

BAB V PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai tahapan dalam perancangan dan pengembangan sistem yaitu mulai dari desain, implementasi hingga pengujian sistem. Bab ini merupakan bagian penting dalam suatu proses perancangan sistem yang diharapkan dapat menghasilkan sistem informasi manajemen penjadwalan *preventive maintenance* yang sesuai dengan kebutuhan *user*.

5.1 Desain Sistem

Tujuan dari desain sistem adalah untuk menciptakan model fisik dari sistem yang memenuhi persyaratan desain yang ditetapkan selama fase analisis sistem. Tahap ini merupakan sebuah model logis dari sistem yang baru dikembangkan dari tahap analisa sebelumnya. Langkah yang akan dikerjakan pada tahap ini mencakup desain *database*, *user interface* dan desain algoritma. Untuk lebih lengkapnya akan dijabarkan dalam sub bab berikut ini.

5.1.1 Desain Database

Pada sub - sub bab ini dilakukan desain *database* untuk sistem yang akan dirancang. Dalam tahap desain *database* secara umum, perlu dilakukan identifikasi terlebih dahulu *file-file* yang diperlukan oleh sistem informasi. Langkah-langkah untuk mendesain *database* adalah dengan desain *database* logis dan desain *database* fisik.

5.1.1.1 Desain Database Logis

Fungsi – fungsi dari sistem dijelaskan kepada *user* melalui desain *database* logis. Pada tahapan ini *user* bisa mengetahui bagaimana sistem bekerja secara logika. Desain *database* logis ini digambarkan dengan menggunakan ERD, dimana ERD dapat menggambarkan entitas dan atribut yang terlibat dalam sistem. Berikut tahapan – tahapan dalam pembuatan ERD .

1. List Entity

Setiap entitas merupakan calon tabel yang merupakan komponen penyusun utama dalam sebuah *database*. Daftar entitas perlu disusun agar dapat diketahui data apa saja

yang akan terlibat dalam *database*. Identifikasi terhadap entitas adalah langkah pertama yang harus dikerjakan dalam membuat daftar entitas. Berikut Tabel 5.1 adalah antitas dan atribut yang terlibat pada sistem informasi manajemen *preventive maintenance*.

Tabel 5.1 Daftar Entitas dan Atribut ERD

Entitas	Atribut
Mesin	<u>kode mesin</u> , nama_mesin, tahun_pembuatan, type, kekuatan
Sparepart	<u>kode sparepart</u> , kode_mesin, nama_sparepart, tanggal_masuk, jumlah_stock, stock_minimun
Mekanik	<u>kode mekanik</u> , nama_mekanik, telp, alamat, tahun_masuk
Jadwal PM	<u>kode pm</u> , kode_mesin, kode_sparepart, tgl_pm, keterangan
Tindakan Kerusakan	<u>kode tindakan</u> , kode_kerusakan, kode_mesin, kode_sparepart, jumlah_sparepart, kode_mekanik, tgl_mulai, jam_mulai, tgl_selesai, jam_selesai
Kerusakan mesin	<u>Kode kerusakan</u> , kode_mesin, tgl_kerusakan, kode_sparepart, nama_sparepart, keterangan
Tindakan PM	<u>kode tindakan pm</u> , kode_pm, kode_mesin, kode_sparepart, jumlah_sparepart, kode_mekanik, tgl_pm_terencana, tgl_pm_aktual

Entitas yang terdapat dalam sistem informasi manajemen *preventive maintenance* merupakan data utama yang merupakan nama tabel dalam *database*. Sedangkan atribut merupakan karakteristik yang menyertai entitas, misalnya dalam entitas mekanik disertai beberapa atribut antara lain kode_mekanik, nama_mekanik, telp, alamat, dan tahun_masuk

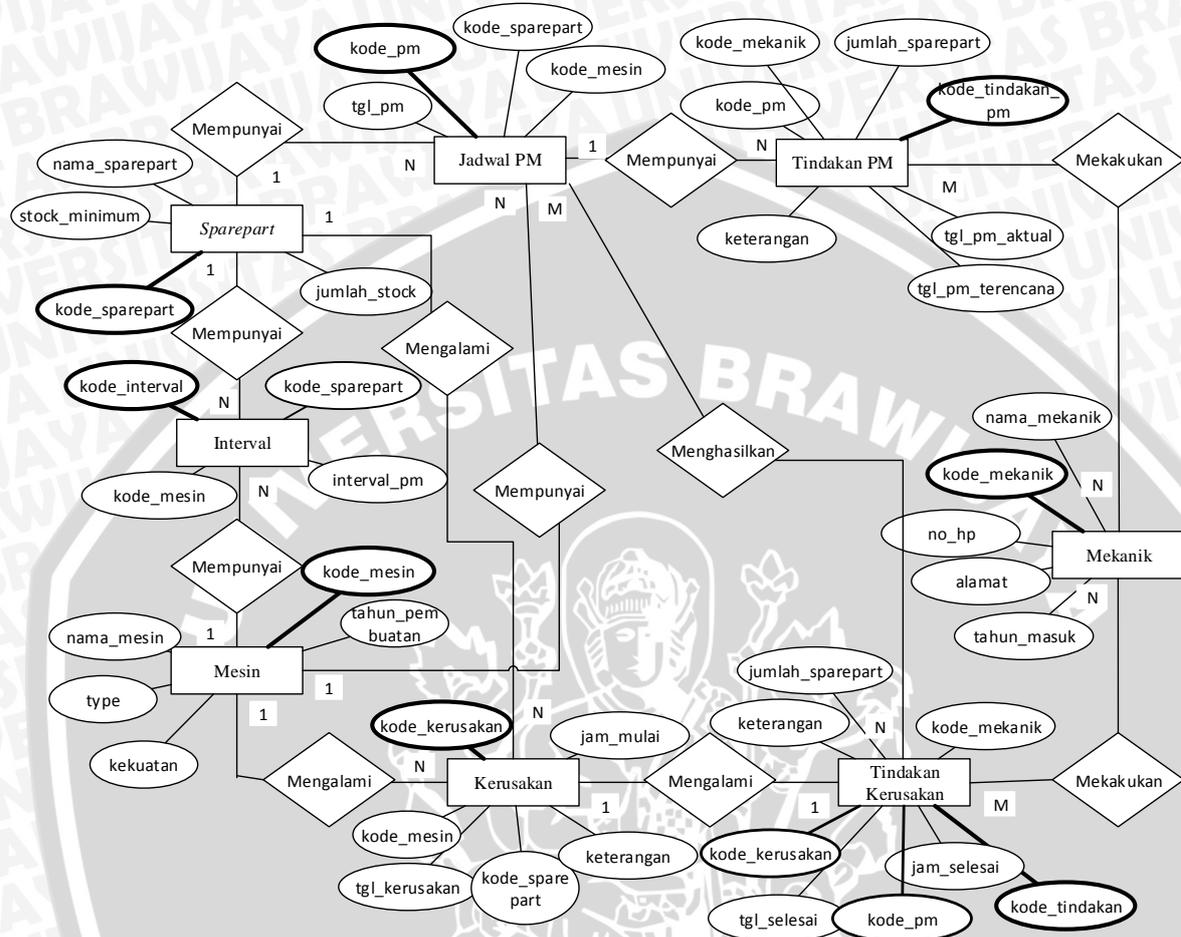
2. Entity-Relational Diagram (ERD)

Dalam penelitian perancangan sistem ini, hubungan atau *relationship* antar objek/entitas digambarkan melalui ERD. Ada beberapa macam hubungan/relasi yang terjadi antar entitas yang sudah dijelaskan pada Bab 2. ERD sangatlah penting, karena pada tahap ini semua entitas diintegrasikan menjadi satu sehingga dapat menghasilkan informasi yang dibutuhkan oleh *user*. Sebelum menggambar ERD, perlu diidentifikasi kardinalitas atau jenis relasi untuk setiap entitas yang terlibat seperti pada Tabel 5.2

Tabel 5.2 Identifikasi Relasi

Entitas	Relasi	Entitas	Derajat Relasi Maks -Min
Mesin	mengalami	Kerusakan Mesin	(1, N)
Mesin	mempunyai	Interval	(1, N)
Mekanik	melakukan	Tindakan Kerusakan	(M, N)
Mekanik	melakukan	Tindakan PM	(M, N)
Sparepart	mempunyai	Interval	(1, N)
Sparepart	mengalami	Kerusakan	(1, N)
Jadwal PM	mempunyai	Tindakan PM	(1, N)
Kerusakan	mengalami	Tindakan Kerusakan	(1, 1)
Tindakan Kerusakan	Mempunyai	Jadwal PM	(M, N)

Berdasarkan identifikasi jenis relasi yang dilakukan, kardinalitas atau jenis relasi yang terjadi terhadap masing-masing entitas dapat digambarkan pada Gambar ERD 5.1 disertai dengan atribut-atribut yang menyertai masing-masing entitas



Gambar 5.1 Diagram E-R dengan atribut

Berdasarkan Gambar 5.1 dan Tabel 5.2, hubungan antar masing-masing entitas yang terlibat dapat dijelaskan sebagai berikut:

- Satu mesin dapat mengalami 1 atau lebih kerusakan mesin.
- Satu mesin dapat mempunyai 1 atau lebih interval
- Satu tindakan kerusakan dapat ditangani 1 atau lebih mekanik.
- Satu tindakan PM dapat ditangani 1 atau lebih mekanik.
- Satu *sparepart* mengalami 1 atau lebih kerusakan mesin
- Satu *sparepart* mempunyai 1 atau lebih interval
- Satu jadwal PM mempunyai 1 atau lebih tindakan PM
- Satu kerusakan mempunyai 1 tindakan kerusakan
- Satu tindakan kerusakan mempunyai 1 atau lebih jadwal PM

5.1.1.2 Desain Database Fisik

Desain fisik merupakan aktualisasi dari desain *database* logis yang sangat bergantung dengan *software* yang dipakai. Oleh karena itu pada tahap ini diawali dengan pemilihan *software* yang akan dipakai terlebih dahulu yaitu *Microsoft Access 2013*. Setelah dilakukan pemilihan maka dirancanglah struktur tabel sesuai dengan kebutuhan untuk melakukan penyimpanan data. Pada tahap ini, *entity* sudah berubah menjadi tabel sesuai dengan format tabel pada *Microsoft Access 2013*. Berikut merupakan rancangan struktur *database* dimana entitas sudah dirubah menjadi tabel.

