

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Terdapat lima langkah dalam melakukan analisis suatu permasalahan untuk dijadikan sebuah penelitian. Proses tersebut dimulai dari langkah dalam pengumpulan informasi hingga menjadikan sebuah hasil penelitian. Langkah pengumpulan informasi tersebut diantaranya adalah :

1. Proses identifikasi permasalahan
2. *User requirement*
3. *Software requirement*
4. *Data requirement*
5. *Nonfunctional requirement*

3.1.1 Identifikasi Permasalahan

Pada proses ini penulis melakukan identifikasi permasalahan yang ada pada PT BJTI. Proses awal yang dilakukan adalah menentukan pada bagian manakah yang dijadikan bahan penelitian. Hal tersebut dapat dilakukan dengan cara tanya jawab singkat dengan pihak perusahaan.

Proses awal identifikasi permasalahan pada perusahaan dimulai dari meneliti, apakah setiap proses yang ada pada bagian tersebut sudah sesuai dengan prosedur yang ada. Tanya jawab kepada pihak *staff* perusahaan untuk mendapatkan informasi yang dibutuhkan. Permasalahan bisa diidentifikasi juga dengan adanya temuan-temuan, seperti proses pada bagian tertentu sangat lambat atau bahkan proses pada bagian tertentu sering mengalami kesalahan pada sistem.

Hal-hal tersebut di atas yang kemudian dapat diangkat menjadi bahan penelitian penulis. Untuk dapat mengetahui lebih jelas permasalahan seperti di atas dapat dilakukan dengan menggunakan beberapa langkah sebagai berikut

A. Wawancara

Proses wawancara dikoordinasikan oleh pihak SDM, kemudian pewawancara dipertemukan dengan narasumber utama yang akan menggunakan aplikasi, yaitu bagian mekanik dan keuangan. Proses ini juga membutuhkan narasumber lainnya, seperti narasumber dari divisi IT untuk diberikan pertanyaan mengenai sistem seperti apa yang telah digunakan pihak perusahaan. Fungsinya untuk menyamakan aplikasi yang akan dibuat dengan sistem yang telah ada agar dapat sesuai dan mudah saat diimplementasi. Hasil dari wawancara tersebut dicatat dan dijadikan dokumentasi untuk acuan pembuatan aplikasi. Sehingga, bila terjadi perbedaan permintaan dapat segera diketahui dan didiskusikan terlebih dahulu sebelum proses selanjutnya dapat dimulai. Narasumber dari proses wawancara tersebut diantaranya adalah Bapak Doni dari bagian keuangan yang bertanggung jawab atas penyusutan nilai aset, Bapak Rendra selaku asisten manager dari bagian IT, Bapak Probo selaku supervisor dari bagian IT, Bapak Pambudi, Bapak Catra, Bapak Dita dan Bapak Eko sebagai karyawan di bagian IT.

Proses ini diperlukan untuk dapat memperoleh informasi mengenai permasalahan-permasalahan apa saja yang terjadi pada perusahaan, yang dalam kasus ini perusahaan tersebut adalah PT BJTI. Proses ini juga berfungsi untuk mengetahui kebutuhan aplikasi yang sebenarnya diperlukan perusahaan untuk membantu proses bisnisnya. Serta, mengetahui keinginan

atau karakteristik *user* yang nantinya akan berhubungan langsung dengan aplikasi tersebut.

B. Analisis Dokumen

Proses ini digunakan untuk mengamati dan menganalisis dokumen-dokumen yang berhubungan dengan kegiatan penilaian kondisi aset. Dokumen yang perlu diamati dan dianalisis diantaranya adalah dokumen penyusutan nilai ekonomis aset dan dokumen perbaikan aset.

Dokumen penyusutan nilai aset dan dokumen perbaikan bagian biaya yang dikeluarkan akan digunakan sebagai acuan untuk menghitung evaluasi kondisi mesin berdasarkan nilai ekonomisnya. Dokumen perbaikan aset akan dipergunakan sebagai acuan penilaian kinerja sebuah mesin. Hal tersebut dapat dihitung melalui lamanya waktu perbaikan dan juga intensitas kegiatan perbaikan pada suatu mesin.

3.1.2 User Requirements

Setelah melakukan proses wawancara dan analisis dokumen pada bagian manajerial, mekanik dan keuangan. Dibutuhkan sebuah sistem yang dapat menilai kondisi sebuah mesin HMC secara kinerja maupun secara nilai ekonomis. Maka, didapat dua *user requirement* yang dibutuhkan oleh perusahaan PT BJTI adalah sebagai berikut

A. Perhitungan Kinerja Aset

Perhitungan kinerja aset dapat digunakan sebagai penghitung kinerja sebuah mesin. Proses berikut ini merupakan proses yang dibutuhkan *user* untuk dapat melihat kondisi mesin dari segi kinerja mesin tersebut selama

digunakan. Berikut adalah penjelasan mengenai *user requirements* perhitungan kinerja aset :

Tabel 3. 1 *User Requirement* Perhitungan Kinerja Aset

Deskripsi	Fungsi ini digunakan oleh bagian mekanik. Bagian mekanik bertugas untuk <i>menginputkan</i> data perbaikan aset yang dilakukan. Fungsi ini akan digunakan untuk mencari nilai kinerja suatu aset yang dapat digunakan oleh pihak mekanik sebagai acuan pelaporan kepada pihak manajerial agar dapat melakukan evaluasi kondisi suatu aset.
Aktor	Bagian Mekanik.
Input	Data perbaikan aset. Yaitu mulai dari tanggal, biaya dan waktu mulai hingga berakhirnya suatu perbaikan.
Proses	1. <i>Menginputkan</i> data perbaikan aset. 2. Simpan data perbaikan aset.
Output	1. Data <i>Availability</i> (ketersediaan) aset. 2. Data <i>Reliability</i> (keandalan) aset.
Peraturan	Hanya berfungsi pada aset yang tidak dalam status tidak aktif

B. Perhitungan Nilai Ekonomis

Perhitungan nilai ekonomis dapat digunakan *user* untuk mengetahui nilai suatu mesin. Proses ini merupakan salah satu proses yang diperlukan untuk memberikan evaluasi kinerja mesin pada sistem. Proses ini akan memberikan pandangan nilai sebuah mesin dari segi ekonomis mesin tersebut selama digunakan. Berikut adalah penjelasan mengenai *user requirement* perhitungan nilai aktiva :

Tabel 3. 2 *User Requirement* Perhitungan Nilai Ekonomis

Deskripsi	Fungsi ini digunakan oleh bagian keuangan. Bagian keuangan akan <i>menginputkan</i> data aktiva dari masing-masing aset. Fungsi ini nantinya akan digunakan untuk menghitung nilai ekonomis suatu aset yang nantinya akan digunakan sebagai acuan untuk melaporkan suatu kondisi keuangan suatu aset kepada manajerial.
Aktor	Bagian Keuangan.

Input	1. Periode
	2. Data rekening
	3. Data aktiva
Proses	1. Memilih aset yang belum memiliki nilai aktiva.
	2. Memilih rekening yang digunakan.
	3. Simpan data.
Output	1. Data aktiva mesin
	2. data nilai ekonomis mesin.
Peraturan	1. Aset yang telah memiliki nilai aktiva pada suatu periode tidak dapat terpilih pada perhitungan aktiva pada periode tersebut lagi.
	2. Data aset yang <i>diinputkan</i> nilai aktiva tidak boleh dua kali.
	3. Id aktiva masing-masing aset tidak boleh ada yang sama dari keseluruhan periode yang ada.

3.1.3 Software Requirements

Berdasarkan hasil analisis dari *user requirements* di atas, maka dibutuhkan *software requirements* yang dapat menunjang fungsi evaluasi kinerja aset. Terdapat tiga *software requirements* yang dibutuhkan, diantaranya adalah:

A. Perhitungan Kinerja Aset

Pada *software requirement* perhitungan kinerja aset sistem akan menghitung nilai *availability* dan *reliability* suatu mesin. hal tersebut nantinya yang akan digunakan sebagai acuan penilaian evaluasi mesin dari segi kinerja.

Tabel 3. 3 *Software Requirement* Perhitungan Kinerja Aset

Deskripsi	Fungsi ini digunakan oleh bagian mekanik. Bagian mekanik akan <i>menginputkan</i> data perbaikan suatu aset yang terjadi agar nantinya data tersebut dapat dihitung oleh sistem untuk mendapatkan nilai kinerja aset tersebut.
Pemicu	

Alur	Aktor	Sistem
	Aktor membuka halaman perbaikan aset	Aplikasi menampilkan seluruh data perbaikan dalam bentuk tabel. Data tersebut akan
		ditampilkan dengan pembagian 10 data setiap halamannya. Dan data akan mulai ditampilkan dari data yang terbaru.
	Aktor mengklik tombol mulai.	Aplikasi menampilkan form perbaikan.
		Aplikasi menampilkan data aset yang sudah ada dalam sistem dalam bentuk pilihan dengan kondisi tertentu.
	Aktor memilih aset yang mengalami perbaikan.	
	Aktor mengisi form perbaikan yang terdiri dari biaya perbaikan waktu dan tanggal mulai perbaikan.	
	Aktor mengklik tombol simpan	Aplikasi membaca dan mengambil data yang <i>diinputkan</i> ke dalam form.
		Aplikasi mengubah status aset menjadi perbaikan.
		Aplikasi menyimpan data tersebut ke dalam <i>datebase</i> .
	Hingga pada saat perbaikan selesai aktor kembali mengakses halaman perbaikan aset.	
	Aktor mengklik tombol selesai.	Aplikasi menampilkan form perbaikan.
		Aplikasi menampilkan data aset yang mengalami perbaikan yang sudah ada pada sistem dalam bentuk pilihan.
	Aktor memilih data aset yang ingin <i>diupdate</i> data perbaikannya.	
	Aktor mengisi form <i>update</i> perbaikan.	
	Aktor mengklik tombol <i>update</i> .	Aplikasi membaca data aset yang telah dipilih oleh aktor.

Alur	Aktor	Sistem
		Aplikasi menghitung jumlah total waktu perbaikan dari aset yang telah dipilih.
		Nilai perhitungan jumlah total waktu perbaikan tersebut akan ditampung ke dalam sebuah variabel.
		Aplikasi menghitung MTTR (<i>Mean Time To Repair</i>) dari aset yang dipilih. Dengan rumus jumlah total waktu perbaikan dibagi dengan jumlah banyaknya aset mengalami <i>down time</i> .
		Nilai dari perhitungan MTTR akan ditampung ke dalam sebuah variabel.
		Kemudian aplikasi akan melakukan perhitungan MTBF (<i>Mean Time Before Failure</i>). Dengan rumus jumlah total waktu perbaikan dibagi dengan jumlah <i>uptime</i> yang telah dialami oleh aset.
		Nilai dari perhitungan MTBF juga akan ditampung ke dalam sebuah variabel.
		Selanjutnya aplikasi akan menghitung nilai <i>availability</i> (ketersediaan) aset tersebut. Dengan rumus MTBF dibagi dengan jumlah MTBF ditambah dengan MTTR.
		Hasil dari perhitungan nilai <i>availability</i> tersebut akan ditampung ke dalam sebuah variabel.
		Kemudian aplikasi akan menghitung nilai <i>reliability</i> (keandalan). Sistem akan menghitung eksponensial pangkat minus dari perkalian antara rate kegagalan dengan jumlah waktu perbaikan.
		Nilai rate kegagalan didapat dari membagi jumlah kegagalan dengan total waktu kerja.

Aktor	Aktor	Sistem
		Hasil perhitungan nilai <i>reliability</i> akan ditampung ke dalam sebuah variabel.
		Aplikasi akan <i>update</i> tanggal selesai perbaikan pada <i>database</i> .
		Aplikasi akan <i>update</i> status aset menjadi aktif.
		Aplikasi akan menyimpan data total biaya perbaikan pada <i>database</i> .
		Aplikasi akan menyimpan data nilai <i>availability</i> dan <i>reliability</i> aset yang ditampung pada beberapa variabel ke dalam <i>database</i> .
		Aplikasi menyimpan biaya perbaikan.
		Aplikasi mengambil data total biaya perbaikan selama ini.
		Aplikasi menjumlahkan biaya total perbaikan selama ini pada aset tersebut dengan biaya perbaikan pada perbaikan saat ini.
		Aplikasi <i>update</i> data total biaya perbaikan.
Akhir	Aplikasi menyimpan data perhitungan tanggal selesai, total biaya perbaikan, <i>availability</i> dan <i>reliability</i> mesin/aset.	
Non Fungsional	<ul style="list-style-type: none"> - Kondisi tertentu dimana aktor mengklik tombol mulai adalah kondisi dimana aplikasi melakukan filter pada data aset yang ditampilkan hanya yang berstatus aktif. - Kondisi tertentu dimana aktor mengklik tombol selesai adalah kondisi dimana aplikasi melakukan filter pada data aset yang ditampilkan hanya yang berstatus diperbaiki. 	

B. Perhitungan Nilai Ekonomis

Pada fungsi ini aplikasi akan menghitung nilai nilai yang berkaitan dengan nilai ekonomis suatu mesin. Yaitu, nilai aktiva mesin, jumlah biaya perbaikan. Hal-hal tersebut yang nantinya dijadikan acuan dalam proses evaluasi mesin berdasarkan nilai ekonomisnya.

Tabel 3. 4 *Software Requirement* Perhitungan Nilai Ekonomis

Deskripsi	Fungsi ini digunakan untuk menghitung nilai ekonomis suatu aset yang nilai <i>input</i> annya diambil dari biaya perbaikan aset dan dari nilai aktiva suatu aset, sehingga nantinya akan muncul pertimbangan nilai ekonomis apakah aset tersebut merugikan atau tidak.	
Pemicu		
Alur	Aktor	Sistem
	Aktor membuka halaman aktiva.	Aplikasi menampilkan form aktiva.
		Aplikasi menampilkan data aset yang sudah ada dalam sistem dalam bentuk pilihan dengan kondisi tertentu.
	Aktor mengisi form aktiva.	
	Aktor mengklik simpan.	Aplikasi akan mengubah status aset menjadi aktif.
		Aplikasi menghitung tafsiran nilai buku wajar (TNBW) dengan rumus nilai perolehan atau nilai beli aset dikali dengan 2%.
		Nilai dari TNBW akan ditampung ke dalam sebuah variable.
		Kemudian aplikasi akan menghitung nilai perolehan setelah tafsiran (NPST) dengan rumus nilai perolehan dikurangi dengan TNBW.
		Hasil dari perhitungan NPST akan ditampung di dalam sebuah variable.
		Selanjutnya aplikasi akan menghitung nilai penyusutan dengan rumus NPST dibagi dengan masa manfaat.
	Nilai susut akan ditampung ke dalam sebuah variabel.	
	Kemudian aplikasi akan menghitung nilai buku. Rumusnya adalah nilai perolehan dikurangi nilai susut pertahunnya.	

