=====*/
/* Table: DAILYPRODTYPE */
/*=====*/
create table DAILYPRODTYPE (
  DID          int          not null,
  DPID         int          not null,
  DPTID        int          not null,
  DPTNAME      varchar(25)  null,
  DPTGS        decimal(10,2) null,
  DPTBUD       decimal(10,2) null,
  DPTHA        decimal(10,2) null,
  DPTNP        decimal(10,2) null,
  DPTGP        decimal(10,2) null,
  DPTRV        decimal(10,2) null,
  DPTRD        decimal(10,2) null,
  DPTLOM       decimal(10,2) null,
  DPTSU        decimal(10,2) null,
  DPTDUR       decimal(10,2) null,
  DPTWEEK      int          null,
  DPTSTART     datetime    null,
  DPTEND       datetime    null,
  constraint PK_DAILYPRODTYPE primary key (DID, DPID, DPTID)
)
```

```

/*=====*/
/* Table: DAYPROD                                     */
/*=====*/
create table DAYPROD (
DID          int          not null,
DDAY         datetime     null,
DWEEK        int          null,
DYEAR        int          null,
constraint PK_DAYPROD primary key (DID)
)

```

```

/*=====*/
/* Table: GLASSTYPE                                   */
/*=====*/
create table GLASSTYPE (
GTID         int          not null,
GROUPSID     char(6)      null,
GTNAME       varchar(15)  null,
GTDES        varchar(100) null,
constraint PK_GLASSTYPE primary key (GTID)
)

```

```

/*=====*/
/* Table: GROUPS                                     */
/*=====*/
create table GROUPS (
GROUPSID     char(6)      not null,
GROUPSNAME   varchar(15)  null,
constraint PK_GROUPS primary key (GROUPSID)
)

```

```

/*=====*/
/* Table: MACHPROD                                    */
/*=====*/
create table MACHPROD (
DID          int          not null,
MPID         int          not null,
MPAM         decimal(10,2) null,
MPABS        decimal(10,2) null,
MPGM         decimal(10,2) null,
constraint PK_MACHPROD primary key (DID, MPID)
)

```

```

/*=====*/
/* Table: MACHPRODDDET                               */
/*=====*/
create table MACHPRODDDET (
DID          int          not null,
MPID         int          not null,
MPDID        int          null,
MPDABS       decimal(10,2) null,
constraint PK_MACHPRODDDET primary key (DID, MPID)
)

```

```

/*=====*/
/* Table: PDM                                         */
/*=====*/
create table PDM (
DID          int          not null,

```

```

PDMID                int                not null,
PDMTIME              datetime           null,
PDMNAME              varchar(25)        null,
PDMEQUIP             varchar(25)        null,
PDMSPEND             decimal(10,2)      null,
PDMTIPE              varchar(25)        null,
constraint PK_PDM primary key (DID, PDMID)
)

```

```

/*=====*/
/* Table: PM */
/*=====*/
create table PM (
DID                int                not null,
PMID               int                not null,
PMTIME            datetime           null,
PMNAME            varchar(25)        null,
PMEQUIP           varchar(25)        null,
PMSPEND           decimal(10,2)      null,
PMTIPE            varchar(25)        null,
constraint PK_PM primary key (DID, PMID)
)

```

```

/*=====*/
/* Table: PRODTARGET */
/*=====*/
create table PRODTARGET (
DID                int                not null,
PTID               int                not null,
PT_PY_OV           decimal(10,2)      null,
PT_PY_GP           decimal(10,2)      null,
PT_PY_RD           decimal(10,2)      null,
PT_PY_RV           decimal(10,2)      null,
PT_PY_LOM          decimal(10,2)      null,
PT_MY_PM           decimal(10,2)      null,
PT_MY_BD_MECH      decimal(10,2)      null,
PT_MY_BD_ELC       decimal(10,2)      null,
PT_MY_BD_FP        decimal(10,2)      null,
PT_MY_BD_SO        decimal(10,2)      null,
PT_MY_CO_OT        decimal(10,2)      null,
PT_MY_CO_CMT       decimal(10,2)      null,
PT_MY_CO_MCF       decimal(10,2)      null,
PT_MY_BO           decimal(10,2)      null,
PT_MY_SU           decimal(10,2)      null,
PT_MY_SO           decimal(10,2)      null,
constraint PK_PRODTARGET primary key (DID, PTID)
)

```

```

/*=====*/
/* Table: PRODTARGETTYPE */
/*=====*/
create table PRODTARGETTYPE (
DID                int                not null,
PTID               int                not null,
PTTID              int                not null,
PTTBUD             decimal(10,2)      null,
constraint PK_PRODTARGETTYPE primary key (DID, PTID, PTTID)
)

```




```
/*=====*/
/* Table: PRODTYPE */
/*=====*/
create table PRODTYPE (
GTID          int          not null,
DID          int          not null,
PTYID       int          not null,
PTYJML      decimal(10,2) null,
PTYNAME     varchar(15)   null,
constraint PK_PRODTYPE primary key (GTID, DID, PTYID)
)

/*=====*/
/* Table: USERLOGIN */
/*=====*/
create table USERLOGIN (
GROUPSID     char(6)      not null,
UID         int          not null,
UNAME       varchar(30)   null,
ULOGIN     varchar(15)   null,
UPWD       varchar(15)   null,
constraint PK_USERLOGIN primary key (GROUPSID, UID)
)
```

LAMPIRAN II