1. Entitas Mesin

Entitas mesin berisi data – data mesin yang dipilih menjadi objek penelitian di PT XYZ. Tabel 5.3 merupakan desain spesifikasi fisik untuk entitas mesin.

Tabel 5.3 Desain Database Entitas Mesin

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Key</i>
kode_mesin	<i>Short Text</i>	6	Kode mesin	PK
nama_mesin	<i>Short Text</i>	40	Nama mesin	
tahun_pembuatan	<i>Number</i>	<i>Integer</i>	Tahun pembuatan	
Type	<i>Short Text</i>	20	Type mesin	
kekuatan	<i>Short Text</i>	20	Kekuatan mesin	

2. Entitas Sparepart

Entitas *sparepart* berisi data *sparepart* dari mesin produksi di PT XYZ. Pada tabel ini informasi *sparepart* seperti nama *sparepart*, *stock* tersedia maupun batas *stock* minimal seperti yang ditunjukkan pada tabel 5.4

Tabel 5.4 Desain Database Entitas Sparepart

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Key</i>
kode_sparepart	<i>Short Text</i>	15	Kode <i>Sparepart</i>	PK
nama_sparepart	<i>Short Text</i>	35	Nama <i>Sparepart</i>	
jumlah_stock	<i>Number</i>	<i>Integer</i>	Jumlah Stok saat ini	
jumlah_minimum	<i>Number</i>	<i>Integer</i>	Jumlah minimal yang harus ada	

3. Entitas Mekanik

Entitas mekanik merupakan data yang berisi informasi mekanik *maintenance* meliputi kode mekanik, nama, telepon, alamat dan tahun berapa mekanik tersebut masuk di bagian *maintenance*. Tabel 5.5 merupakan desain spesifikasi fisik untuk entitas mekanik.

Tabel 5.5 Desain Database Entitas Mekanik

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Key</i>
kode_mekanik	<i>Short Text</i>	4	Kode mekanik	PK
nama_mekanik	<i>Short Text</i>	30	Nama mekanik	
Telp	<i>Number</i>	<i>Double</i>	Nomor Telepon mekanik	
Alamat	<i>Short Text</i>	150	Alamat mekanik	
tahun_masuk	<i>Number</i>	<i>Integer</i>	Tahun mulai awal masuk kerja	

4. Entitas Kerusakan Mesin

Entitas kerusakan merupakan tabel data *history* kerusakan mesin yang pernah terjadi pada mesin produksi di PT XYZ. Tabel 5.6 merupakan desain spesifikasi fisik untuk entitas kerusakan mesin.

Tabel 5.6 Desain Database Entitas Kerusakan Mesin

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Key</i>
kode_kerusakan	Short Text	10	Kode Kerusakan Mesin	PK
kode_mesin	Short Text	6	Kode Mesin	FK
tanggal_kerusakan	Date/Time	Long Date	Tanggal kerusakan mesin	
kode_sparepart	Short Text	15	Kode Sparepart	FK
nama_sparepart	Short Text	35	Nama Sparepart	
keterangan	Long Text		Keterangan	

5. Entitas Tindakan Kerusakan

Entitas tindakan kerusakan merupakan tabel data informasi tindakan kerusakan yang harus dilakukan untuk mesin. Tabel 5.7 ini merupakan desain spesifikasi fisik untuk entitas tindakan kerusakan.

Tabel 5.7 Desain Database Entitas Tindakan Kerusakan

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Key</i>
kode_tindakan	Short Text	10	Kode Tindakan Kerusakan	PK
kode_kerusakan	Short Text	10	Kode Kerusakan Mesin	
kode_mesin	Short Text	6	Kode mesin	
kode_sparepart	Short Text	15	Kode Sparepart	
jumlah_sparepart	Number	Integer	Jumlah sparepart terpakai	
kode_mekanik	Short Text	4	Kode mekanik	
tgl_mulai	Date/Time	Long Date	Tanggal mulai tindakan	
jam_mulai	Date/Time	Short Time	Jam mulai tindakan	
tgl_selesai	Date/Time	Long date	Tanggal selesai tindakan	
jam_selesai	Date/Time	Short Time	Jam mulai tindakan	
keterangan	Long Text		Keterangan tindakan	

6. Entitas Jadwal Preventive maintenance

Entitas jadwal *Preventive maintenance* berisi jadwal perawatan *preventive* untuk mesin produksi PT XYZ. Tabel 5.8 merupakan desain spesifikasi fisik untuk entitas jadwal *preventive maintenance*.

Tabel 5.8 Desain Database Entitas Jadwal Perawatan Mesin

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Key</i>
kode_pm	Short Text	6	Kode preventive maintenance	PK
kode_mesin	Short Text	6	Kode mesin	FK
kode_sparepart	Short Text	15	Kode Sparepart	FK
tanggal_pm	Date/Time	Long Date	Tanggal preventive maintenace	

7. Entitas Tindakan PM

Entitas tindakan pm merupakan tindak lanjut dari jadwal PM. Semua kegiatan PM akan dicatat pada entitas ini. Berikut Tabel 5.9 merupakan desain spesifikasi fisik untuk entitas tindakan PM

Tabel 5.9 Desain *Database* Entitas Tindakan PM

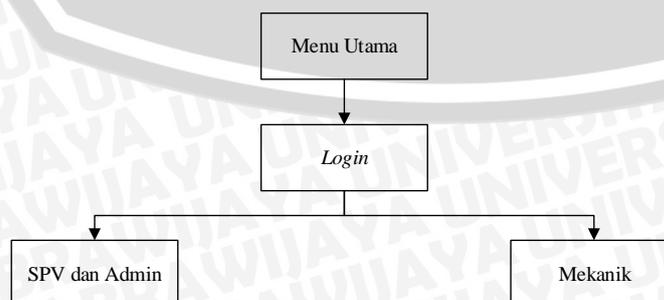
<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Key</i>
kode_tindakan_pm	Short Text	10	Kode Tindakan PM	PK
kode_pm	Short Text	10	Kode PM	
kode_mesin	Short Text	6	Kode mesin	
kode_sparepart	Short Text	15	Kode Sparepart	
jumlah_sparepart	Number	Integer	Jumlah sparepart terpakai	
kode_mekanik	Short Text	4	Kode mekanik	
tgl_pm_terencana	Date/Time	Long Date	Tanggal PM direncanakan	
tgl_pm_aktual	Date/Time	Long Date	Tanggal PM dilaksanakan	
keterangan	Long Text	Long Text	Keterangan tindakan PM	

5.1.2 Desain *User interface*

Setelah melakukan desain *database* fisik, maka tahap selanjutnya adalah perancangan desain *user interface*. *User interface* adalah tampilan dimana *user* akan berkomunikasi atau berhubungan langsung dengan sistem. Perancangan *user interface* yang baik akan mempermudah *user* dalam menjalankan dan mengidentifikasi fungsi serta fasilitas – fasilitas yang terdapat dalam sistem. Maka dari itu sangatlah perlu untuk memperhatikan tata letak maupun kombinasi penggunaan warna dalam perancangan *user interface*. Desain *user interface* ini meliputi hierarki menu, *form* dan *report*.

5.1.2.1 Bagan Hierarki Menu

Hierarki menu adalah urutan susunan menu yang terdapat dalam sistem informasi manajemen penjadwalan PM ini. Urutan tersebut dimulai dari *form login* yang otomatis tampil saat *user* membuka aplikasi. Pada menu ini terdapat sistem *security* yang dapat membedakan hak akses dari pengguna seperti yang terlihat pada gambar 5.2



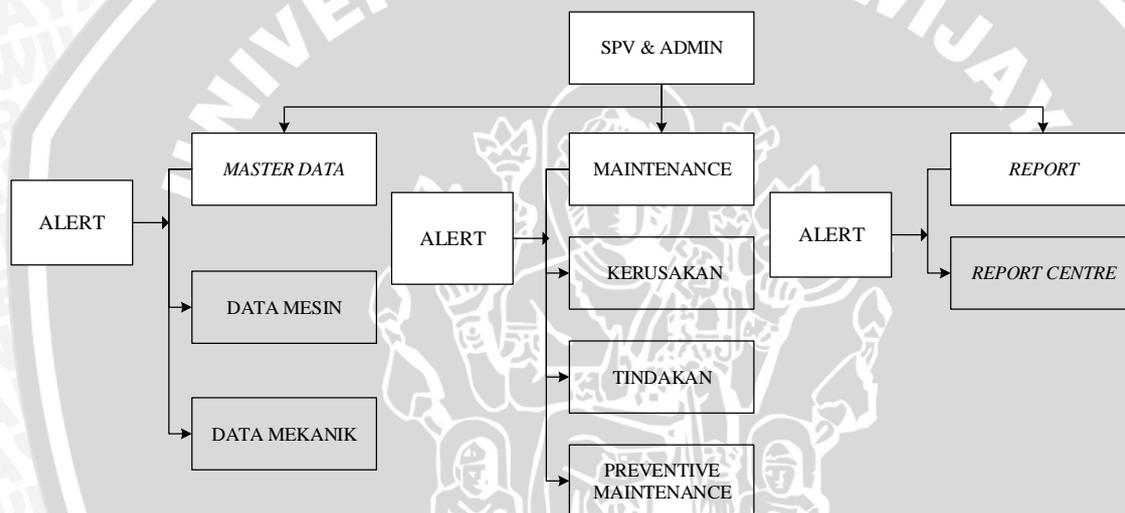
Gambar 5.2 Hierarki menu SIM Penjadwalan PM

Berdasarkan gambar 5.2 dapat dilihat bahwa sistem mengakomodasi dua menu utama, yaitu untuk SPV beserta Admin dan untuk mekanik, yang mana sistem mampu membedakannya bedasar *username* dan *password* yang digunakan. Mekanik hanya mempunyai hak akses untuk melihat jadwal PM beserta master data yang meliputi data mesin, *sparepart* dan mekanik. Sedangkan SPV dan Admin memiliki hak untuk mengakses seluruh fitur yang dimiliki oleh sistem, meliputi tambah atau edit keseluruhan data dan untuk melihat *report*. Berikut hirarki menu yang terdapat pada masing-masing *form*:

1. Form SPV

Form SPV merupakan *form* yang ditujukan untuk SPV, pada *form* ini terdapat menu-menu seperti master data, jadwal perawatan mesin, *report*, *setting*, dan *home*.