Alur	Aktor	Sistem
		<p>Nilai buku akan disimpan ke dalam sebuah variabel.</p> <p>Selanjutnya aplikasi akan menghitung sisa manfaat dengan rumus jumlah dari nilai buku dikurangi TNBW kemudian dibagi dengan nilai susut.</p> <p>Nilai sisa manfaat akan ditampung ke dalam sebuah variabel.</p> <p>Kemudian aplikasi akan menyimpan nilai-nilai tersebut ke dalam <i>database</i>.</p>
	Aktor membuka halaman transaksi aktiva.	<p>Aplikasi menampilkan form transaksi aktiva.</p> <p>Aplikasi mengambil data transaksi aktiva.</p> <p>Aplikasi menampilkan data transaksi aktiva dalam bentuk tabel dimulai dari transaksi yang paling baru.</p>
	Aktor memasukkan periode aktiva.	
	Aktor mengisi form transaksi aktiva.	
	Aktor mengklik tombol proses.	<p>Aplikasi mengambil data aset yang memiliki nilai aktiva.</p> <p>Aplikasi memfilter data nilai aktiva dengan kondisi tertentu.</p> <p>Aplikasi akan mengurangi nilai aktiva aset yang terfilter dengan masing-masing nilai residunya pada periode tersebut sesuai yang ada pada tabel aset yang berbeda nilainya satu dengan yang lainnya..</p> <p>Aplikasi akan <i>update</i> nilai buku masing-masing aset yang terkena proses transaksi di atas.</p> <p>Aplikasi menyimpan nilai aktiva baru sesuai periode <i>inputan</i> ke dalam <i>database</i>.</p>
Akhir		Aplikasi menyimpan data-data aktiva aset.

Non Fungsional	<ul style="list-style-type: none"> – Kondisi tertentu pada halaman aktiva adalah dimana data aset yang tampil pada pilihan hanya data aset yang berstatus tanpa aktiva. – Kondisi tertentu pada proses transaksi filter data aktiva adalah kondisi dimana data transaksi aktiva masih belum ada pada periode yang telah <i>diinputkan</i> oleh aktor. Aktifa yang terfilter masih belum mempunyai data transaksi pada periode tersebut akan ditampung pada variabel array yang nantinya satu persatu akan dihitung proses transaksinya oleh aplikasi. – kondisi periode pada halaman aktiva dan transaksinya adalah dalam periode bulanan.
-----------------------	---

C. Evaluasi Mesin HMC

Pada fungsi ini aplikasi akan menilai dari kedua penilaian mesin.

Penilaian pertama dari segi kinerja dan penilaian kedua dari segi ekonomis.

Sehingga aplikasi dapat memberikan informasi apakah mesin tersebut masih layak digunakan atau tidak.

Tabel 3. 5 *Software Requirement* Evaluasi Mesin HMC

Deskripsi	Fungsi ini digunakan untuk mengambil nilai dari perhitungan kinerja dan perhitungan nilai ekonomis. Dari pengambilan data kedua hasil perhitungan tersebut, aplikasi dapat memberikan informasi tentang kelayakan sebuah mesin.	
Pemicu		
Alur	Aktor	Sistem
	Aktor mengakses halaman evaluasi.	Aplikasi menampilkan form evaluasi. Aplikasi menampilkan data aset yang sudah ada dalam sistem dalam bentuk pilihan dengan kondisi tertentu.
	Aktor memilih aset.	
	Aktor memilih periode.	
	Aktor mengklik tombol proses.	Aplikasi membaca <i>inputan</i> . Aplikasi mengambil data perhitungan <i>availability</i> dan <i>reliability</i> sesuai dengan aset dan periode <i>inputan</i> .

Alur	Aktor	Sistem
		Nilai tersebut akan ditampung dalam sebuah variable dan ditampilkan dalam bentuk persentase.
		Kemudian nilai tersebut akan dibandingkan dengan standar dan bobot 90% yang telah diberikan oleh pihak perusahaan.
		Aplikasi membandingkan nilai <i>availability</i> dan <i>reliability</i> dengan bobot yang ada dan memberikan saran.
		Pemberian saran akan dibagi tiga, masih layak, perlu pertimbangan, dan sudah tidak layak.
		Apabila nilai <i>availability</i> dan <i>reliability</i> diatas 90%. Maka panel akan berwarna biru dan mengeluarkan evaluasi “Mesin Masih Layak”
		Apabila salah satu nilai <i>availability</i> atau <i>reliability</i> dibawah 90%. Maka panel akan berwarna hijau dan mengeluarkan evaluasi “Mesin Perlu pertimbangan”
		Apabila nilai <i>availability</i> dan <i>reliability</i> dibawah 90%. Maka panel akan berwarna merah dan mengeluarkan evaluasi “Mesin Tidak Masih Layak”
		Aplikasi mengambil nilai aktiva dan biaya perbaikan.
		Aplikasi menghitung biaya total perbaikan.
		Nilai aktiva akan dinilai, apakah mesin tersebut masih memiliki nilai ekonomis yang tersisa.

Alur	Aktor	Sistem
		Aplikasi menampilkan penilaian dari kedua sisi dan memberikan masukan berupa masih layak atau tidaknya mesin tersebut. Sehingga pihak aktor akan menentukan selanjutnya.
Akhir	Aplikasi menampilkan informasi kelayakan mesin.	
Non Fungsional	– Kondisi tertentu pada proses adalah dimana aset yang ditampilkan hanyalah aset yang bukan berstatus tidak aktif.	

3.1.4 Data Requirements

Dari tabel *software requirements* di atas, maka diperlukan beberapa data yang dibutuhkan dan dapat mendukung kinerja *software requirements* tersebut, data tersebut antara lain adalah:

A. Data Aset

Data aset ini sudah dimiliki oleh pihak PT BJTI sehingga penulis diperbolehkan untuk melihat data dan mencatat beberapa diantaranya. Data tersebut dijadikan sampel penelitian kasus yang ditangani di perusahaan tersebut.

B. Data Rekening

Data ini sudah dimiliki oleh pihak PT BJTI. Data ini nantinya akan berhubungan dengan penggunaan aktiva aset sehingga penulis diberikan izin untuk mengkopi tabel tersebut dengan persyaratan tidak boleh disebar luaskan kepada pihak lain.

C. Data Aktiva.

Data ini sudah dimiliki oleh pihak PT BJTI. Namun data tersebut masih belum berupa tabel *datebase*. Karena selama ini pihak perusahaan masih menghitung nilai aktiva suatu aset secara manual. Nilai aktiva masih dihitung dengan menggunakan *Microsoft Excel*. Oleh karena itu penulis membuat tabel baru untuk data aktiva. Pengisian data aktiva akan disesuaikan dengan data aktiva yang sudah ada di *Microsoft Excel*. Penghitungan nilai aktiva nantinya akan lebih mudah dilakukan pihak perusahaan dengan mengotomatiskan perhitungan. Nantinya pihak perusahaan cukup dengan memilih aset dan periode yang ingin diketahui data aktivanya. Setelah itu aplikasi akan menghitung nilai aktiva dan menyimpannya ke dalam tabel aktiva dan transaksi.

D. Data Perbaikan Aset.

Data ini sudah dimiliki oleh pihak PT BJTI. Data ini dapat dilihat oleh penulis untuk dapat mengetahui alur perbaikan sebuah aset. Dari mulainya perbaikan hingga proses jam kerja suatu aset.

E. Data Kinerja

Data ini akan dibuat oleh penulis dengan tujuan sebagai tabel yang nantinya digunakan untuk menyimpan hasil penilaian kinerja suatu aset.

3.1.5 *Nonfunctional Requirements*

Kebutuhan *nonfunctional* adalah salah satu kebutuhan yang harus diperhatikan dalam pembuatan aplikasi evaluasi ini selain kebutuhan kebutuhan lainnya, kebutuhan *nonfunctional* diantaranya adalah :

A. Keamanan (*Security*)

Aplikasi diberikan beberapa fitur pencegahan pengguna yang tidak berkepentingan untuk menggunakannya. Diantaranya adalah fitur *id password* sehingga hanya *user* tertentu yang dapat mengakses aplikasi tersebut. Kemudian *password* akan terenkripsi sehingga pihak adminpun tidak mengetahui *password* milik orang lain apalagi pihak lain yang ingin mencoba membaca *password user* lain. Selain itu terdapat fitur *auto logout* yang akan membuat halaman kembali ke halaman *login* setelah ditinggal beberapa menit oleh *user*.

3.2 Perancangan Aplikasi

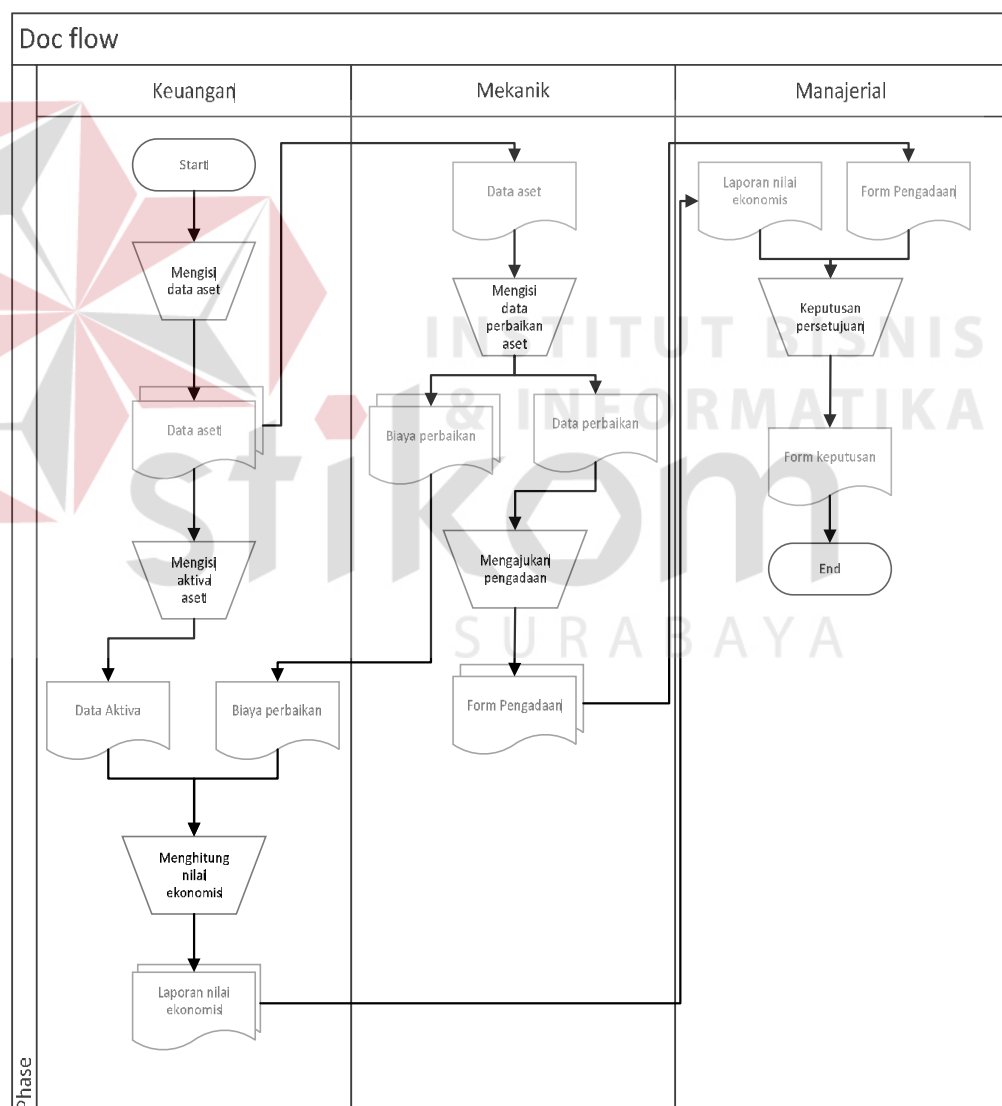
3.2.1 Desain Proses

Dari hasil *Software requirement* di atas, maka akan dapat kita lihat terdapat beberapa fungsi yang menjadi bagian utama aplikasi. Dari hal tersebut maka akan dapat digambarkan *Doc Flow*, *System Flow*, *Context* dan DFD untuk dapat lebih jelas melihat alur dari sistem tersebut.

A. *Document Flow*

Pada sistem yang lama perusahaan PT BJTI masih terdapat langkah yang tidak sejalan. Misalnya saja pihak mekanik yang tidak menyertakan bukti untuk mengajukan pengadaan. Hal ini bukan tanpa alasan, dikarenakan pihak mekanik masih hanya mencatat perbaikan aset saja tanpa menghitungnya menjadi sebuah angka yang dapat menjadi bukti tertulis kondisi aset saat ini. Hal ini tentu saja dapat menyebabkan ketidakpercayaan pihak manajerial kepada pihak mekanik bila kondisi mesin memang harus

diganti. Permasalahan yang terdapat pada pihak keuangan disistem yang lama masih melakukan perhitungan nilai aktiva secara manual menggunakan *Ms.Excel*. Jelas ini akan memperbanyak tugas pihak keuangan, dikarenakan harus menghitung setiap aset yang ada di perusahaan tersebut setiap bulannya. Hal tersebut jelas memakan waktu yang lama dan rawan akan terjadinya kesalahan. Untuk lebih jelasnya dapat dilihat pada gambar 3.1 di bawah.



Gambar 3 1 *Document Flow*

Dari gambar 3.1 di atas dapat kita lihat bagaimana alur dokumen dahulu yang terjadi pada perusahaan. Alur dokumen dimulai dari pengisian data aset perusahaan oleh bagian keuangan. Data catatan tersebut dijadikan sebuah dokumen dan digandakan. Dokumen yang digandakan gunanya untuk membantu proses lainnya yang berkaitan dengan aset dibagian lain. Pada bagian keuangan dokumen aset tersebut akan langsung digunakan untuk membuat dokumen aktiva aset. Dokumen aktiva ini gunanya untuk mengetahui nilai ekonomis suatu aset berdasarkan usianya. Pada bagian mekanik dokumen aset akan digunakan untuk mengisi kegiatan perbaikan suatu aset. Kegiatan perbaikan tersebut nantinya akan menghasilkan dua buah dokumen yang berkaitan. Yang pertama dokumen perbaikan yang isinya mengenai catatan perbaikan sebuah aset. Dan yang kedua adalah dokumen biaya perbaikan. Dokumen ini berisikan tentang rincian biaya yang dikeluarkan oleh pihak mekanik selama proses perbaikan aset. Dokumen tersebut digandakan dan salinannya akan diserahkan kepada pihak keuangan. Dokumen yang berisikan rincian biaya perbaikan tersebut akan dihitung bersamaan dengan dokumen aktiva setiap bulannya. Kegiatan tersebut gunanya adalah untuk mengetahui nilai ekonomis suatu aset dan dijadikan dokumen laporan. Dokumen nilai ekonomis aset tersebut digandakan dan salinannya akan diserahkan kepada pihak manajerial. Gunanya dokumen nilai ekonomis tersebut adalah pada saat pihak mekanik mengajukan pengadaan aset. Pihak manajerial akan beracuan pada dokumen nilai ekonomis untuk menentukan persetujuannya.

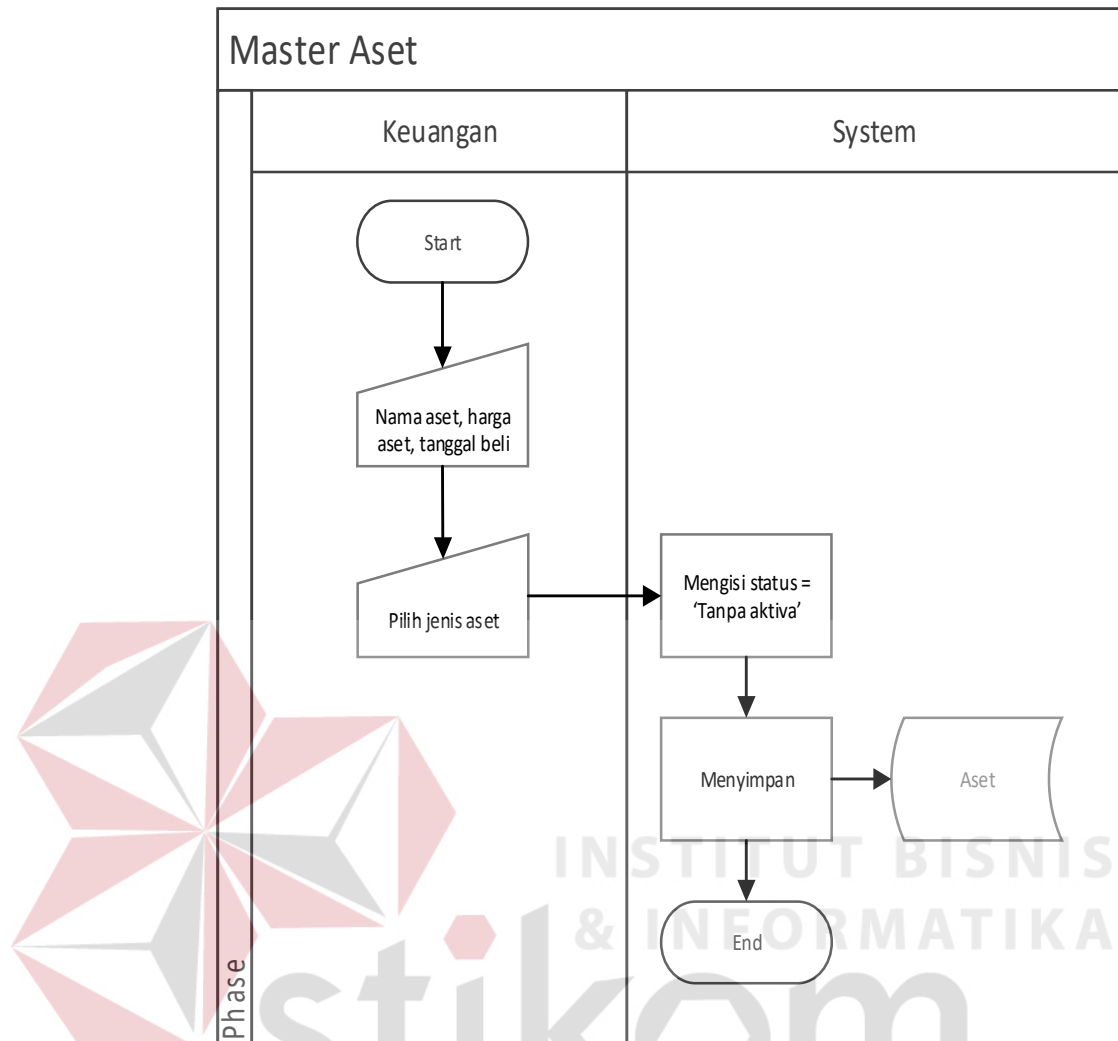
Dalam proses ini masih terlihat bahwa pihak mekanik tidak memiliki acuan kuat bila ingin mengajukan pengadaan. Karena pada dasarnya mereka tidak bisa memberikan penjelasan secara bukti atau nilai yang menunjukkan bahwa aset tersebut memang butuh diganti. Hal ini disebabkan karena pihak manajerial masih hanya menerima satu masukkan penilaian saja, yaitu dari segi nilai ekonomis.

B. System Flow

System flow merupakan sebuah gambaran alur suatu proses yang terjadi dengan sistem. *System flow* dibuat berdasarkan kejadian yang akan dialami oleh sistem secara runtun atau teratur dari mulai hingga berakhirnya suatu proses di dalam sistem.

B.1 System flow Master Aset

System flow ini menggambarkan tentang alur suatu proses dimasukkannya data aset ke dalam basis data melalui sistem. Hal ini dimulai dari proses menginputkan nama, harga dan tanggal beli aset ke dalam form yang telah disediakan oleh aplikasi dari dokumen yang sudah ada. Kemudian, aktor (bag.keuangan) memilih jenis aset yang akan disimpan dari pilihan yang ada pada sistem. Terakhir aktor akan mengklik simpan dan data akan tersimpan. Pada bagian ini sistem akan langsung memberi nilai *default* pada tabel keterangan bahwa kondisi mesin dalam keadaan baik. Untuk lebih jelasnya dapat dilihat pada gambar di bawah, gambar 3.2 *System flow master aset*.

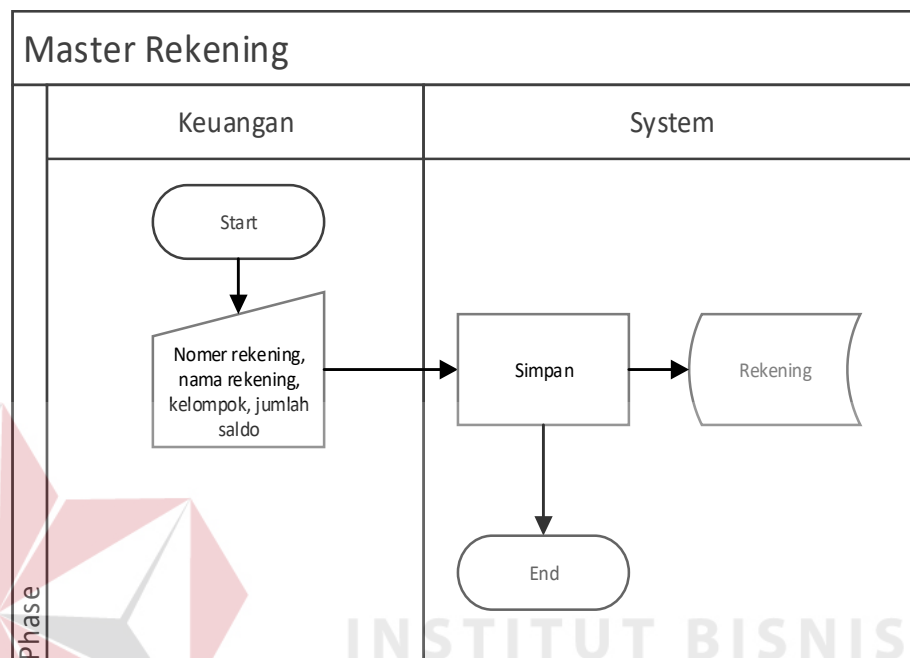


Gambar 3 2 *System Flow Master Aset.*

B.2 *System flow Master Rekening*

System flow ini menggambarkan tentang alur suatu proses dimasukkannya data rekening kedalam basis data melalui sistem. Proses ini dimulai dengan aplikasi mengambil data tanggal sistem saat ini secara otomatis dan menampilkannya di dalam *form inputan*. Kemudian aktor (bag.keuangan) akan *menginputkan* nomer rekening, nama rekening, kelompok, jumlah saldo dan lainnya. Dan terakhir, aktor akan mengklik tombol simpan agar data yang telah *diinputkan*

dapat tersimpan ke dalam tabel rekening. Untuk lebih jelasnya dapat anda lihat pada gambar di bawah, gambar 3.3 *System flow master rekening*.

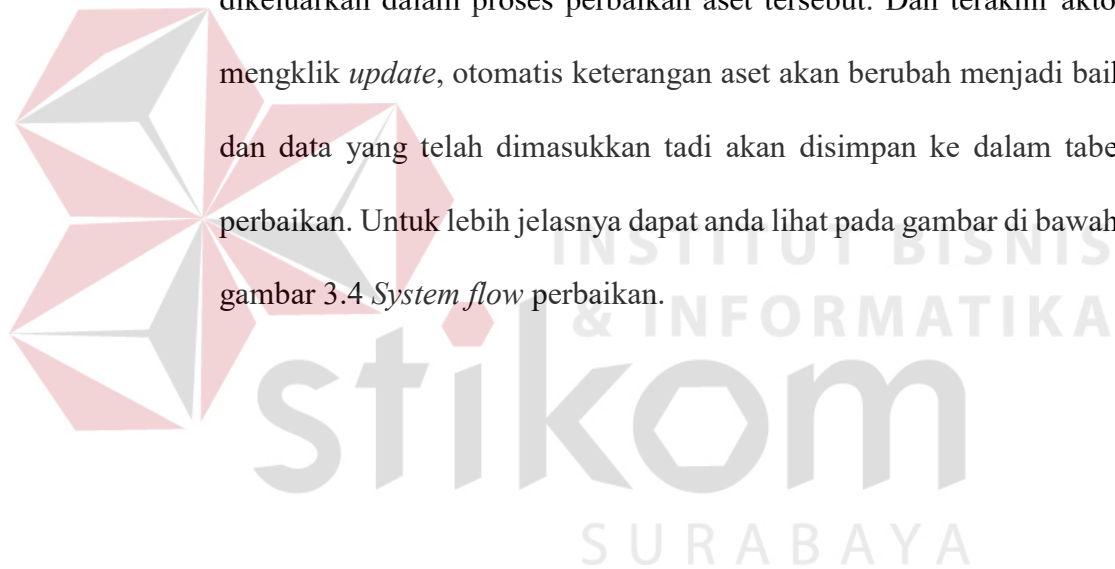


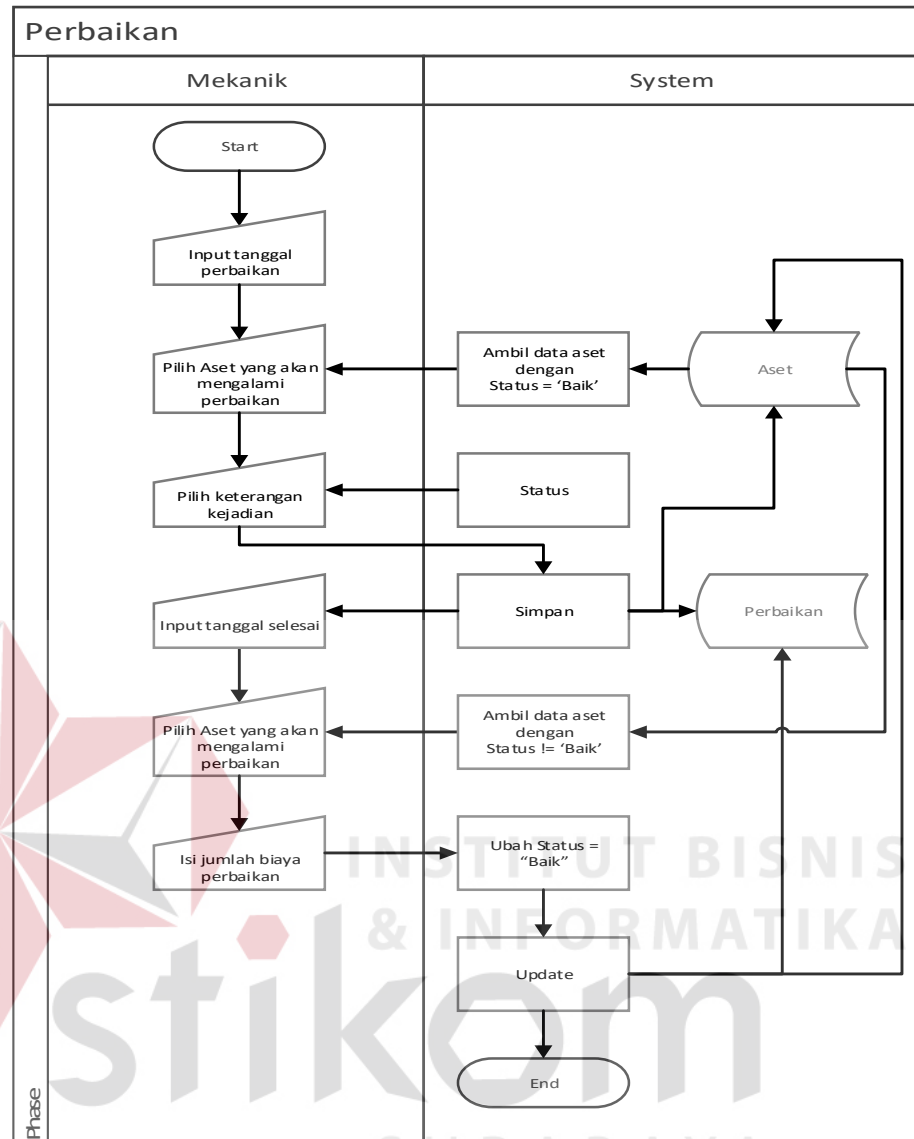
Gambar 3.3 *System Flow Master Rekening*.

B.3 *System Flow Perbaikan*

System flow ini menggambarkan tentang alur suatu proses dimasukkannya data kejadian *realtime* perbaikan aset ke dalam basis data melalui sistem. Proses ini dimulai dengan aplikasi mengambil data tanggal sistem saat ini secara otomatis dan menampilkannya di dalam *form inputan*. Aktor akan mengubah tanggal sesuai kejadian bila tidak sesuai dengan tanggal saat ini. Kemudian aktor (bag.mekanik) akan memilih aset mana yang sedang mengalami perbaikan dari daftar aset yang disaring oleh sistem masih dalam status baik. Setelah itu aktor akan memilih keterangan kejadian yang dialami aset, perbaikan atau

breakdown. Kemudian aktor akan mengklik tombol simpan agar data yang telah *diinputkan* dapat tersimpan ke dalam tabel perbaikan. Apabila proses perbaikan telah selesai dilakukan, maka pihak aktor akan kembali mengakses halaman perbaikan untuk mengembalikan keterangan aset menjadi baik. Pertama aktor akan diminta untuk memilih aset mana yang sudah selesai diperbaiki dari data yang telah disaring sistem berdasarkan status aset tidak sama dengan baik. Kemudian aktor akan diminta untuk mengisi jumlah biaya yang dikeluarkan dalam proses perbaikan aset tersebut. Dan terakhir aktor mengklik *update*, otomatis keterangan aset akan berubah menjadi baik dan data yang telah dimasukkan tadi akan disimpan ke dalam tabel perbaikan. Untuk lebih jelasnya dapat anda lihat pada gambar di bawah, gambar 3.4 *System flow* perbaikan.