Gambar 5.3 berikut merupakan desain hierarki *form* SPV.



Gambar 5.3 Desain hierarki menu SPV dan Admin

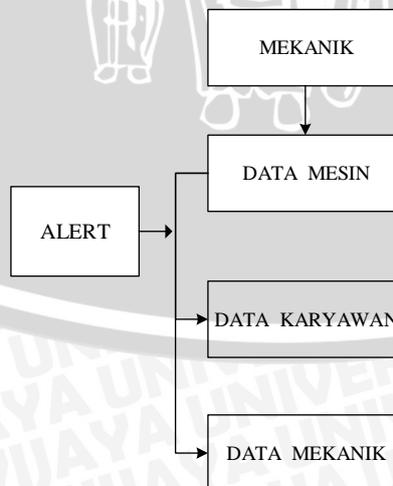
Pada saat SPV maupun admin masuk ke dalam menu *home*, maka akan terlihat menu- menu seperti pada gambar 5.3. berikut merupakan penjelasan untuk masing – masing menu..

a. *Master Data*, merupakan menu untuk melihat data – data induk yang terdiri dari data mesin, *sparepart* dan mekanik. Berikut penjelasan untuk masing – masing menu master data.

- 1) Data Mesin, merupakan submenu untuk melihat dan mengubah data mesin.
- 2) Data *Sparepart*, merupakan submenu untuk melihat dan mengubah data *sparepart*.
- 3) Data Mekanik, merupakan submenu untuk melihat dan mengubah data mekanik

- b. *Maintenance*, merupakan menu utama untuk melihat menu –menu tentang perawatan maupun kerusakan mesin. Dalam menu *maintenance* terdapat submenu sebagai berikut.
- 1) Kerusakan, merupakan submenu untuk menambah data kerusakan maupun untuk melihat *history* kerusakan mesin.
 - 2) Tindakan, merupakan submenu untuk menambah data tindakan maupun untuk melihat *history* tindakan mesin
 - 3) *Preventive Maintenance*, merupakan submenu untuk melihat jadwal *preventive maintenance* yang belum terlaksana atau yang akan datang.
- c. *Report*, merupakan menu yang didalamnya terdapat satu buah menu yaitu *report centre*. Di dalam submenu tersebut SPV dan admin bisa melihat seluruh *report* yang ada, mulai dari *summary*, *report* rutin, *report* dadakan maupun *report* khusus.
- d. *Alert*, merupakan submenu yang terdapat pada setiap menu. Submenu alert ini digambarkan dalam sebuah *icon* tanda seru, dimana submenu ini hanya akan muncul jika ada mesin yang mendekati waktu *preventive maintenance* maupun ada *sparepart* yang jumlah *stocknya* kurang dari atau sama dengan jumlah minimal yang ditentukan.
2. *Form Mekanik*

Form mekanik merupakan *form home* khusus mekanik, dimana mekanik tidak bisa menambah atau merubah data, melainkan hanya bisa melihat isi data. Struktur menu *form home* mekanik akan diperlihatkan pada gambar 5.4 berikut ini.



Gambar 5.4 Desain hierarki menu Karyawan

Berdasarkan Gambar 5.4 dapat dilihat bahwa pada *form* mekanik terdapat menu sebagai berikut ini:

- a. Data Mesin, merupakan submenu untuk melihat data mesin.
- b. Data *Sparepart*, merupakan submenu untuk melihat data *sparepart*.
- c. Data Mekanik, merupakan submenu untuk melihat data mekanik
- d. *Alert*, merupakan submenu yang digambarkan dalam sebuah icon tanda seru, dimana submenu ini hanya akan muncul jika ada mesin yang mendekati waktu *preventive maintenance* maupun ada *sparepart* yang jumlah *stock*nya kurang dari atau sama dengan jumlah minimal yang ditentukan.

5.1.2.2 Desain *User interface Form*

Perancangan desain *user interface form* ditujukan agar *user*, yaitu SPV, admin maupun mekanik mendapatkan kemudahan saat menggunakan aplikasi. Kemudahan yang dimaksud antara lain adalah kemudahan dalam menemukan menu yang dibutuhkan karena submenu submenu sudah digolongkan ke dalam tiga besar menu utama seperti yang sudah dijelaskan pada menu sebelumnya. Selain itu penggunaan tombol command button dan caption yang tidak terlalu kecil juga akan mempermudah untuk menemukan menyang dituju. Penggunaan warna yang dominan adalah warna orange, dikarenakan warna tersebut identik dengan PT XYZ. Berikut merupakan perancangan *form* yang sesuai dengan hierarki menu pada gambar 5.2

1. *Form* SPV dan Admin

Pada *form* ini, menu utama dirancang dengan menggunakan Navigation Control, dimana penggunaannya biasa diterapkan pada website. Pada navigation Control, jika *user* memilih salah satu menu utama, maka submenu – submenu yang bersangkutan akan muncul, dan jika *user* memilih menu utama yang lain maka submenu sebelumnya akan disembunyikan dan diganti dengan submenu menu utama yang aktif tanpa perlu berpindah *form*. Untuk lebih jelasnya akan diterangkan pada Gambar 5.5 sampai 5.7

LOGO PT XYZ

SISTEM INFORMASI MANAJEMEN PREVENTIVE MAINTENANCE

1	2	3
---	---	---

4

5

6

7

Gambar 5.5 Desain *form* SPV & Admin Saat Menu 1 Aktif

Gambar 5.5 memperlihatkan *form home* SPV dan Admin saat menu 1 aktif. Berikut adalah keterangan menu dari gambar 5.5 :

1. *Master Data*
2. *Maintenance*
3. *Report*
4. Alert
5. Data Mesin
6. *Data Sparepart*
7. *Data Mekanik*

LOGO PT XYZ

SISTEM INFORMASI MANAJEMEN PREVENTIVE MAINTENANCE

1	2	3
---	---	---

4

8

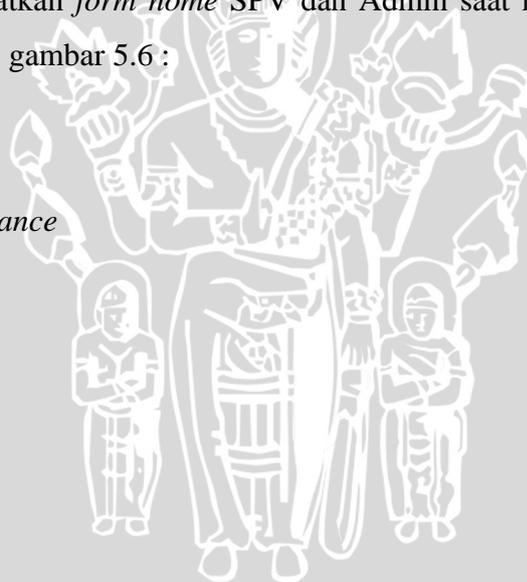
9

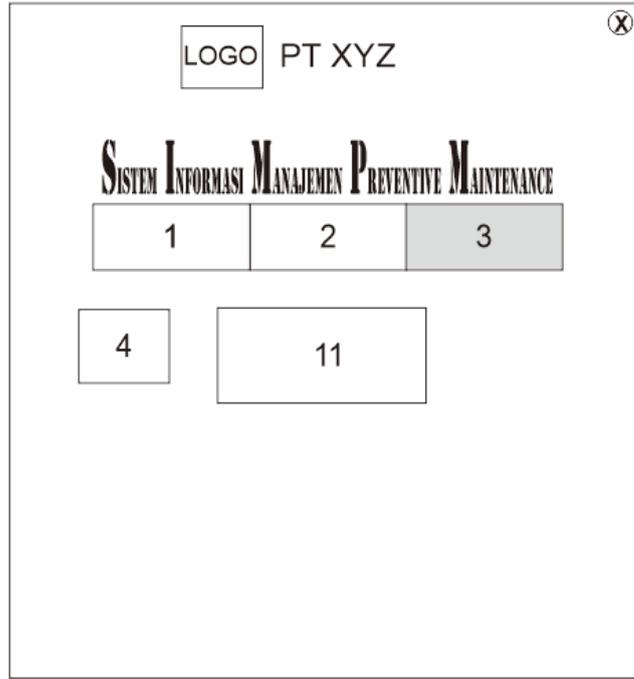
10

Gambar 5.6 Desain *Form Home* SPV & Admin Saat Menu 2 Aktif

Gambar 5.6 memperlihatkan *form home* SPV dan Admin saat menu 2 aktif. Berikut adalah keterangan menu dari gambar 5.6 :