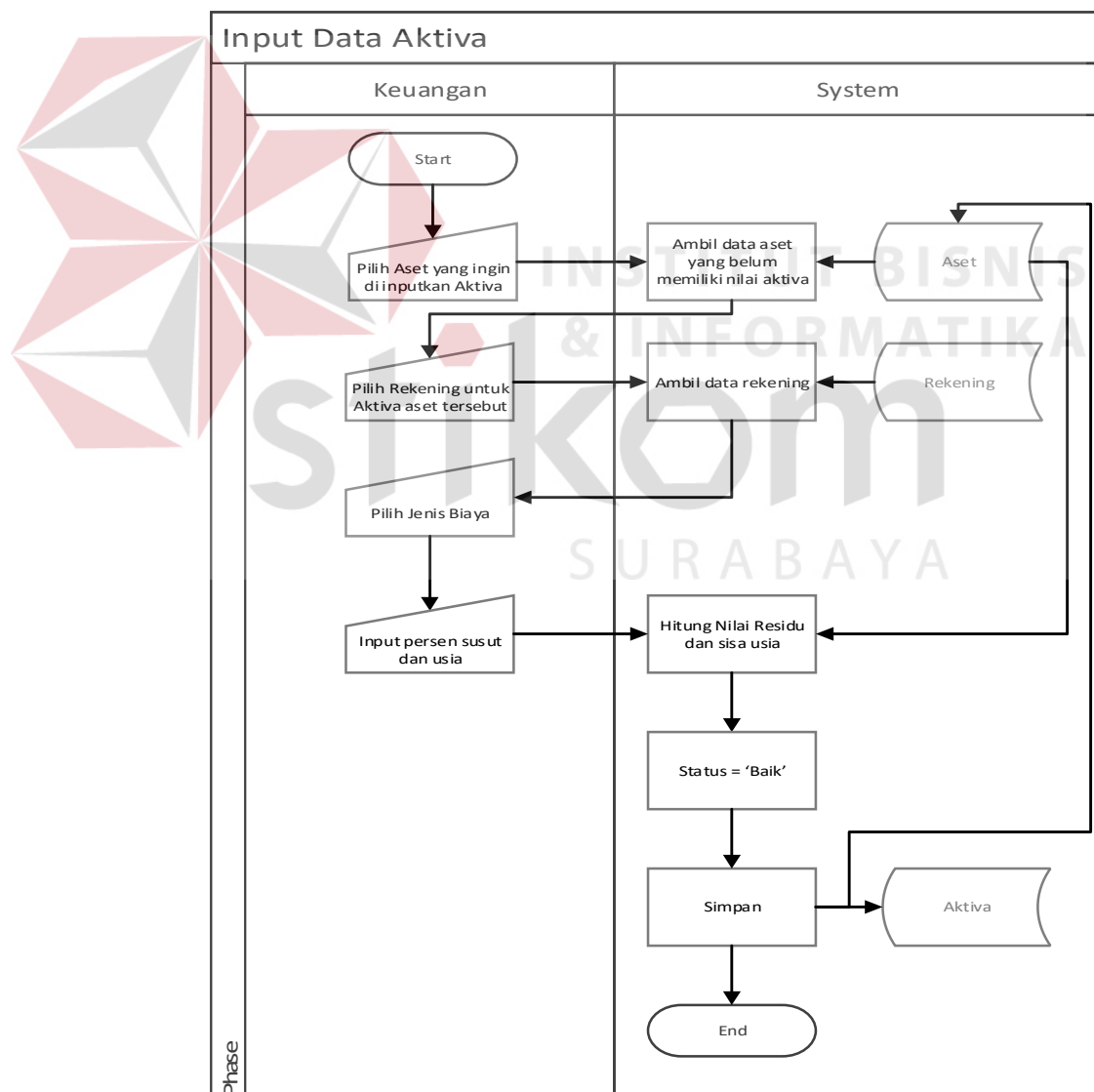


Gambar 3 4 *System Flow* Perbaikan.

B.4 *System Flow* Aktiva

System flow ini menggambarkan tentang alur suatu proses dimasukkannya data aktiva kedalam basis data melalui sistem. Proses ini dimulai dengan aplikasi mengambil data aset yang masih belum mempunyai nilai aktiva. Kemudian aktor (bag.keuangan) akan memilih aset yang akan diinputkan data aktivanya. Kemudian aktor akan memilih rekening dari tabel rekening yang akan digunakan pada

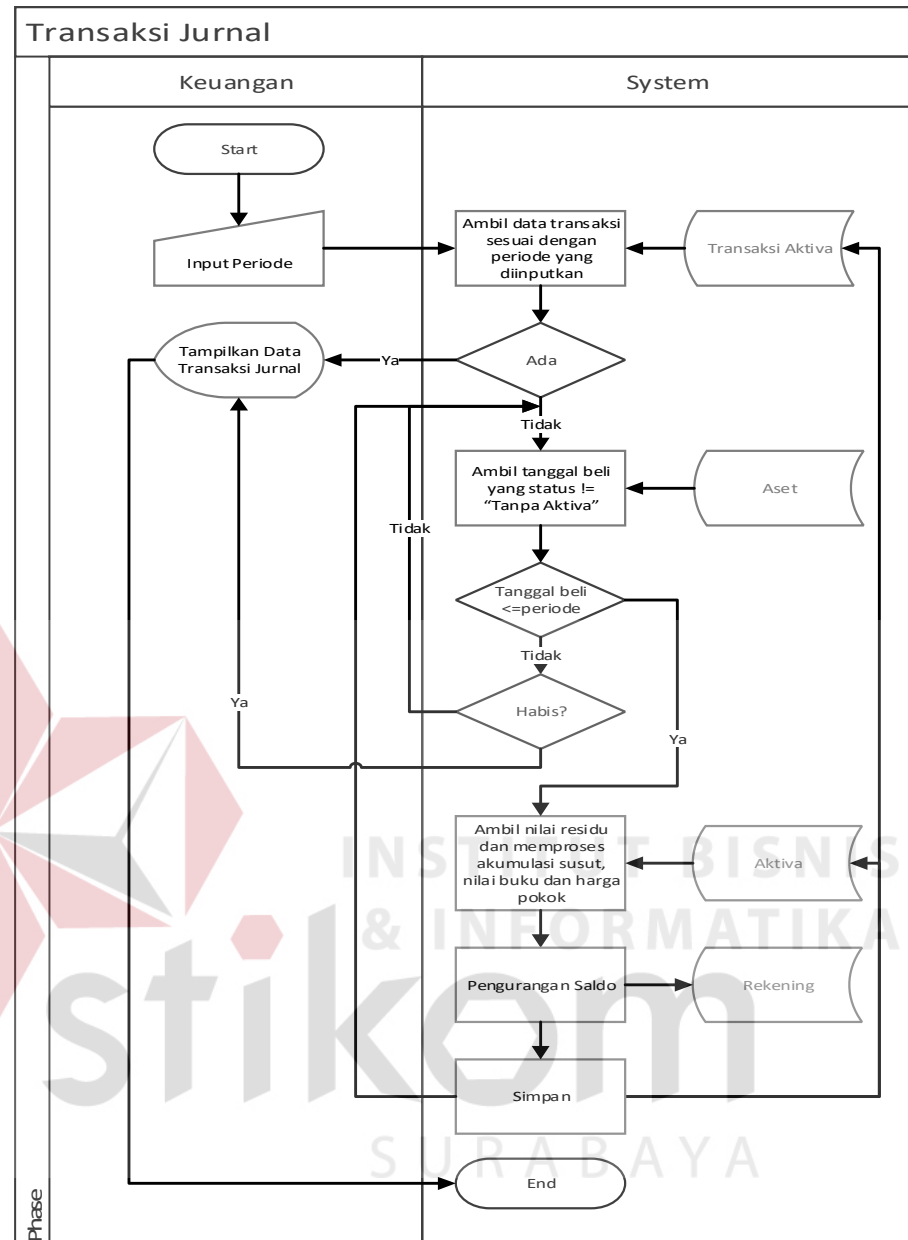
aktiva tersebut. Selanjutnya aktor akan memilih jenis aktiva dari tabel jenis. Kemudian aktor akan *menginputkan* informasi lainnya seperti persen susut aktiva, prediksi usia dan lainnya. Dan terakhir aktor akan mengklik tombol simpan agar data yang telah *diinputkan* dapat tersimpan ke dalam tabel aktiva. Bersamaan dengan hal tersebut sistem akan menghitung nilai residu dan usia aktiva aset dan menyimpannya bersamaan dengan *inputan* aktiva. Untuk lebih jelasnya dapat anda lihat pada gambar di bawah, gambar 3.5 *System flow* aktiva.



Gambar 3.5 *System Flow* Aktiva.

B.5 *System Flow* Transaksi Jurnal

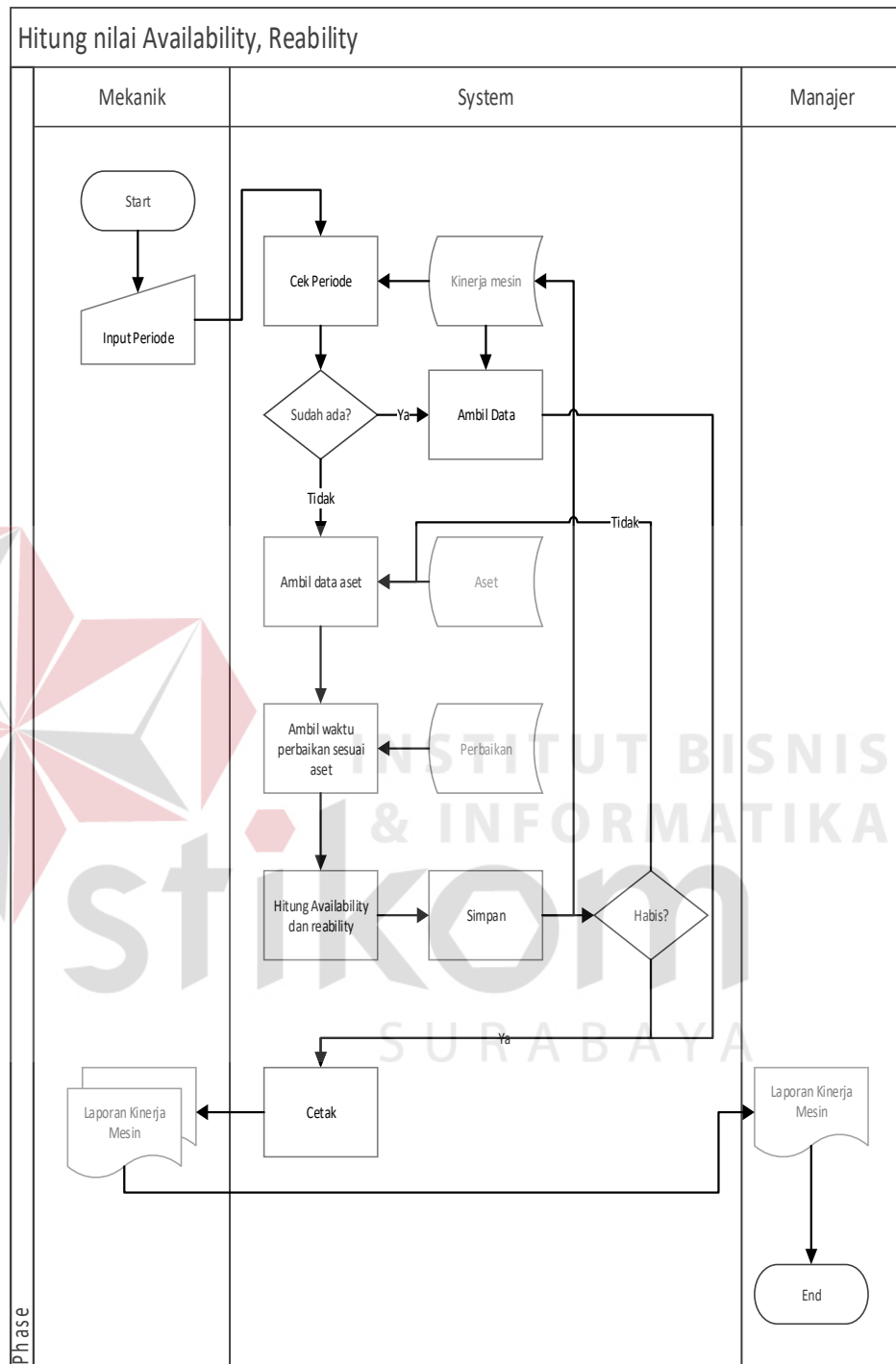
System flow ini menggambarkan tentang alur suatu proses transaksi jurnal melalui sistem. Proses ini dimulai dengan aktor (bag.keuangan) akan menginputkan periode yang ingin diproses transaksi jurnalnya. Kemudian aktor akan mengklik proses. Setelah itu sistem akan mengambil data dari tabel transaksi aktiva yang sesuai dengan *inputan* periode dari aktor. Apabila data yang diinginkan sudah ada, maka sistem akan langsung menampilkan data tersebut ke halaman. Namun, bila data yang dimasukkan tidak ada dalam tabel, maka sistem akan mengambil data tanggal beli aset pada tabel aset yang statusnya tidak sama dengan tanpa aktiva. Aset akan dicek satu persatu oleh sistem, manakah aset yang memiliki tanggal beli kurang dari sama dengan *inputan* periode oleh aktor. Apabila sesuai dengan kriteria tersebut sistem selanjutnya akan mengambil nilai residu aset tersebut dari tabel aktiva. Kemudian akan dihitung akumulasi susut, nilai buku dan harga pokok dari aset tersebut dan diupdate ke dalam tabel aktiva dan transaksi aktiva. Untuk lebih jelasnya dapat anda lihat pada gambar di bawah, gambar 3.6 *System flow* transaksi jurnal.



Gambar 3 6 *System Flow* Transaksi Jurnal.

B.6 *System Flow Availability dan Reliability*

System flow ini menggambarkan tentang alur suatu proses transaksi jurnal melalui sistem. Lebih jelasnya dapat dilihat pada gambar 3.7 di bawah.