8. *Kerusakan*
9. *Tindakan*
10. *Preventive Maintenance*



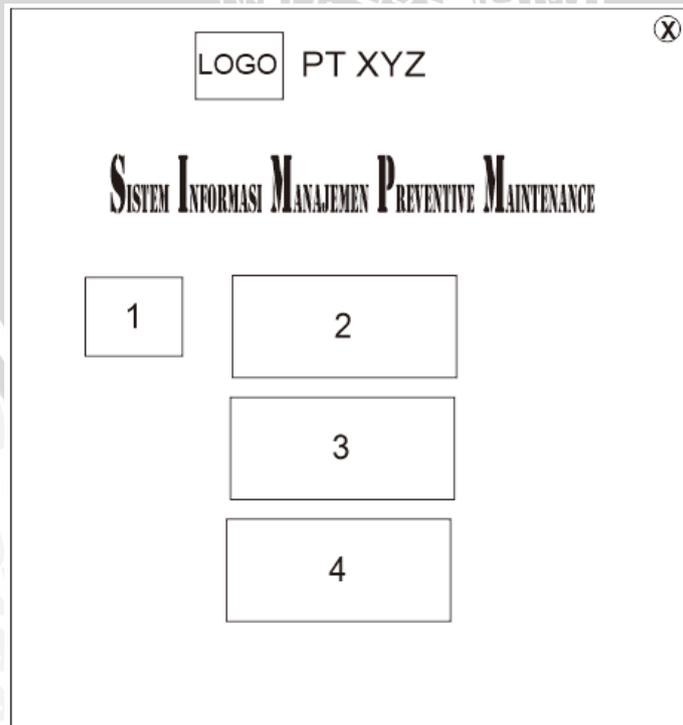


Gambar 5.7 Desain form SPV & Admin Saat Menu 3 Aktif

Gambar 5.7 memperlihatkan form home SPV dan Admin saat menu 3 aktif. Berikut adalah keterangan menu dari gambar 5.7 :

- 11. Report Centre
- 2. Form Mekanik

Berikut adalah form Home Mekanik yang merupakan visual dari rancangan hierarki menu dari gambar 5.8.



Gambar 5.8 Desain form Mekanik

Keterangan:

1. *Alert*
2. Data Mesin
3. Data *Sparepart*
4. Data Mekanik

5.1.2.3 Desain Report

Salah satu fungsi utama dari sistem informasi manajemen adalah untuk menghasilkan *report* untuk SPV dan administrator. *Report* tersebut berguna untuk SPV sebagai bahan pertimbangan pengambilan keputusan. Dalam SIM penjadwalan PM ini ada 4 kategori *report* yaitu *report* rekapan *summary*, *report* rutin, *report* dadakan dan *report* khusus. Berikut adalah penjelasan untuk masing – masing kategori *report*.

1. *Report* Rekapan / *Summary*

Report summary ini berisi rangkuman dari keseluruhan data yang ada di bagian *maintenance* PT XYZ. Ada beberapa *report* yang ada pada kategori ini, antara lain.

- a. *Summary* Mesin, berisi tentang data – data mesin.
- b. *Summary Sparepart*, berisi tentang data – data *sparepart*.
- c. *Summary* Mekanik, berisi tentang data – data mekanik bagian *maintenance*.
- d. *Summary* Tindakan, berisi tentang data – data tindakan terhadap kerusakan mesin.
- e. *Summary* PM, berisi data kegiatan *preventive maintenance* yang telah dilakukan.

2. *Report* Rutin

Report rutin berisi tentang pengolahan data yang umumnya digolongkan berdasarkan satuan waktu, misalnya harian, mingguan, per tiga bulan, maupun tahunan. Data tersebut digunakan untuk memantau kondisi suatu objek dalam kurun waktu tertentu. Dalam *report* rutin ini terdapat dua laporan yaitu

- a. Laporan kerusakan mesin harian
- b. Laporan kerusakan mesin bulanan.

3. *Report* Dadakan

Report dadakan berfungsi untuk mengetahui secara cepat *informasi* yang dibutuhkan, misalnya untuk mengetahui apakah *sparepart* yang pada saat itu dibutuhkan sedang mengalami *stock out* atau tidak. Berikut merupakan laporan yang masuk dalam kategori *report* dadakan.

- a. *Alert* PM, berisi tentang data mesin dan *sparepart* mana saja yang mendekati waktu *preventive maintenance*.

- b. Alert *Sparepart*, berisi tentang data *sparepart* mana saja yang mendekati *stock out* atau yang mendekati batas *stock* minimal.

4. Report Khusus

Report khusus merupakan suatu kategori laporan yang hanya bersifat tambahan atau sebagai pelengkap saja. Misalnya untuk menunjukkan sebuah *trend* kejadian dan sebagainya. Berikut merupakan laporan yang dirancang untuk kategori *report* khusus.

- a. *Chart* Pemakaian *Sparepart*, merupakan laporan yang didalamnya terdapat bagan atau *chart* yang menunjukkan angka total pemakaian *sparepart*. *Report* ini di kembangkan untuk mengetahui tingkat jumlah pemakaian *sparepart*.
- b. *Chart Downtime* Mesin, merupakan laporan yang didalamnya terdapat bagan atau *chart* yang menunjukkan angka *downtime* mesin pada setiap bulan.

5.1.3 Desain Algoritma

Algoritma merupakan serangkaian urutan – urutan langkah logis untuk menyelesaikan suatu masalah yang disusun secara sistematis. Pada tahap desain algoritma ini, membahas tentang tahapan proses untuk menghasilkan *output* yang diperlukan dari sistem informasi yang dinyatakan dalam *pseudocode*. Berikut merupakan *pseudocode* untuk beberapa proses yang terjadi pada Sistem Informasi Manajemen Penjadwalan PM.

1. *Pseudocode* untuk menambah data

Proses penambahan data pada sistem ini digunakan pada data mesin, *sparepart*, mekanik, kerusakan dan data tindakan mesin. Pada gambar 5.9 berikut ini merupakan contoh *pseudocode* tambah data *sparepart*.

```
//menambah dan menyimpan data sparepart mesin
Set [sparepart] sebagai Recordsource
AddNew → perintah untuk menambah record
Input data → kode sparepart, nama sparepart, jumlah stock, jumlah minimum
UPDATE → memperbarui data pada [sparepart]
```

Gambar 5.9 *Pseudocode* proses menambah dan menyimpan data
Pseudocode proses menambah dan menyimpan data diatas merupakan contoh penerapan untuk data *sparepart*. *Pseudocode* tersebut bisa juga diimplementasikan untuk menambah dan menyimpan data yang lain dengan merubah *Recordsource* dan data yang diinputkan.

2. *Pseudocode* untuk menentukan penjadwalan PM

Penjadwalan PM adalah hasil dari penjumlahan tanggal selesai tindakan ditambah dengan interval PM masing – masing mesin dan sparepart. Berikut Tabel 5.10 adalah algoritma penjadwalan PM

```
//penjadwalan PM
Set [interval] sebagai Recordsource
Cocokkan value kode_mesin dan kode_sparepart pada [interval] dengan value
kode_mesin dan kode_sparepart pada form tindakan kerusakan.
Cocok→text box interval terisi dengan nilai interval
tgl_pm.Value = tgl_selesai.Value + interval.Value
```

Gambar 5.10 *Pseudocode* penjadwalan PM

Nilai interval disimpan dalam tabel Interval, agar administrator bisa dengan mudah merubah nilai interval tersebut tanpa harus mempelajari bahasa pemrograman terlebih dahulu. *Pseudocode* ini akan diimplementasikan pada modul program penjadwalan PM.

3. *Pseudocode report*

Report disajikan untuk SPV yang berguna untuk bahan pengambilan keputusan. Sistem informasi manajemen penjadwalan *preventive maintenance* menyajikan berbagai macam *report* seperti yang dibahas pada sub bab sebelumnya. Berikut Gambar 5.11 merupakan contoh *pseudocode* dari beberapa *report* yang tersedia pada sistem.

```
//report tindakan
Pilih data dari [tindakan], [mesin], [sparepart]
Urutkan berdasarkan “tanggal kerusakan” secara Ascending
Tampilkan report tindakan

//report kerusakan mesin bulanan
Pilih data dari [kerusakan], [mesin], [sparepart]
Setting group berdasarkan bulan
Tampilkan report kerusakan mesin bulanan

//report alert pm
Pilih data dari [jadwal pm], [mesin], [sparepart] dengan criteria “check = false “
Urutkan berdasarkan “tanggal_pm” secara Ascending
Tampilkan report alert pm

//report chart downtime mesin
Pilih data dari [qrydowntime]
Setting “nama_mesin” sebagai column heading, “downtime” sebagai value
Golongkan berdasarkan bulan
Tampilkan report chart downtime mesin
```

Gambar 5.11 *Pseudocode* proses *report* dan *alert* jadwal perawatan mesin

Pseudocode proses *report* .ini diimplementasikan bukan dengan bahasa pemrograman VBA melainkan dengan kode *Structure Query Language* (SQL) dalam *Microsoft Acces* 2013.