Gambar 3 7 System Flow Availability dan Reliability

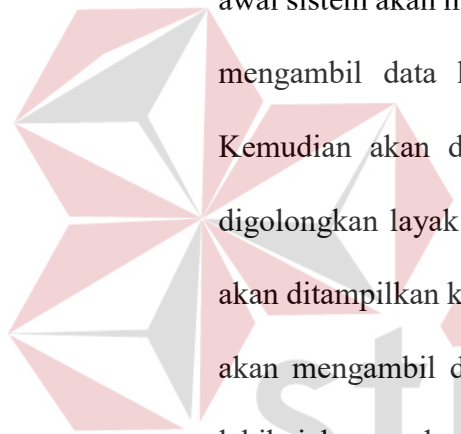
Proses ini dimulai dengan aktor (bag.mekanik) akan menginputkan periode yang ingin diproses perhitungan nilai

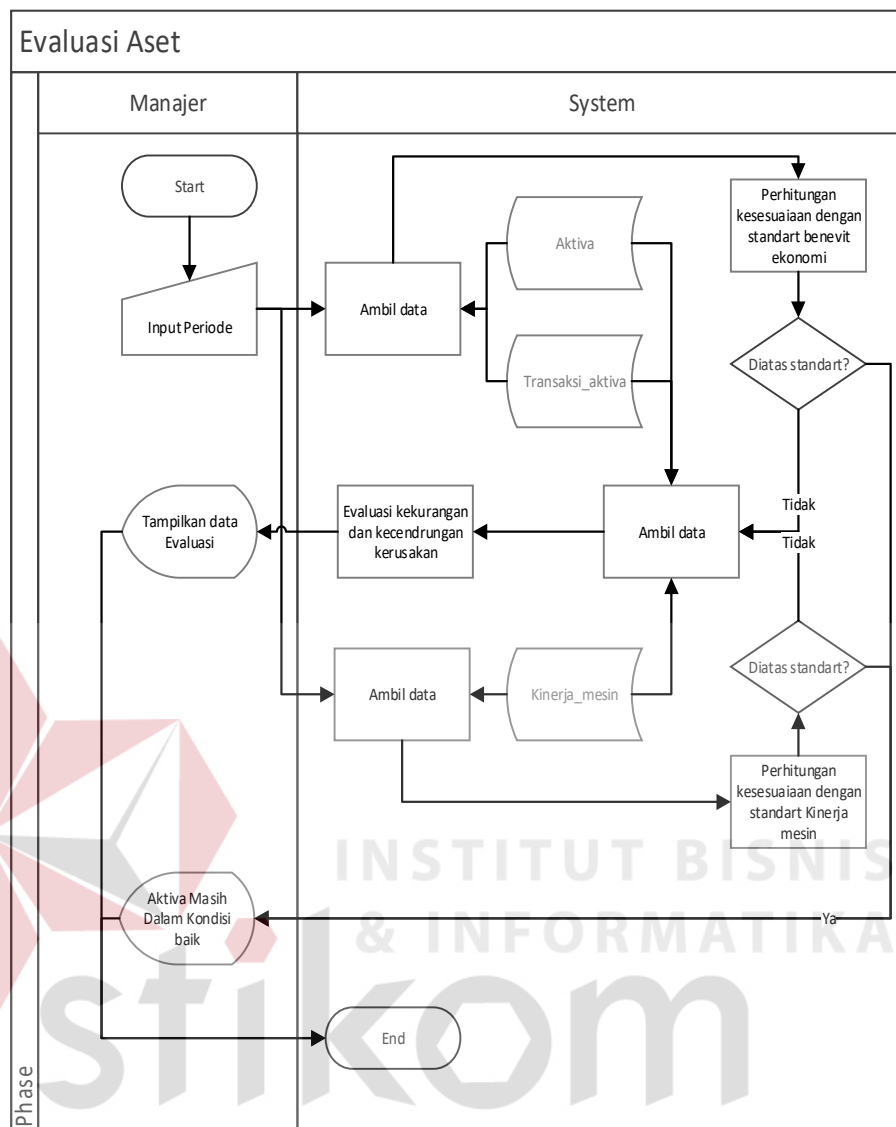
availability dan *reliability*nya. Kemudian aktor akan mengklik proses. Setelah itu sistem akan mengambil data dari tabel kinerja mesin yang sesuai dengan *inputan* periode dari aktor. Apabila data yang diinginkan sudah ada, maka sistem akan langsung menampilkan data tersebut ke halaman dari data yang diambil dari tabel kinerja mesin. Namun, bila data yang dimasukkan tidak ada dalam tabel, maka sistem akan mengambil data id aset pada tabel aset. Kemudian dari id aset yang telah diambil akan dijadikan sebagai acuan dalam pengambilan data waktu perbaikan dalam tabel perbaikan sesuai dengan aset yang telah dipilih. Kemudian dari data perbaikan tersebut akan dihitung nilai *availability* dan *reliability*nya. Kemudian hasil perhitungan akan disimpan. Dan terakhir sistem akan mengecek apakah data masih ada atau sudah habis. Apabila data sudah habis maka proses akan menampilkan semua data yang telah dihitung. Namun, apabila data tersebut belum habis maka proses akan kembali kepada bagian pengecekan dan pengambilan data aset. Untuk lebih jelasnya dapat anda lihat pada gambar 3.7 *System flow* hitung nilai *availability* dan *reliability* di atas.

B.7 System Flow Evaluasi Aset

System flow ini menggambarkan tentang alur suatu proses penilaian aset berdasarkan nilai ekonomi dan kinerjanya. Proses ini bertujuan untuk menampilkan aset-aset yang dianggap tidak memenuhi standar perusahaan. Proses ini diawali dengan bagian manjer mengisi periode yang ingin ditampilkan pada halaman. Dari *inputan* tersebut sistem akan mengambil data dari beberapa tabel untuk dinilai apakah

aset tersebut masih layak atau tidak. Dalam hal ini akan ditentukan dengan dua penilaian. Yang pertama dinilai berdasarkan nilai ekonominya. Sistem akan mengambil data dari tabel aktiva dan transaksi aktiva dari setiap aset yang ada. Kemudian dinilai apakah masih layak atau tidak. Apabila aset dianggap tidak layak maka akan ditampilkan ke halaman berdasarkan kriteria penilaian secara ekonomis, bila layak maka sistem akan melanjutkan ke data aset berikutnya. Kedua sistem akan menilai dari kinerja aset. Dari *inputan* awal sistem akan mengambil data dari tabel kinerja mesin. Sistem akan mengambil data kinerja, seperti nilai *availability* dan *reliability*. Kemudian akan diukur dengan standar perusahaan, apakah masih digolongkan layak atau tidak. Apabila digolongkan tidak layak maka akan ditampilkan ke halaman. Apabila data dianggap layak maka sistem akan mengambil data berikutnya untuk dinilai kelayakannya. Untuk lebih jelasnya dapat anda lihat pada gambar di bawah, gambar 3.8 *System flow* evaluasi aset.



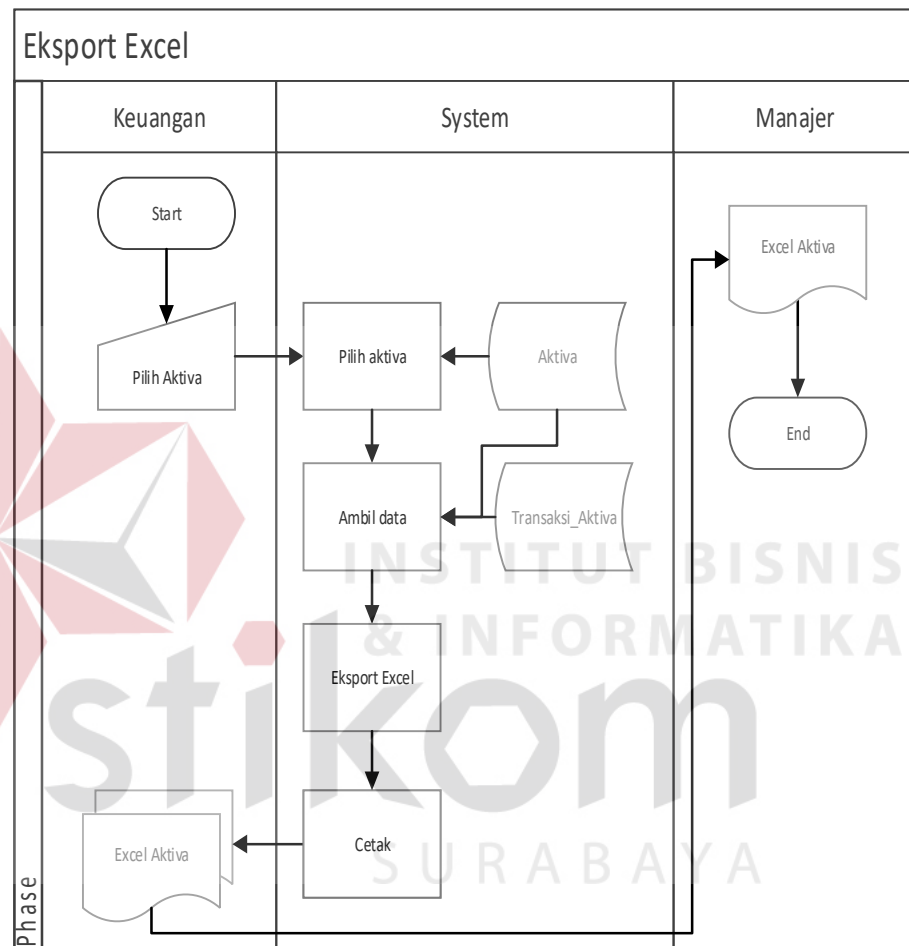


Gambar 3 8 *System Flow* Evaluasi Aset.

B.8 *System Flow Export to Ms.Excel*

System flow ini menggambarkan tentang alur suatu proses diexport suatu data dari tabel aktiva basis data ke *microsoft excel* yang digunakan untuk laporan bulanan pihak keuangan. Proses ini dimulai dari aktor (bag. keuangan) memilih aktiva mana yang ingin dijadikan *excel*. Dari data aktiva yang diambil, sistem selanjutnya akan mengambil transaksi aktiva yang ada untuk aset yang dipilih

sebelumnya. Setelah semua data dikumpulkan dalam tabel. Selanjutnya sistem akan mulai mengekspor tabel kedalam *excel*. Untuk lebih jelasnya dapat anda lihat pada gambar di bawah, gambar 3.9 *System flow export to excel* di bawah.

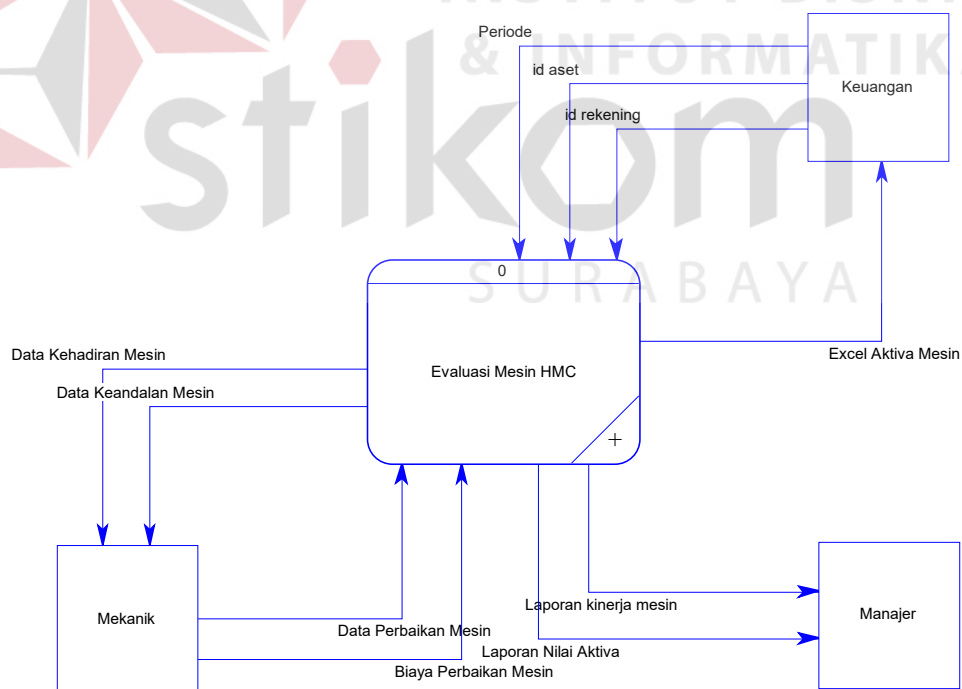


Gambar 3 9 *System Flow Export to Excel*

C. *Context*

Context diagram dibuat dengan tujuan untuk mempermudah pembaca mengerti tentang alur sistem yang ingin dibangun secara menyeluruh. *Context diagram* dibuat berdasarkan proses analisis yang sudah dilakukan oleh penulis sesuai dengan *survey software requirement*. *Context diagram* pada

kasus ini memiliki tiga bagian *Software Requirement*. Diantaranya, perhitungan nilai kinerja aset, pencatatan nilai aktiva beserta perhitungan nilai aktiva dan evaluasi kinerja berdasarkan dua data sebelumnya. Dalam penggunaannya aplikasi ini ditujukan untuk tiga *user*. Yaitu, bagian keuangan, bagian mekanik dan manajerial. Masing-masing *user* akan menjalankan fungsi yang berbeda. Bagian keuangan *user* akan fokus pada proses perhitungan nilai aktiva suatu mesin. Data mesin yang telah dimasukkan nantinya akan diberikan nilai aktiva. Kemudian proses tersebut akan menghitung nilai aktiva suatu mesin secara berkala. Pada bagian mekanik *user* akan fokus pada proses perhitungan kinerja suatu mesin. Mulai dari awal mesin tersebut mengalami kerusakan, hingga proses perbaikan sampai selesai. *User* terakhir adalah bagian manajerial



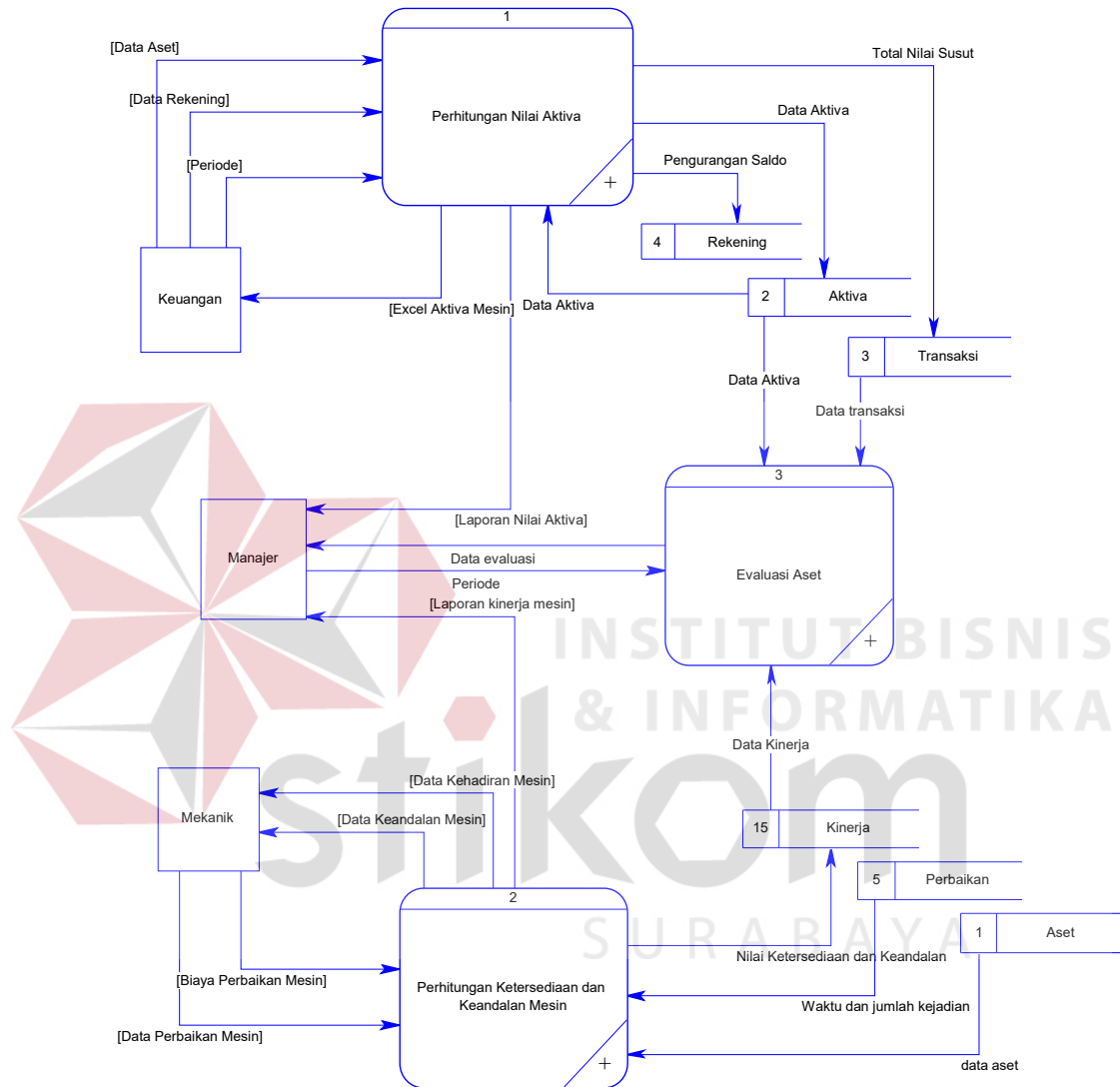
Gambar 3 10 Context Diagram.

Pada bagian keuangan, *user* ditujukan untuk pengisian nilai yang berkaitan dengan nilai ekonomis suatu aset. Nilai aset tersebut akan dihitung berdasarkan rumus perhitungan nilai aktiva. Hasil perhitungan tersebut akan dijadikan acuan nilai pada pengevaluasian kinerja mesin berdasarkan nilai ekonomisnya. Berbeda dengan bagian keuangan, pada bagian mekanik *user* akan memberikan *inputan* berupa data kejadian mesin yang terjadi dilapangan. *Inputan* tersebut berupa data kejadian perbaikan mesin yang berupa tanggal kejadian, biaya dan tanggal selesai. Proses penghitungan dilakukan dengan menggunakan metode perhitungan MTBF MTTR. Dimana hasil perhitungan tersebut akan dijadikan acuan nilai pada pengevaluasian kinerja mesin berdasarkan kinerja mesin. *User* ketiga adalah bagian manajerial, dimana *user* tersebut akan menerima laporan dari nilai evaluasi kinerja masing-masing mesin. Prosesnya adalah *user* akan memilih mesin mana yang akan dilihat dan pada periode berapa. Sistem akan menghitung dan mengumpulkan data nilai dari dua proses sebelumnya dan mengeluarkan nilai dari kedua perhitungan tersebut.

D. DFD Level 0 Aplikasi Evaluasi Mesin HMC

Pada proses ini digambarkan alur sistem yang menjabarkan isi dari *context diagram* diatas. Dalam proses ini akan ditunjukkan hubungan antara ketiga *software requirement* yang telah disebutkan di atas. Ketiga *software requirement* diatas akan dijadikan sebagai proses utama dari aplikasi evaluasi mesin HMC. Dimana pada gambar 3.11 di bawah akan ditunjukkan bagaimana ketiga proses utama tersebut dapat melakukan interaksi dengan ketiga entitas yang ada.

Pada proses DFD *Level 0* Evaluasi Mesin HMC digambarkan secara lebih detil bagaimana relasi antar masing-masing proses utama ataupun dengan entitas. Bagaimana data mengalir dari satu entitas, proses atau *database*.



Gambar 3 11 DFD *Level 0* Aplikasi Evaluasi Mesin HMC

Pada gambar 3.11 di atas ditunjukkan proses pertama dari proses utama adalah pada bagian perhitungan nilai aktiva. Pada proses itu *user* keuangan memasukkan beberapa *inputan*. Diantaranya id aset, id rekening dan periode. Kemudian *user* akan mendapatkan masukan dari sistem berupa dokumen

aktiva mesin dalam bentuk *Ms.Excel*. Sistem dari proses perhitungan nilai aktiva akan mengambil data dari *database*. Data tersebut diantara lain adalah data aset, data rekening dan data aktiva. Dan sistem akan menyimpan *inputan* beberapa data kedalam *database*. Data tersebut diantaranya adalah pengurangan saldo, data aktiva dan total nilai susut. Untuk lebih lengkapnya dapat dilihat pada penjelasan DFD *Level 1* proses perhitungan nilai aktiva.

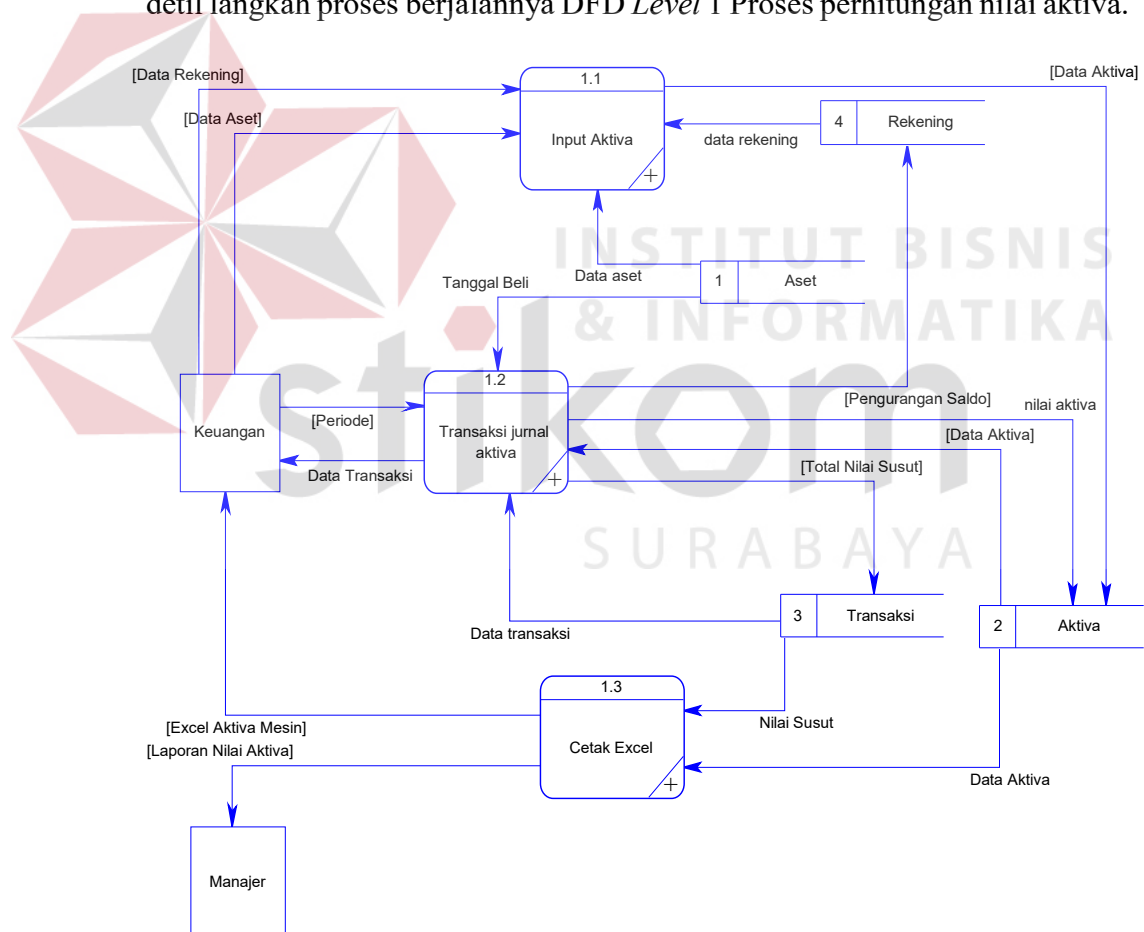
Proses yang kedua pada DFD *Level 0* Aplikasi Evaluasi Mesin HMC adalah proses terjadinya perhitungan kinerja mesin HMC. Dalam gambar do lapmiran 4 ditunjukkan bahwa sistem menerima masukan dari *user* mekanik dan dari beberapa tabel. Data masukan tersebut antara lain adalah data perbaikan mesin, biaya perbaikan mesin, nilai ketersediaan dan keandalan dan yang terakhir adalah waktu dan jumlah kejadian. Pada gambar juga terlihat sistem memberikan *output* kepada *user* mekanik dan kepada beberapa tabel. Data *output* tersebut antara lain adalah data kehadiran mesin, data keandalan mesin dan yang terakhir adalah nilai ketersediaan dan keandalan kinerja mesin. Untuk mengetahui proses yang lebih detil pada proses perhitungan nilai kinerja mesin dapat dilihat pada proses DFD *Level 1* proses perhitungan nilai kinerja mesin.

Pada proses yang terakhir digambarkan adalah proses evaluasi kinerja mesin HMC. Dalam proses tersebut digambarkan *user* manajer yang melakukan interaksi dengan sistem. Dimana sistem menerima masukan dari *user* manajer dan dari tabel. Masukan tersebut diantara lain adalah data aktiva, data detil aktiva, biaya perbaikan, data kinerja mesin dan periode. Kemudian sistem juga memberikan *output* hanya kepada *user* manajer.

Output tersebut diantaraa lain adalah laporan nilai aktiva mesin dan laporan kinerja mesin. Untuk lebih detilnya pada proses ini dapat dilihat pada DFD *Level 1* evaluasi kinerja mesin HMC.

E. DFD *Level 1* Proses Perhitungan Nilai Aktiva

Proses ini menggambarkan secara detil bagaimana satu diantara tiga proses utama berjalan. Proses dari gambar 3.12 di bawah menggambarkan proses perhitungan nilai aktiva. Pada gambar terlihat bahwa proses ini akan dijabarkan menjadi 3 sub-proses sehingga dapat menggambarkan secara lebih detil langkah proses berjalannya DFD *Level 1* Proses perhitungan nilai aktiva.



Gambar 3 12 DFD *Level 1* Perhitungan Nilai Aktiva

Pada gambar 3.12 di atas terlihat bahwa terdapat 3 proses yang ada pada DFD *Level 1* proses perhitungan nilai aktiva. Yang terdiri dari proses *input* aktiva, proses transaksi jurnal aktiva dan terakhir proses cetak ke *Ms,Excel*.

Proses dimulai dari *user* memasukkan data id rekening dan id aset ke dalam proses *input* aktiva. Selanjutnya proses *input* aktiva menerima *inputan* dari beberapa tabel. *Inputan* tersebut antara lain adalah data aset, data rekening dan data jenis. Setelah menerima beberapa *inputan*, proses *input* aktiva memberikan *output* ke tabel aktiva berupa data aktiva.

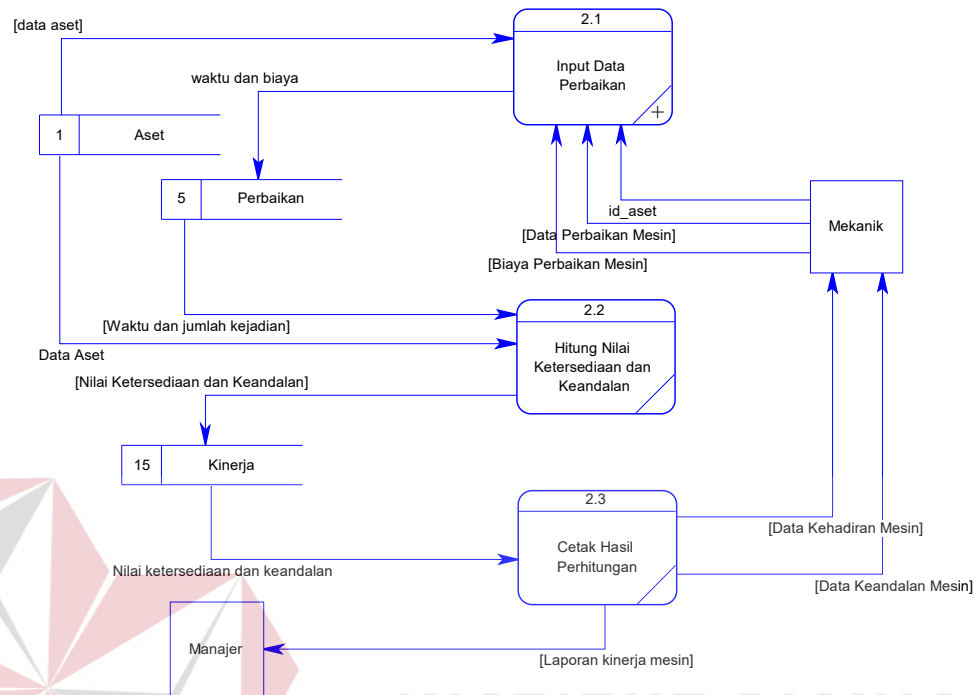
Pada proses selanjutnya dimulai dari *user* keuangan yang memberikan *input* berupa periode kepada proses transaksi jurnal aktiva. Data periode itu dibaca dari tabel aktiva dan menjadikan acuan data yang akan diambil pada tabel aktiva. Kemudian sistem akan melakukan proses perhitungan nilai aktiva berdasarkan usia mesin. Hal ini akan memberikan nilai *output* dari proses kebeberapa tabel. Nilai tersebut antara lain adalah nilai aktiva(baru), pengurangan saldo pada rekening dan menyimpan nilai total susut.

Pada proses ketiga melanjutkan dari proses kedua, dengan alur proses yang menampilkan data berdasarkan *inputan* periode oleh *user* ke sistem, dalam proses ketiga, data yang sudah dihitung sebelumnya akan langsung diekspor ke *Ms.Excel* dan dijadikan sebuah *output* kepada *user* berupa data aktiva mesin.

F. DFD *Level 1* Proses Perhitungan Ketersediaan dan Kinerja Mesin HMC

Proses berikut ini menggambarkan tentang proses perhitungan ketersediaan dan keandalan mesin HMC. Pada gambar 3.13 di bawah akan menjelaskan bagaimana proses dari DFD *Level 1* proses perhitungan

ketersediaan dan keandalan mesin HMC secara lebih detil dari proses *context* sebelumnya. Proses ini adalah salah satu dari tiga proses utama sistem.