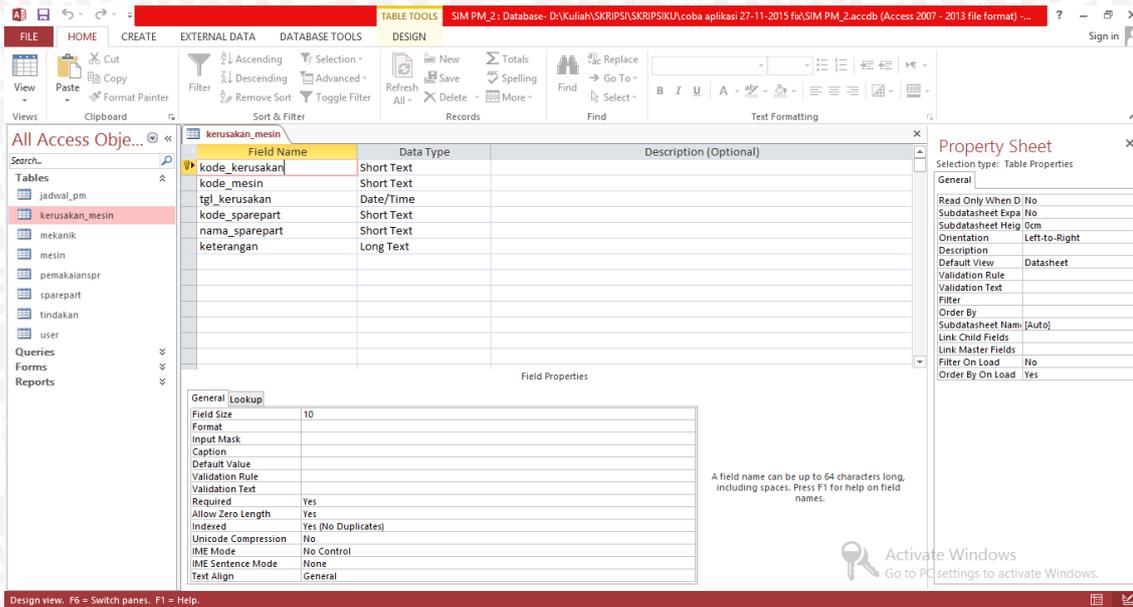
5.2 Implementasi

Tahap implementasi adalah penerapan dari proses perancangan aplikasi dari awal, yaitu mulai desain *database* fisik, desain *database* logis sampai desain algoritma ke dalam *Microsoft Acces* 2013. Tahap implementasi sistem terdiri dari implementasi *database*, implementasi *user interface*, dan implementasi modul program.

5.2.1 Implementasi Database

Pada tahapan implementasi *database* ini dimulai dengan membuat tabel – tabel yang sebelumnya sudah dirancang pada sub bab desain *database* fisik. Tabel – tabel inilah yang nantinya akan menampung seluruh informasi dari sistem informasi manajemen penjadwalan *preventive maintenance*. Selanjutnya akan dibentuk relasi untuk masing – masing tabel agar saling terhubung satu sama lain. Proses implementasi *database* dengan *Microsoft Acces* 2013 selengkapnya dilakukan dengan langkah sebagai berikut.

1. Membuka *Microsoft Access* 2013 dan pilih *Blank desktop database*.
2. Selanjutnya memilih *Create* → *Table Design*
3. Mengisi *field name* dengan judul kolom dan *data type* dengan tipe data pada kolom tersebut.
4. Menentukan *Primary Key* untuk setiap tabel.
5. Memilih *Save* dan memberi nama tabel.
6. *Tabel* sudah terbentuk.



Gambar 5.12 Tampilan *Table Design* pada *Microsoft Access 2013*

Gambar 5.12 merupakan salah satu *table* atau entitas dalam sistem, yaitu tabel kerusakan mesin. Isi beserta pengaturan dalam *Field Name* dan *Data Type* harus disesuaikan pada saat perancangan *database* fisik. Jika semua *Field* yang diperlukan sudah dibuat, maka akan menghasilkan sebuah *table* yang siap digunakan untuk menginputkan data ke dalam *database*, seperti yang terlihat pada gambar 5.13

1. Tabel Kerusakan Mesin

Desain *database* fisik tabel 5.6, diimplementasikan menjadi tabel kerusakan mesin. Berikut Gambar 5.13 yang merupakan implementasi tabel *kerusakan_mesin* untuk PT XYZ.

kode_kerus	kode_mesin	tgl_kerusakan	kode_spare	nama_sparepart	keterangan	Click to Add
R0100	PPA3	15 Oktober 2015	S005	Gripper	kabel putus	
R0101	PPA4	20 Oktober 2014	S006	Bearing 6004	pecah	
R0102	PPA3	03 November 2014	S009	KQL M5	rusak	
R0103	PPA3	06 November 2014	S003	Proximity Sensor		
R0105	PPA4	17 November 2014	S008	Y Axis		
R0108	PPA4	20 Januari 2015	S008	Y Axis		
R0109	PPA3	23 Januari 2015	S002	KQH 08-00	rusak	
R0110	PPA3	24 November 2015	S003	Proximity Sensor	kabel putus	
R0111	PPA4	23 September 2015	S001	KQH 06-00	rusak	
R0112	PPA4	29 November 2015	S008	Y Axis	macet	

Gambar 5.13 *Printscreen* Tabel Kerusakan Mesin pada *Microsoft Access 2013*

Implementasi tabel kerusakan mesin pada Gambar 5.13 berdasarkan dengan desain *database* fisik entitas kerusakan mesin pada Tabel 5.6. Pada tabel ini terdapat atribut *kode_kerusakan*, *kode_mesin*, *tgl_kerusakan*, *kode_sparepart*, *nama_sparepart*, dan *keterangan*.

2. Tabel Mesin

Desain *database* fisik Tabel 5.3, diimplementasikan menjadi tabel mesin. Berikut Gambar 5.14 yang merupakan implementasi tabel mesin untuk PT XYZ.

kode_mesin	nama_mesin	tahun_pem	type	kekuatan	Click to Add
PPA3	Pounding Auto 3	2008	Yang Li	63 Ton	
PPA4	Pounding Auto 4	2009	yang Li	63 Ton	
*		0			

Gambar 5.14 *Printscreen* Tabel Mesin pada *Microsoft Access* 2013

Implementasi tabel mesin pada Gambar 5.14 berdasarkan desain *database* fisik entitas mesin pada Tabel 5.3. Pada tabel ini terdapat atribut *kode_mesin*, *nama_mesin*, *tahun_pembuatan*, *type*, dan *kekuatan*.

3. Tabel Jadwal PM

Desain *database* fisik tabel 5.8, diimplementasikan menjadi tabel jadwal PM. Berikut Gambar 5.15 yang merupakan implementasi tabel jadwal PM untuk PT XYZ.

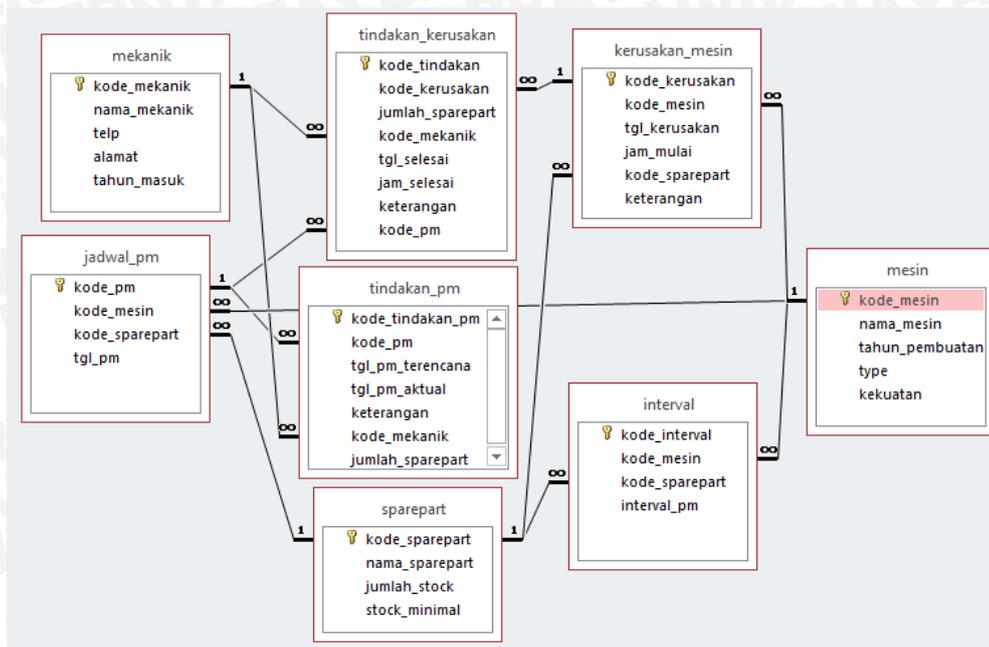
kode_pm	kode_sparepart	kode_mesin	tgl_pm
12	S003	PPA3	06 November 2014
7	S002	PPA3	03 Mei 2015
10	S001	PPA4	23 September 2015
2	S002	PPA3	14 November 2015
9	S003	PPA3	24 November 2015
6	S004	PPA3	26 November 2015
11	S008	PPA4	29 November 2015
3	S005	PPA3	11 Februari 2016
4	S008	PPA4	09 Maret 2016
*	(New)		

Gambar 5.15 *Printscreen* Tabel Jadwal PM pada *Microsoft Access* 2013

Implementasi tabel mesin pada Gambar 5.15 berdasarkan desain *database* fisik entitas jadwal PM pada Tabel 5.8. Pada tabel ini terdapat atribut *kode_pm*, *kode_sparepart*, *kode_mesin*, dan *tgl_pm*.

4. Relasi Antar Tabel

Fasilitas *Relationships* pada *Microsoft Acces* 2013 berfungsi untuk saling menghubungkan antara tabel satu dengan yang lainnya. Pada gambar 5.16 berikut ini adalah relasi yang terbentuk dari sistem informasi manajemen penjadwalan *preventive maintenace*.



Gambar 5.16 *Printscreen* Relasi Antar Tabel Entitas

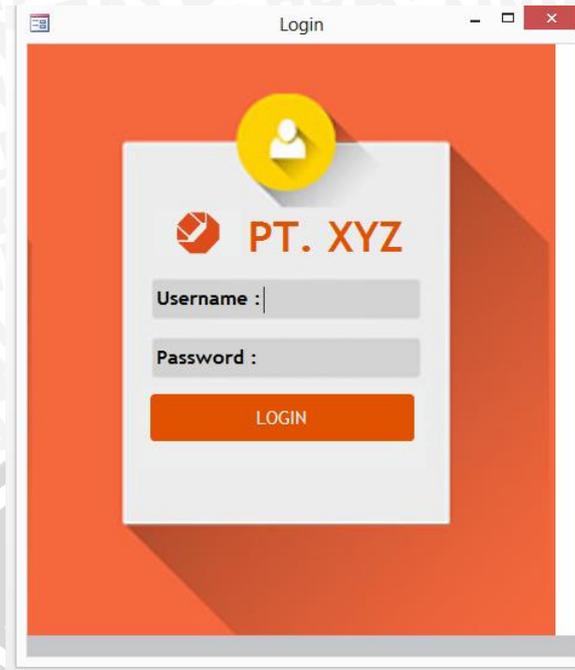
Relasi yang terbentuk pada gambar 5.16 sesuai dengan desain *database* fisik yaitu pada *Entity Relational Diagrams* (ERD). Relasi ini menghubungkan antar tabel melalui *Primary* dan *foreign key*.

5.2.2 Implementasi *User interface*

Implementasi *user interface* ini ditujukan agar *user* dapat dengan mudah menggunakan aplikasi serta mendapatkan informasi yang dibutuhkan. Implementasi *user interace* berdasarkan tahap desain pada sub bab sebelumnya. Berikut adalah implementasi *user interface* sistem dengan menggunakan *Microsoft Acces 2013*.

1. *Form Login*

Form login adalah tampilan pertama yang muncul saat aplikasi di jalankan. Di *form login* ini *user* bisa memasukkan *username* dan *password* sesuai jabatan di bagian *maintenance*. Berikut merupakan implementasi *form login* yang ditunjukkan pada gambar 5.17



Gambar 5.17 *Printscreen Form Login*

2. *Form Home*

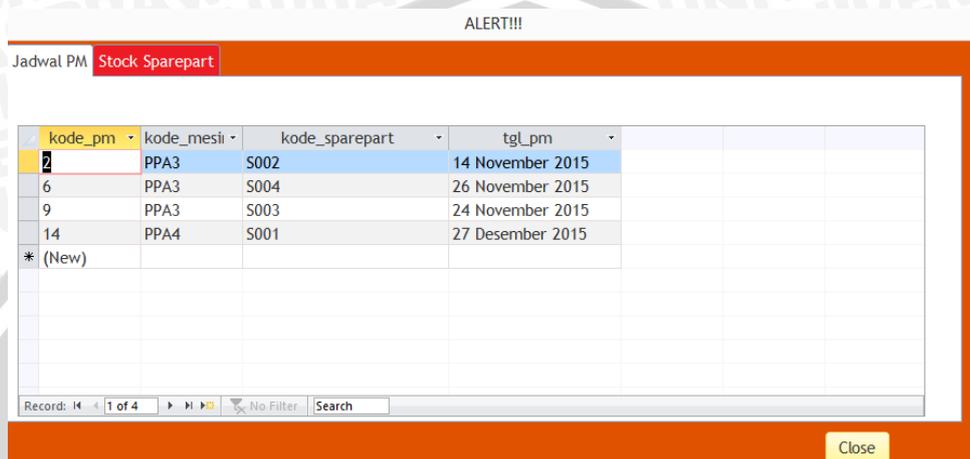
Form home adalah *form* dimana *user* bisa mengakses menu utama dan submenu yang terdapat pada sistem. Khusus untuk SPV dan administrator bisa membuka semua fitur yang dimiliki SIM penjadwalan *preventive maintenance*, mulai dari menu master data sampai menu *report*. Gambar 5.18 berikut ini adalah tampilan dari menu *home* SPV dan Administrator.



Gambar 5.18 *Prinscreen Form Home SPV &Admin*

3. Form Alert

Setelah *user login*, *form alert* atau peringatan akan muncul secara otomatis jika ada mesin yang mendekati dan melebihi jadwal *preventive maintenance* atau ada *sparepart* yang mengalami *stock out* maupun mencapai batas *stock* minimal. Sebaliknya *form alert* tidak akan muncul apabila hal – hal tersebut tidak terjadi. Gambar 5.19 berikut adalah tampilan dari *form alert*.



kode_pm	kode_mesin	kode_sparepart	tgL_pm
2	PPA3	S002	14 November 2015
6	PPA3	S004	26 November 2015
9	PPA3	S003	24 November 2015
14	PPA4	S001	27 Desember 2015
* (New)			

Gambar 5.19 Prinscreen Form Alert

4. Report

Report adalah salah satu *output* yang dihasilkan oleh sistem informasi manajemen. *Report* ini sangatlah penting untuk SPV sebagai bahan pengambilan keputusan. Dalam SIM ini, menu *report* dikumpulkan dalam satu *form* yaitu *form Report Centre* agar lebih mudah di akses oleh *user*. Berikut Gambar 5.20 adalah tampilan dari *form Report Centre*.



Gambar 5.20 Printscreen Form Report Centre

a. *Report Kerusakan Mesin Bulanan*

Report kerusakan mesin bulanan ini dibuat agar SPV lebih mudah memantau kondisi mesin pada setiap bulannya, misalnya mesin mana yang sering mengalami kerusakan pada bulan tertentu. Berikut Gambar 5.21 adalah tampilan dari *report* kerusakan mesin harian.

Bulan	Mesin	Nama Sparepart	Tanggal Kerusakan	Keterangan
Oktober 2014				
	PPA4	Bearing 6004	20 Oktober 2014	pecah
				Total Kerusakan Per Bulan : 1 Kali
November 2014				
	PPA3	KQL M5	03 November 2014	rusak
	PPA3	Proximity Sensor	06 November 2014	
				Total Kerusakan Per Bulan : 2 Kali
	PPA4	Y Axis	17 November 2014	
				Total Kerusakan Per Bulan : 1 Kali
Januari 2015				
	PPA3	KQH 08-00	23 Januari 2015	rusak
				Total Kerusakan Per Bulan : 1 Kali

Gambar 5.21 *Printscreen Report* Kerusakan Mesin Bulanan

b. *Report Jadwal PM*

Pada *report* jadwal PM, admin maupun SPV dapat dengan cepat untuk mengetahui mesin mana saja yang sudah mendekati waktu *preventive maintenance*. Gambar 5.22 berikut adalah tampilan dari *report* jadwal PM.

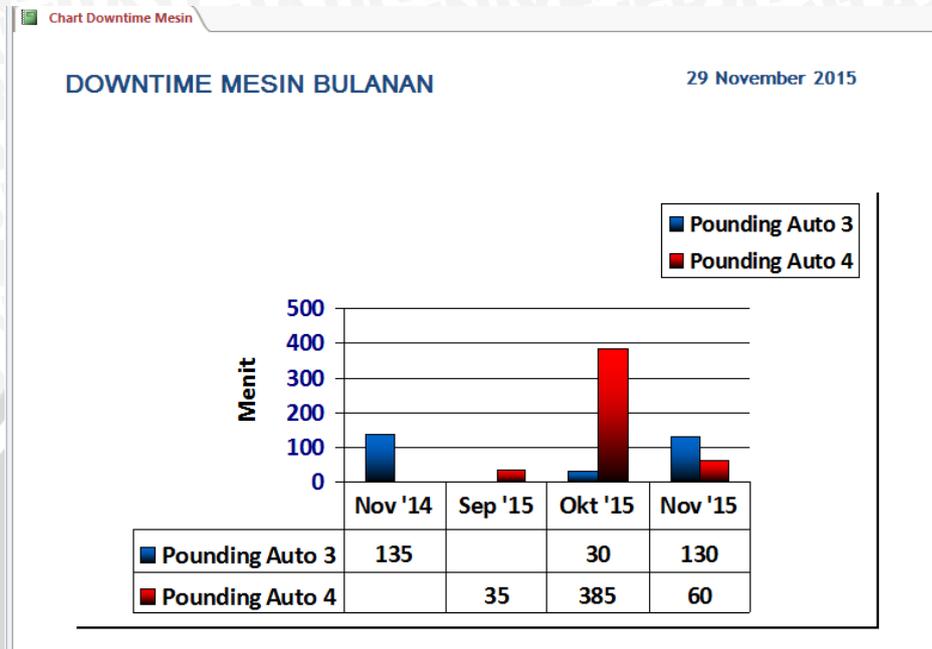
Tanggal PM	Kode Sparepart	Nama Sparepart	Kode Mesin	Nama Mesin	Check
26 November 2015	S004	Servo	PPA3	Pounding Auto 3	<input type="checkbox"/>
29 November 2015	S008	Y Axis	PPA4	Pounding Auto 4	<input type="checkbox"/>

Page 1 of 1

Gambar 5.22 *Printscreen Report* Jadwal PM

c. *Report Chart Downtime Mesin*

Pada *report* ini terdapat sebuah *chart* agar SPV lebih mudah untuk membandingkan lama *downtime* mesin Pounding yang disajikan berdasarkan siklus bulanan. Gambar 5.23 berikut adalah tampilan dari *chart downtime* mesin.



Gambar 5.23 *Printscreen Report Chart Downtime Mesin*

5.2.3 Implementasi Modul Program

Berikut contoh beberapa penggalan *source code* sesuai dengan desain algoritma proses yang telah disusun sebelumnya menggunakan VBA *with Microsoft Access 2013*.