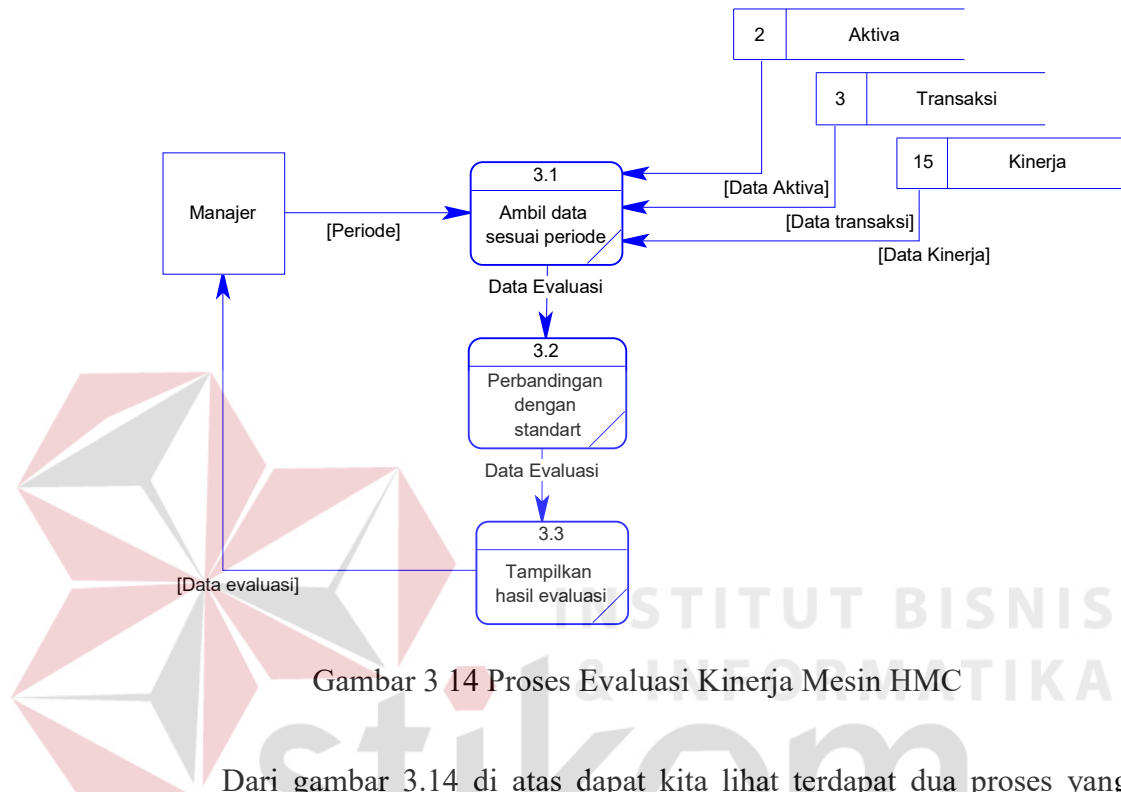


Gambar 3.13 Proses Perhitungan Ketersediaan dan Keandalan Mesin HMC

Dari gambar 3.13 di atas terlihat bahwa proses terbagi menjadi empat bagian. Pada proses pertama terlihat mekanik melakukan *input* kedalam sistem berupa periode yang diambil dari tabel kinerja mesin yang berfungsi untuk mengecek status dari perhitungan kinerja mesin. Dari pengambilan data tersebut akan terlihat, apakah data sudah pernah dihitung sebelumnya atau belum. Apabila data sudah pernah dihitung maka sistem akan langsung menampilkan data yang ada. Apabila data kinerja belum ada, maka sistem akan langsung mengambil data dari setiap aset yang ada. Kemudian sistem juga mengambil data dari tabel perbaikan dengan berdasarkan pada masing-masing data perbaikan yang sesuai dengan data aset. Kemudian sistem akan melakukan perhitungan ketersediaan dan keandalan mesin HMC. Terakhir

sistem akan melakukan proses pencetakan hasil perhitungan ketersediaan dan keandalan mesin HMC tersebut.

G. DFD *Level 1* Proses Evaluasi Kinerja Mesin HMC



Gambar 3.14 Proses Evaluasi Kinerja Mesin HMC

Dari gambar 3.14 di atas dapat kita lihat terdapat dua proses yang berbeda yang terjadi pada *level* tersebut. Proses yang pertama yaitu menyusun atau membuat laporan berdasarkan segi perhitungan ekonomis mesin HMC. Proses tersebut diambil dari perhitungan nilai aktiva dan biaya perbaikan yang pernah dialami. Bila biaya yang dialami terlalu besar dan berulang-ulang dalam jangka waktu yang singkat, maka dapat dikategorikan mesin tersebut dapat merugikan perusahaan.

Pada proses yang kedua pada gambar di atas merupakan proses penilaian mesin HMC secara *availability* dan *reliability*. Dari kedua nilai tersebut dapat kita ambil kesimpulan tentang kondisi kinerja mesin, apakah mesin masih bekerja secara produktif atau tidak. Nilai tersebut diambil dari

perhitungan seberapa lama waktu mesin tersebut bekerja dan seberapa lama waktu mesin tersebut mengalami kerusakan. Dari perhitungan tersebut akan diperoleh nilai ketersediaan mesin HMC, apakah mesin tersebut tingkat kehadirannya tinggi atau rendah. Kemudian nilai keandalan mesin dapat dihitung melalui data banyaknya perbaikan dan lamanya waktu mengalami perbaikan dibanding dengan lamanya waktu kerja mesin. Bila mesin sering mengalami kerusakan maka tingkat keandalan mesin akan terlihat jelek pada hasil perhitungan nilai keandalannya.

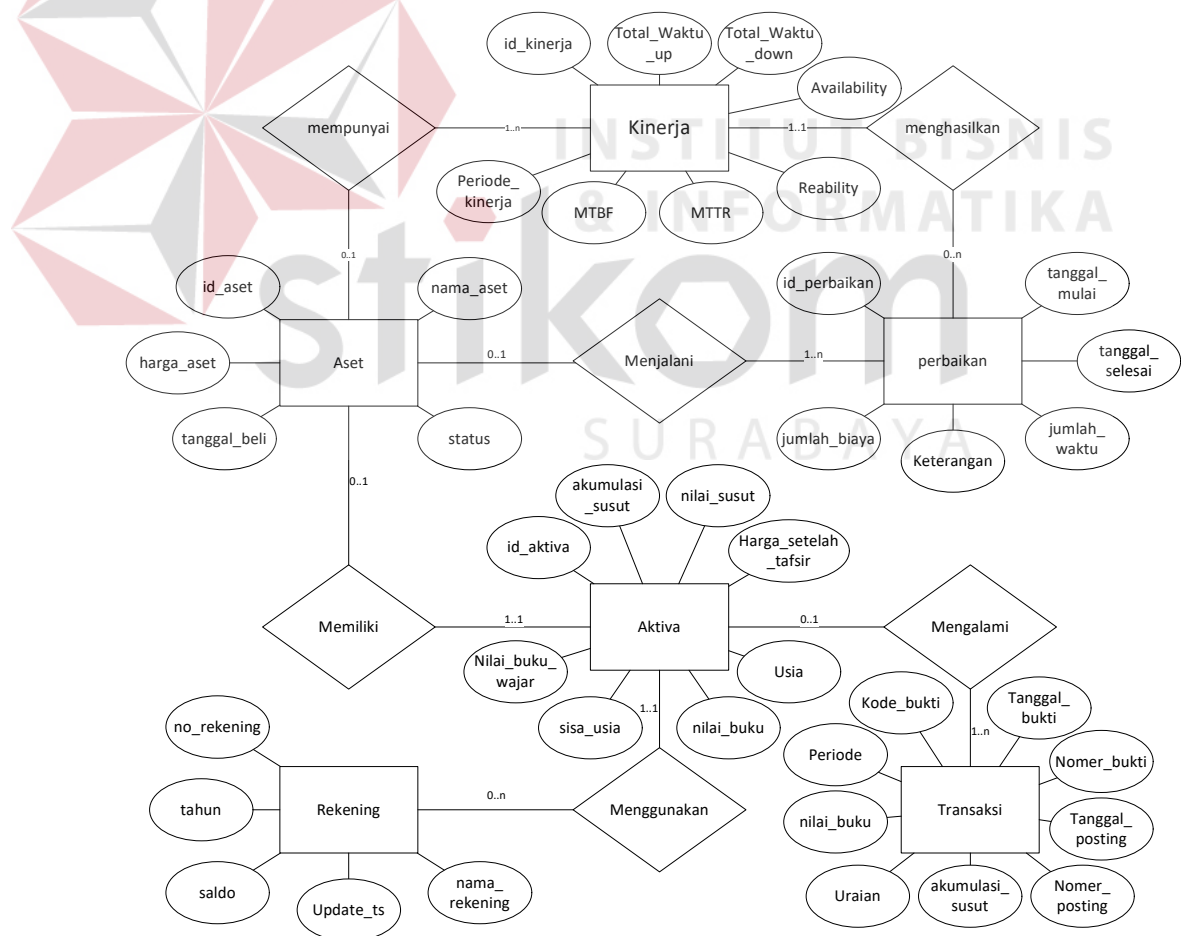
3.2.2 Desain Basis Data

Setelah melalui langkah desain proses yang dimulai dari menentukan *User Requirement* dan *Software Requirement* aplikasinya. Setelah itu menggambarkan *Document Flow* dari sistem yang akan dibuat nantinya dan menjadi *System Flow*. Dari gambaran *System Flow* kemudian dijadikan acuan untuk membuat *Context* dan *DFD Level 0 & DFD Level 1* sebagai alur sistem secara keseluruhan.

Langkah selanjutnya setelah desain proses adalah merancang skema dari *database* yang akan digunakan pada aplikasi. Mendesain *database* dimulai dari pembuatan *Entity Relationship (ER)* diagram. Gunanya adalah untuk memetakan hubungan antar entitas yang akan digunakan pada proses yang ada di aplikasi. Dari ER kita kemudian dapat merancang model data konseptual atau yang lebih dikenal dengan *Conceptual Data Model (CDM)*. CDM digunakan untuk menggambarkan alur data dari masing-masing hubungan antar entitas. Dari CDM maka akan dihasilkan model data fisik atau yang lebih dikenal dengan *Physical Data Model (PDM)*. PDM adalah hasil normalisasi dari CDM dan model data yang digunakan dalam aplikasi

A. Entity Relationship Diagram (ERD)

Entity Relationship Diagram adalah gambar pemetaan relasi antar entitas yang digunakan dalam sistem yang dibangun. Dalam ERD akan terlihat bagaimana kebutuhan antar kedua entitas atau lebih yang saling terhubung. Dalam ERD juga akan terlihat apakah sebuah entitas tersebut bersifat *many* atau *singel* kepada entitas lainnya yang berhubungan dengan entitas tersebut dan begitu sebaliknya. Dari hal tersebut dapat terlihat apakah dari sebuah relasi antar dua entitas dapat memunculkan entitas baru berupa *detil*. Pada umumnya kejadian ini terjadi apabila kedua entitas memiliki relasi yang *many to many*.



Gambar 3 15 Desain Model ERD

Terlihat pada gambar 3.15 di atas terdapat enam entitas yang dipetakan. Diantaranya adalah Aset, Kinerja, Aktiva, Rekening, Perbaikan dan Transaksi. Tergambar di bawah bahwa entitas aset memiliki relasi dengan entitas aktiva dan relasi tersebut bersifat *one to one*. Relasi selanjutnya adalah relasi antara entitas aset dengan entitas perbaikan yang bersifat *one to many*. Kemudian ada relasi antara entitas aktiva dengan entitas rekening yang bersifat *one to many*. Relasi selanjutnya adalah relasi antara entitas aktiva dengan entitas transaksi yang bersifat *one to many*. Terakhir adalah relasi antara entitas perbaikan dengan entitas kinerja yang bersifat *one to many*.

B. Normalisasi

Menurut (Connolly, 2010) normalisasi merupakan suatu teknik untuk menghasilkan sekumpulan hubungan dengan properti yang diinginkan, yang memberikan kebutuhan data terhadap suatu perusahaan. Tujuan dari normalisasi adalah sebagai berikut :

1. Meminimalkan jumlah atribut yang diperlukan untuk mendukung kebutuhan data dari suatu perusahaan.
2. Untuk memperoleh atribut yang bersifat functional dependencies.
3. Untuk menghilangkan data yang bersifat *redundancy* pada tiap atribut.

B.1 Tabel Aset (*Unnormal*)

Pada tabel 3.6 di bawah ditampilkan sebuah tabel yang masih belum normal yang akan digunakan dalam aplikasi evaluasi kinerja mesin HMC. Tabel *unnormal* tersebut berisikan *variable* yang digunakan untuk aplikasi nantinya.

Tabel 3. 6 Tabel Aset (*Unnormal*)

Id aset	Nama aset	Harga aset	Tanggal beli	Status	Id perbaikan	Tanggal mulai	Tanggal selesai	Jumlah waktu
HMC-001	HMC Turbo	40.000.000.000	31/01/2010	Baik	HMC-001[001]	31/01/2010	03/02/2010	84
					HMC-001[002]	20/04/2010	30/04/2010	231
HMC-002	HMC Trak	45.000.000.000	03/11/2011	Baik	HMC-002[001]	13/01/2012	31/01/2012	399
					HMC-002[002]	15/04/2012	20/04/2012	126

B.2 Normalisasi Tabel (1 NF)

Tabel 3. 7 Tabel Aset (1 NF)

<u>Id aset</u>	Nama aset	Harga aset	Tanggal beli	Status	Id perbaikan	Tanggal mulai	Tanggal selesai	Jumlah waktu
HMC-001	HMC Turbo	40.000.000.000	31/01/2010	Baik	HMC-001[001]	31/01/2010	03/02/2010	84
HMC-001	HMC Turbo	40.000.000.000	31/01/2010	Baik	HMC-001[002]	20/04/2010	30/04/2010	231
HMC-002	HMC Trak	45.000.000.000	03/11/2011	Baik	HMC-002[001]	13/01/2012	31/01/2012	399
HMC-002	HMC Trak	45.000.000.000	03/11/2011	Baik	HMC-002[002]	15/04/2012	20/04/2012	126

Tabel 3. 8 Tabel Aktiva (1 NF)

<u>Id aset</u>	Nama aset	Harga aset	Tanggal beli	Status	Id aktiva	Akumulasi susut	Nilai susut	Nilai buku wajar
HMC-001	HMC Turbo	40.000.000.000	31/01/2010	Baik	Atv-001	329.441.764	329.441.764	800.000.000
HMC-001	HMC Turbo	40.000.000.000	31/01/2010	Baik	Atv-001	654.865.492	325.423.728	800.000.000
HMC-002	HMC Trak	45.000.000.000	03/11/2011	Baik	Atv-002	370.588.235	370.588.235	900.000.000
HMC-002	HMC Trak	45.000.000.000	03/11/2011	Baik	Atv-002	736.689.929	366.101.694	900.000.000