1. Memasukkan dan menyimpan data

Pada Gambar 5.24 berikut merupakan implementasi modul pemrograman untuk menambah dan menyimpan data. Berikut ini merupakan contoh implementasi yang diterapkan untuk menambah dan menyimpan data *sparepart*.

```

Private Sub cmdsave_Click() 'untuk menyimpan data
Me.RecordSource = "sparepart"
Set rs = Me.RecordsetClone
rs.AddNew
rs(0) = kode_sparepart.Value
rs(1) = nama_sparepart.Value
rs(2) = jumlah_stock.Value
rs(3) = stock_minimal.Value
rs.Update
MsgBox "Data anda telah tersimpan!", vbInformation
DoCmd.close acForm, "addsparepart", acSaveYes
End Sub

Private Sub Form_Load() 'untuk membuat pengkodean otomatis pada
sparepart
Dim kode As String
Dim kd As String
Me.kode_sparepart = kd
Set RS3 = Me.RecordsetClone
If Not RS3.EOF Then
RS3.MoveLast
kd = Val (Right (RS3(0), 2)) + 1

Select Case Len(kd)
Case 1
kode_sparepart.Value = "S00" + Trim(Str(kd))
Case 2
kode_sparepart.Value = "S0" + Trim(Str(kd))
End Select
Else
kode_sparepart.Value = "S001"
End If
End Sub

```

Gambar 5.24 Syntax proses memasukkan dan menyimpan data *sparepart*

2. Proses Penentuan Jadwal PM

Proses penentuan Jadwal PM dilakukan dengan beberapa tahap, yaitu perhitungan MTTF, perhitungan interval PM dan selanjutnya memasukkan hasil perhitungan interva PM ke dalam *source code*.

- a. Berikut Tabel. 5.10 adalah contoh perhitungan MTTF untuk *sparepart gripper* mesin PPA4.

Tabel 5.10 MTTF dan MTTR Kerusakan *Gripper* Mesin PPA4

No	Mulai		Selesai		TTF (dalam jam)	TTR (dalam menit)
	Tanggal	Jam	Tanggal	Jam		
1	13/07/2013	14:30	13/07/2013	19:30	0	300
2	02/09/2013	06:20	02/09/2013	08:55	1215,83	155
3	19/12/2013	23:00	19/12/2013	01:40	2608,67	160
4	04/03/2014	23:30	05/03/2014	03:00	1800,50	210
5	05/06/2014	16:10	05/06/2014	17:30	2224,67	80
Total					7849,67	905

Selanjutnya dari tabel 5.9 akan dilakukan perhitungan MTTF dan MTTR *gripper* mesin *Pounding automatic* 4. Perhitungan tersebut menggunakan persamaan 2-1 sebagai berikut berikut.

$$\text{Mean Time To Failure} = \frac{\text{Total Time To Failure}}{\text{Number of Failure}} = \frac{7849,67}{5} = 1569,93 \text{ Jam}$$

$$\text{Mean Time To Repair} = \frac{\text{Total Time To Repair}}{\text{Number of Repair}} = \frac{905}{5} = 181 \text{ Menit}$$

Untuk perhitungan Nilai MTTF dan MTTR yang lain ditampilkan pada Lampiran 3 dan 4.

b. Interval PM

Untuk mempermudah otomatisasi penjadwalan, maka interval *preventive maintenance* akan dinyatakan dalam hari. Untuk itu, hasil perhitungan MTTF yang berupa jam akan dirubah ke dalam hari. Berikut Tabel 5.11 merupakan interval perawatan *sparepart* untuk mesin PPA4.

Tabel. 5.11 Interval *Preventive maintenance* Mesin PPA4

No	<i>Sparepart</i>	MTTF (Jam)	Interval PM (Hari)
1	KQH 06-00	674,03	28
2	Gripper	1569,93	65
3	Servo	1481,37	61
4	Proximity Sensor	1812,18	75
5	KQH 08-00	1523,43	63
6	Bearing 6004	1100,17	45
7	Y axis	2214,55	92
8	KQL M5	1048,63	43

Untuk perhitunag interval PM mesin selanjutnya ditampilkan pada Lampiran 5

c. *Source Code*

Selanjutnya nilai interval yang diperoleh pada tahap sebelumnya akan dimasukkan ke dalam *source code*. Pada Gambar 5.25 berikut ini merupakan *source code* untuk menentukan penjadwalan *peventive maintenance* secara otomatis yang dilakukan pada *form* tindakan.

```
Private Sub tgl_selesai_AfterUpdate()
Me.RecordSource = "interval"
Set rs1 = Me.RecordsetClone
If cmbkerusakan.Column(1) <> 0 And cmbkerusakan.Column(2) <> 0 Then
If Not rs1.EOF Then
rs1.MoveFirst
rs1.FindFirst " kode_mesin = '" & cmbkerusakan.Column(1) & _
'" And kode_sparepart = '" & cmbkerusakan.Column(2) & "'" _
rs1.Edit
interval.Value = rs1(3)
tgl_pm.Value = tgl_selesai.Value + interval.Value
rs1.Update
Me.Refresh
End If
End If
End Sub
```

Gambar 5.25 *Syntax* Proses Penjadwalan PM

Setiap mesin dan sparepart yang sudah mempunyai jadwal PM, maka akan ditindak lanjuti dengan tindakan PM. Dimana untuk kegiatan tersebut, sistem sudah menyediakan form tindakan PM. Dalam form tersebut, akan dicatat tanggal PM yang direncanakan serta tanggal PM aktual agar user bisa memantau tingkat efektifitas dari sistem.

3. Menampilkan *Alert* Jadwal PM Dan *Stock Sparepart*

From *Alert* hanya akan muncul jika terdapat data mesin yang mendekati waktu PM dan terdapat data *sparepart* yang jumlah *stock* mendekati jumlah *stock* minimal. Untuk itu dibutuhkan source kode seperti pada Gambar 5.26

```
Sub alert()
Dim terlambat As Integer
terlambat = DCount("[kode_pm]", "[qryalertpm]" Or DCount("[kode_sparepart]",
"[qryalertspr]")
If terlambat = 0 Then
DoCmd.OpenForm "home", acNormal
Exit Sub
Else
If MsgBox("Terdapat mesin atau sparepart yang memerlukan perhatian anda.
Apakah anda ingin mengetahuinya?", vbYesNo, "ALERT!!...") = vbYes Then
DoCmd.OpenForm "frmalert", acNormal
Else
DoCmd.OpenForm "home", acNormal
End If
End If
End Sub
```

Gambar 5.26 *Syntax* Untuk Menampilkan *Form Alert*

5.3 Pengujian

Tahap pengujian adalah tahapan terakhir dari pengembangan *prototype*. Tujuannya adalah untuk mengetahui apakah sistem sudah berjalan sesuai harapan atau tidak. Tahap pengujian ini ditinjau dari tiga segi, yaitu uji verifikasi, uji validasi dan uji *prototype* yang masing-masing terdapat tujuan yang saling terhubung.

5.3.1 Verifikasi

Pada tahap ini akan menguji apakah *prototype* sudah berjalan seperti tahap perencanaan. Uji verifikasi ini dilakukan dengan cara membandingkan desain *database*, desain *user interface* an desain modul pemrograman yang dilakukan pada tahap desain dengan tahap implementasi. Perbandingan model *database* yang dirancang pada Tabel 5.25 sampai 5.30 dengan implementasi pengembangan *database* pada Gambar 5.12 sampai 5.15 sudah sesuai. Penggunaan *pseudocode* dan ditunjukkan pada Gambar 5.9 sampai 5.11 sudah sesuai dengan implementasi modul program seperti yang ditunjukkan pada gambar 5.23 sampai 5.26. Perbandingan untuk rancangan hierarki menu Seperti yang ditunjukkan pada Gambar 5.2 sudah sesuai dengan implementasi *user interface* pada Gambar 5.17 yaitu

Form Login. Dimana pada *Form Login* tersebut *user* harus memasukkan *username* dan *password* yang akan membawa *user* menuju *form* yang berbeda. *User SPV* dan *Administrator* akan dibawa menuju *form* pada Gambar 5.18 yang sesuai dengan hierarki menu pada Gambar 5.5 sampai 5.7. Begitu juga untuk *User mekanik* juga akan ditunjukkan pada *form home* mekanik.

Pengujian terhadap ketelitian program saat menambah dan menyimpan data. Sebagai contoh dilakukan penambahan data kerusakan dari *Form Kerusakan*. Seperti yang tampak pada Gambar 5.27 berikut ini.

Gambar 5.27 Input Kerusakan Mesin

Setelah semua *field* terisi dan *user* menekan tombol *save* maka data kerusakan tersebut secara otomatis akan tersimpan pada tabel kerusakan. Hasil input tersebut bisa dilihat pada Gambar 5.28 berikut ini.

kode_kerusakan	kode_mesin	tgl_kerusakan	kode_sparepart	nama_sparepart	keterangan
R0113	PPA4	29 November 2015	S001	KQH 06-00	sparepart bocor

Gambar 5.28 Penambahan dan Penyimpanan Data Kerusakan

Berdasarkan Gambar 5.28 data yang diinputkan dari *form* kerusakan sudah tercatat semua ke dalam tabel tujuan yaitu tabel kerusakan. Hal ini menunjukkan bahwa sistem sudah melakukan pekerjaannya sesuai dengan harapan *user*.

Selanjutnya akan dilakukan pengujian ketelitian untuk penjadwalan *preventive maintenance* untuk mengetahui apakah *syntax* dari gambar 5.25 sudah berjalan sesuai

harapan. Data awal akan di input dari *form* tindakan. Untuk data yang diinputkan bisa dilihat pada Gambar 5.29

Gambar 5.29 Input Tindakan Mesin

Gambar 5.29 merupakan input data tindakan yang juga menghasilkan output data tindakan, perubahan pada jumlah *stock sparepart* pada tabel *sparepart*, dan juga menambah jadwal *preventive maintenance* pada mesin serta *sparepart* yang diinputkan. Jadwal *preventive maintenance* tersebut berasal dari *field* tanggal selesai ditambah dengan nilai interval PM (hari) pada tabel 5.21 dan 5.22. Nilai interval PM berbeda beda tergantung nama mesin dan *sparepart* yang diinputkan pada *field* Kode Mesin dan *field* Kode *Sparepart*. Gambar 5.30 berikut adalah jadwal *preventive maintenance* yang dihasilkan.

kode_pm	kode_sparepart	kode_mesin	tgl_pm	check	Click to Add
14	S001	PPA4	27 Desember 2015	<input type="checkbox"/>	

Gambar 5.30 Penambahan Jadwal PM

Seperti yang dapat dilihat pada Gambar 5.30, semua *field* dalam tabel *jadwal_pm* sudah mengisi *field – field* yang bersangkutan. Nilai pada *field* *tgl_pm* merupakan hasil dari penjumlahan *field* *Tgl_selesai* pada *form* tindakan kerusakan ditambah nilai interval PM untuk mesin dan *sparepart* bersangkutan yang sudah dihitung dari nilai MTTF pada sub bab sebelumnya. Berikut perhitungan selengkapnya

$$\text{Tgl_pm} = \text{tgl_selesai} + \text{Interval PM sparepart terkait}$$

$$\text{Tgl_pm} = 29 \text{ November } 2015 + 28 \text{ hari}$$

$$\text{Tgl_pm} = 27 \text{ Desember } 2015$$

Hasil perhitungan 27 Desember 2015 tersebut sudah sesuai dengan yang ditampilkan pada Gambar 5.30 yang menandakan sistem penjadwalan sudah berjalan dengan benar.

5.3.2 Validasi

Uji validasi dilakukan dengan cara menguji coba aplikasi sistem informasi penjadwalan *preventive maintenance* secara langsung oleh seluruh *user*, yang meliputi SPV, administrator dan mekanik bagian mesin PT XYZ. Tujuannya adalah untuk menguji apakah fungsi *prototype* yang telah dihasilkan sudah merepresentasikan tujuan awal dari perancangan sistem yaitu untuk mengelola data – data dan menjadi sumber informasi di bagian mesin PT XYZ berdasarkan spesifikasi kebutuhan pada SRC. Berikut Tabel 5.12 adalah hasil perbandingan hasil *prototype* sistem dengan SRC.

Tabel 5.12 Uji Validasi Sistem Informasi Manajemen Perawatan Mesin

Pengguna	Kebutuhan pengguna yang dipenuhi
SPV dan Administrator	Administrator dapat memasukkan dan merubah data – data yang ada pada bagian <i>maintenance</i> seperti data mesin, <i>sparepart</i> , mekanik dan kerusakan.
	Sistem mampu memberikan laporan / <i>report</i> tentang data mesin, <i>sparepart</i> , jadwal perawatan, dan kerusakan mesin.
	Sistem mampu menjadwalkan <i>preventive maintenance</i> berdasarkan perhitungan nilai MTTF.
	Sistem mampu memberikan peringatan kepada SPV dan Admin jika ada mesin yang mendekati jadwal <i>preventive maintenance</i> .
	Sistem mampu memberikan peringatan jika ada <i>sparepart</i> yang mencapai jumlah <i>stock</i> minimal.
Mekanik	Sistem mampu menampilkan jadwal perawatan mesin kepada mekanik.
	Sistem bisa memberikan informasi jumlah <i>stock sparepart</i> kepada mekanik.
	Sistem mampu menampilkan data – data yang berkaitan dengan bagian <i>maintenance</i> , seperti data mesin, <i>sparepart</i> dan mekanik

Berdasarkan hasil uji validasi pada Tabel 5.12, *prototype* sistem informasi manajemen penjadwalan PM sudah mampu memenuhi kebutuhan pengguna seperti yang dijabarkan pada SRC. Sistem informasi memberikan fasilitas kepada administrator untuk mengelola data – data yang ada pada bagian mesin. SPV memperoleh laporan – laporan yang dibutuhkan untuk bahan pertimbangan pengambilan keputusan, seperti laporan kerusakan harian maupun laporan mesin yang mendekati jadwal PM. Sedangkan mekanik bisa melihat informasi yang berkaitan dengan jadwal PM dan *stock sparepart*.

5.3.3 Uji Prototype

Uji *prototype* bertujuan untuk mengetahui apakah *prototype* dapat mengatasi masalah dan kelemahan sistem perawatan mesin yang telah dipaparkan pada tabel PIECES di Bab I (Latar Belakang). Berikut Tabel 5.13 merupakan perbandingan antara sistem lama dan sistem baru.

Tabel 5.13 Perbandingan Performa Antara Sistem Lama dan Sistem Baru

Jenis analisis	Sisem Lama	Sistem Baru
<i>Performance</i>	Kegiatan penjadwalan perawatan mesin dan pengelolaan data <i>sparepart</i> masih dilakukan secara manual sehingga memerlukan waktu lama untuk melakukannya.	Sudah menggunakan sistem informasi berbasis <i>database</i> untuk mengolah seluruh data yang ada pada bagian mesin
	Metode penentuan penjadwalan saat ini masih belum dilakukan dengan perhitungan secara teoritis.	Penjadwalan PM didasarkan atas perhitungan <i>Mean Time To Failure</i> (MTTF).
	Beberapa kali ada kejadian dimana bagian <i>maintenance</i> mengalami <i>stock out</i> pada <i>sparepart</i> yang sedang dibutuhkan untuk mengganti <i>sparepart</i> yang rusak	Sistem mampu memberikan kontrol untuk terus memantau jumlah <i>stock sparepart</i> yang tersedia setiap harinya.
<i>Information</i>	Pencatatan secara manual membuat alur informasi tidak efektif. Jika admin, kabag atau teknisi ingin mengakses data hitoris kerusakan mesin dan sebagainya, harus mencari cari dahulu di buku catatan atau <i>spreadsheet</i> yang bersangkutan	Sistem sudah dilengkapi dengan berbagai macam <i>report</i> , termasuk <i>report</i> kerusakan mesin baik harian maupun bulanan, yang bisa dengan mudah di akses oleh SPV melalui menu <i>report centre</i> .
	Informasi yang dicatat secara <i>text book</i> memiliki kemungkinan lebih besar untuk hilang atau rusak.	Sistem penyimpanan digital berbasis <i>database</i> dapat menyimpan data selamanya.
<i>Economy</i>	Kegiatan produksi bisa tertunda sehari – hari, pada saat terjadi <i>stock out</i> pada <i>sparepart</i> yang sedang dibutuhkan untuk memperbaiki mesin, untuk menunggu pembelian <i>sparepart</i> yang bersangkutan	Telat membeli <i>stock sparepart</i> karena lupa kalau <i>stock</i> sudah minimal tidak akan terjadi karena sistem peringatan mampu memberi peringatan secara terus menerus selama aplikasi diaktifkan.
<i>Control</i>	Belum menerapkan sistem yang bisa memberikan peringatan jumlah <i>sparepart</i> untuk berjaga –jaga jika sewaktu – waktu <i>sparepart</i> yang bersangkutan dibutuhkan dengan segera	Sistem informasi ini dilengkapi dengan sistem alert yang dapat memperingatkan SPV dan administrator jika ada <i>sparepart</i> yang mendekati jumlah <i>stock</i> minimal
	Pada saat ini, semua informasi di bagian <i>maintenance</i> bisa diakses siapa saja karena belum memiliki sistem pengamanan.	Sudah menerapkan sistem <i>security</i> yang akan membatasi hak akses pengguna, terutama untuk pengguna selain SPV dan administrator.
<i>Efficiency</i>	Terjadi penumpukan catatan kertas di bagian <i>maintenance</i> .	Penyimpanan data sudah menggunakan fasilitas digital.
	Pengolahan data secara manual membutuhkan waktu yang lebih lama serta tenaga yang lebih banyak	Hanya membutuhkan satu orang saja sebagai untuk menjalankan aplikasi yaitu administrator.
<i>Services</i>	Pelayanan informasi dari bagian <i>maintenance</i> berlangsung lama karena pencatatan manual yang belum terintegrasi.	Pelayanan informasi bisa disajikan secara cepat dengan sistem berbasis <i>database</i> yang terintegrasi.

Berdasarkan Tabel 5.13 diatas dapat diketahui sistem baru sudah mampu mengatasi kelemahan sistem lama dilihat dari sisi *performance*, *information*, *economi*, *control*, *efficiency* dan *sevices*. Sistem mampu mengatasi masalah berkaitan dengan penjadwalan *preventive maintenance* yang sebelumnya dijadwalkan secara manual, namun sekarang sudah dijadwalkan secara otomatis oleh sistem berdasarkan penghitungan MTTF. Penyimpanan berbasis *database* membuat pengelolaan data menjadi terorganisir, bisa menghindari pencatatan berganda yang tidak efisien baik dari segi waktu maupun dari segi tenaga. Kebutuhan informasi secara cepat bukan lagi hal yang tidak mungkin, karena dalam sistem sudah disediakan berbagai macam *report* yang berisi seluruh informasi berkaitan dengan bagian *maintenance* yang bisa diakses dengan mudah dan cepat.



UNIVERSITAS BRAWIJAYA

(Halaman ini sengaja dikosongkan)