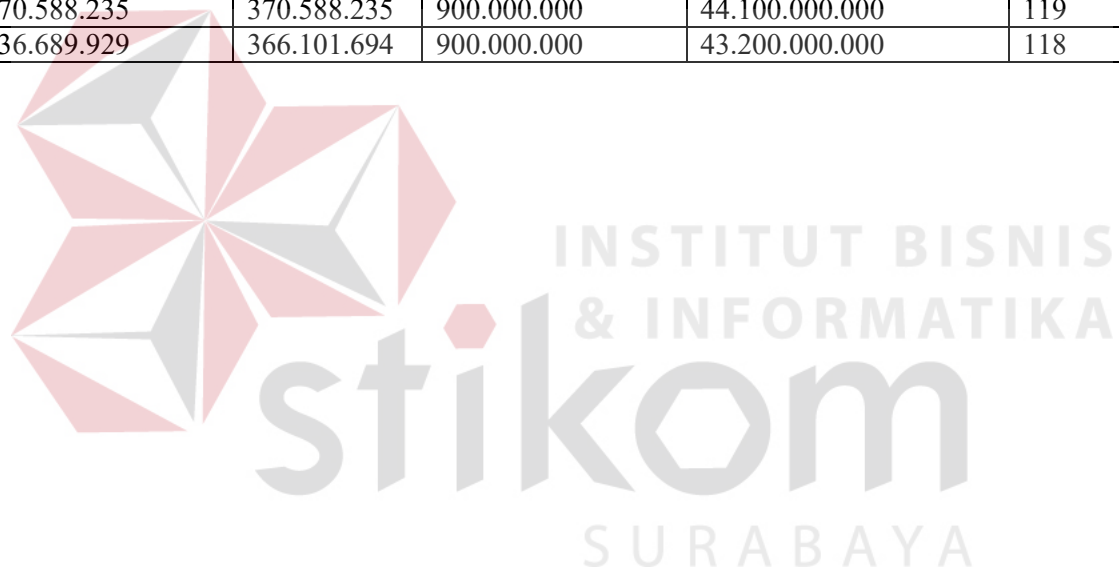
Jumlah biaya	Keterangan	Id kinerja	Periode kinerja	Total waktu up	Total waktu down	Availability	Reliability	MTTR
2.000.000	Rusak	K-001[001]	651	567	84	79%	88%	84
4.500.000	Tali putus	K-001[002]	2520	2205	315	58%	88%	157,5
1.590.000	Engine break	K-002[001]	7665	1491	399	45%	81%	399
2.570.000	Engine break	K-002[002]	7665	3276	525	45%	87%	262,5

Jumlah biaya	Keterangan	Id kinerja	Periode kinerja	Total waktu up	Total waktu down	Availability	Reliability	MTTR
2.000.000	Rusak	K-001[001]	651	567	84	79%	88%	84
4.500.000	Tali putus	K-001[002]	2520	2205	315	58%	88%	157,5
1.590.000	Engine break	K-002[001]	7665	1491	399	45%	81%	399
2.570.000	Engine break	K-002[002]	7665	3276	525	45%	87%	262,5

Harga setelah tafsir	Sisa usia	Usia	Nilai buku	No rekening	Nama rekening	Tahun	Saldo	Update ts	Periode
39.200.000.000	119	1	38.870.558.236	123456	BNI	2009	?????	Admin	Atv-001[001]
38.400.000.000	118	2	37.745.134.571	123456	BNI	2009	?????	Admin	Atv-001[002]
44.100.000.000	119	1	43.729.411.765	789010	BCA	2008	?????	Admin	Atv-002[001]
43.200.000.000	118	2	42.463.310.071	789010	BCA	2008	?????	Admin	Atv-002[002]

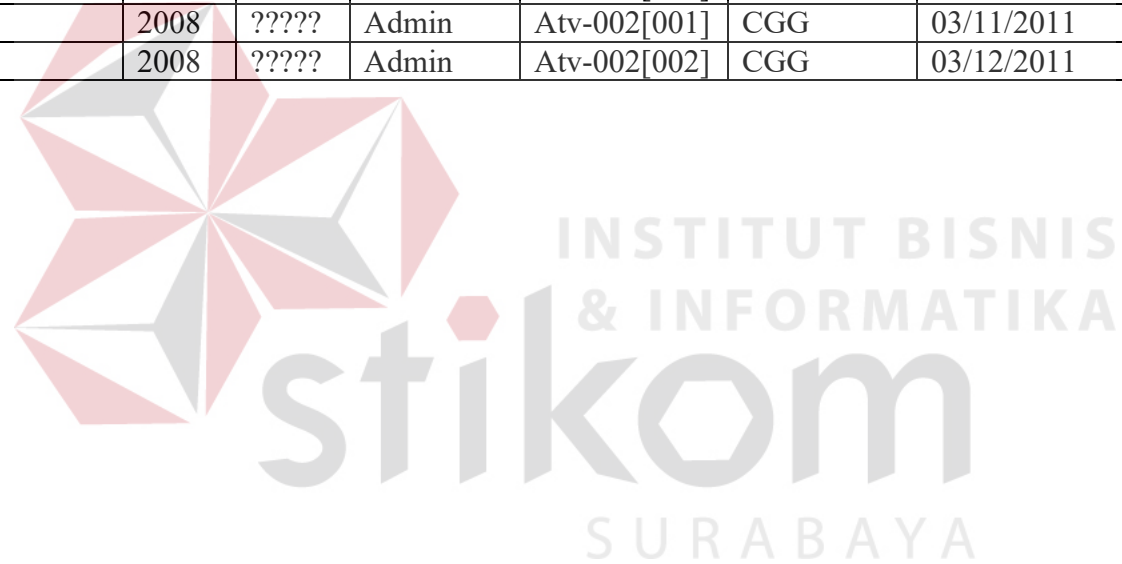
MTBF	Id aktiva	Akumulasi susut	Nilai susut	Nilai buku wajar	Harga setelah tafsir	Sisa usia	Usia	Nilai buku
325,5	Atv-001	329.441.764	329.441.764	800.000.000	39.200.000.000	119	1	38.870.558.236
217		654.865.492	325.423.728	800.000.000	38.400.000.000	118	2	37.745.134.571
325,5	Atv-002	370.588.235	370.588.235	900.000.000	44.100.000.000	119	1	43.729.411.765
2117		736.689.929	366.101.694	900.000.000	43.200.000.000	118	2	42.463.310.071

MTBF
325,5
217
325,5
2117



Kode_bukti	Tanggal_bukti	No_bukti	Tanggal_posting	No_posting	Uraian	Akumulasi_susut	Nilai_buku
CGG	31/02/2010				Periode Bulan : 02/2010	329.441.764	38.870.558.236
CGG	31/03/2010				Periode Bulan : 03/2010	654.865.492	37.745.134.571
CGG	03/11/2011				Periode Bulan : 11/2011	370.588.235	43.729.411.765
CGG	03/12/2011				Periode Bulan : 12/2011	736.689.929	42.463.310.071

No rekening	Nama rekening	Tahun	Saldo	Update ts	Periode	Kode bukti	Tanggal bukti	No bukti	Tanggal posting
123456	BNI	2009	?????	Admin	Atv-001[001]	CGG	31/02/2010		
		2009	?????	Admin	Atv-001[002]	CGG	31/03/2010		
789010	BCA	2008	?????	Admin	Atv-002[001]	CGG	03/11/2011		
		2008	?????	Admin	Atv-002[002]	CGG	03/12/2011		



No posting	Uraian	Akumulasi susut	Nilai buku
	Periode Bulan : 02/2010	329.441.764	38.870.558.236
	Periode Bulan : 03/2010	654.865.492	37.745.134.571
	Periode Bulan : 11/2011	370.588.235	43.729.411.765
	Periode Bulan : 12/2011	736.689.929	42.463.310.071



Pada proses normalisasi 1 NF dibutuhkan beberapa persyaratan yang harus dilewati. Persyaratan tersebut yang pertama adalah agar data tidak terjadi redundan atau ganda. Persyaratan kedua adalah memisahkan masing-masing *variable* berdasarkan kebutuhannya. Pada tabel di atas terlihat bahwa dari tabel *unnormal* untuk memenuhi persyaratan 1 NF tabel *unnormal* dipisahkan berdasarkan *variable* yang saling berkaitan. Di atas dapat terlihat pada hasil 1 NF yang membuat tabel *unnormal* terbagi menjadi dua bagian tabel yaitu tabel 3.7 dan tabel 3.8.

Pada tabel 3.7 di atas tabel dibagi ke dalam golongan *variable* yang memiliki kebutuhan data tentang kinerja mesin. Masing-masing *variable* tersebut adalah *id_aset*(PK), *nama_aset*, *harga_aset*, *tanggal_beli*, *status*, *id_perbaikan*, *tanggal_mulai*, *tanggal_selesai*, *jumlah_waktu*, *jumlah_biaya*, *keterangan*, *id_kinerja*, *periode_kinerja*, *total_waktu_up*, *total_waktu_down*, *availability*, *reliability*, MTTR, dan MTBF. Dari kesemua *variable* tersebut digolongkan ke dalam tabel kinerja karena masing-masing *variable* memiliki peran dalam menentukan nilai kinerja dari sebuah mesin nantinya.

Pada tabel 3.8 di atas tabel dibagi ke dalam golongan *variable* yang memiliki kebutuhan data untuk mendapatkan nilai ekonomis suatu mesin HMC. Tabel tersebut berisikan beberapa *variable*, diantaranya adalah *id_aset*(PK), *nama_aset*, *harga_aset*, *tanggal_beli*, *status*, *id_aktiva*, *akumulasi_susut*, *nilai_susut*, *nilai_buku_wajar*, *harga_setelah_tafsir*, *sisa_usia*, *usia*, *nilai_buku*, *no_rekening*, *nama_rekening*, *tahun*, *saldo*, *update_ts*, *periode*, *kode_bukti*, *tanggal_bukti*, *nomer_bukti*, *tanggal_posting*, *nomer_posting*, *uraian*, *akumulasi_susut*, dan *nilai_buku*.

Variable tersebut dikumpulkan menjadi satu tabel karena memiliki golongan data yang sama yaitu data yang dapat menghasilkan nilai ekonomis sebuah mesin nantinya.

Persyaratan untuk memenuhi 1 NF adalah tidak adanya data yang redundan atau ganda. Pada tabel 3.6 di atas terdapat beberapa contoh redundan data. Redundan data dapat dilihat pada kolom *variable* *id_aset*, *no_rekening* dan lainnya yang mengalami redundan data. Data tersebut kemudian dipisah berdasarkan masing-masing *id* agar tidak terdapat data yang ganda, lihat pada kolom *variable* *id_aset* di gambar 3.7 dan 3.8.

B.3 Normalisasi Tabel (2 NF)

Proses normalisasi selanjutnya setelah melalui proses normalisasi 1 NF adalah melalui proses normalisasi 2 NF. Persyaratan untuk dapat memenuhi kriteria normalisasi 2 NF adalah harus sudah memenuhi persyaratan 1 NF dan dalam satu tabel *variable* harus bergantung penuh pada *primary_key*. Hal tersebut menyebabkan tabel 3.7 dan 3.8 di atas akan mengalami pengecilan. *Variable* yang tidak bergantung pada suatu *primary_key* akan dikeluarkan dan dimasukkan ke dalam golongan *primary_key* yang sesuai dengan pembahasan datanya.

Pada tabel 3.7 sebenarnya terlihat beberapa *variable* yang dapat dijadikan sebuah *primary_key*. Oleh karena itu, kondisi tabel 3.7 yang masih bergantung hanya pada *primary_key* *id_aset* sebenarnya masih belum benar untuk memenuhi persyaratan normalisasi 2 NF. Pada tabel 3.7 terdapat tiga buah *variable* yang dapat dijadikan *primary_key*. *Variable* tersebut antara lain adalah *id_aset*, *id_perbaikan*, dan *id_kinerja*. Lihat tabel 3.9 di bawah.

Tabel 3. 9 *Primary Key* Tabel Kinerja

<u>Id aset</u>	<u>Id perbaikan</u>	<u>Id kinerja</u>
HMC-001	HMC-001[001]	K-001[001]
HMC-001	HMC-001[002]	K-001[002]
HMC-002	HMC-002[001]	K-002[001]
HMC-002	HMC-002[002]	K-002[002]

Setelah itu pada tabel 3.8 sebenarnya juga terlihat beberapa *variable* yang dapat dijadikan sebuah *primary key* selain dari *primary key* awalnya yaitu *id_aset*. Oleh karena itu, kondisi tabel 3.8 yang masih bergantung hanya pada *primary_key* *id_aset* sebenarnya masih belum benar untuk memenuhi persyaratan normalisasi 2 NF. Karena pada tabel 3.8 sebenarnya masih terdapat tiga buah *variable* lain yang dapat dijadikan *primary key*. *Variable* tersebut antara lain adalah *id_aktiva*, *no_rekening*, dan *periode*. Lihat tabel 3.10 di bawah.

Tabel 3. 10 *Primary Key* Tabel Aktiva

<u>Id aset</u>	<u>Id aktiva</u>	<u>No rekening</u>	<u>Periode</u>
HMC-001	Atv-001	123456	Atv-001[001]
HMC-001	Atv-001	123456	Atv-001[002]
HMC-002	Atv-002	789010	Atv-002[001]
HMC-002	Atv-002	789010	Atv-002[002]

B.3.1 Tabel Aset

Pada tabel 3.7 di atas dapat dilihat bahwa kesemua *variable* pada tabel tersebut masih berketergantungan kepada PK yaitu *variable* *id_aset*. Pada kenyataannya *variable* yang bergantung pada *primary_key* *id_aset* hanyalah *variable* *nama_aset*, *harga_aset*, *tanggal_beli*, dan *status*. *Variable* lainnya yang tidak bergantung pada PK *id_aset* akan dikeluarkan dari tabel. Tabel tersebut akan dipisah dan diubah namanya berdasarkan golongannya. Tabel

tersebut menjadi tabel aset yang digunakan sebagai tabel yang menampung data-data aset perusahaan. Tabel 3.11 di bawah merupakan tabel aset yang sudah melalui proses normalisasi 2 NF. Terlihat bahwa semua *variable* yang ada bergantung penuh pada *primary keynya* yaitu *variable* *id_aset*.

Tabel 3. 11 Tabel Aset

Id aset	Nama aset	Harga aset	Tanggal beli	Status
HMC-001	HMC Turbo	40.000.000.000	31/01/2010	Baik
HMC-002	HMC Trak	45.000.000.000	03/11/2011	Baik

B.3.2 Tabel Perbaikan

Pada tabel 3.7 di atas juga terdapat beberapa *variable* yang masih belum berdiri pada *primary keynya* masing-masing. Pada tabel tersebut masih dapat dibagi menjadi dua tabel berbeda setelah golongan data aset dikeluarkan. Pada tabel tersebut masih terdapat *variable* yang dapat dijadikan sebagai *primary key*. *Variable* tersebut adalah *id_perbaikan*.

Dari *variable* *id_perbaikan* dapat kita golongan bahwa tabel tersebut akan berisikan *variable-variable* yang membahas tentang data perbaikan. *Variable* tersebut antara lain adalah *tanggal_mulai*, *tanggal_selesai*, *total_waktu*, *jumlah_biaya*, dan *keterangan*. *Variable-variable* tersebut dapat dikeluarkan dari tabel 3.7 dan membentuk tabel baru, yaitu tabel perbaikan. Tabel tersebut akan digunakan untuk mencatat data perbaikan mesin. Tabel 3.12 di bawah merupakan tabel perbaikan yang sudah melalui proses normalisasi 2 NF. Terlihat bahwa semua *variable* yang ada bergantung penuh pada *primary keynya* yaitu *variable* *id_perbaikan*.

Tabel 3. 12 Tabel Perbaikan

<u>Id perbaikan</u>	Tanggal_Mulai	Tanggal_selesai	Total_Waktu	Jumlah_biaya	Keterangan
HMC-001[001]	31/01/2010	03/02/2010	84	2.000.000	Rusak
HMC-001[002]	20/04/2010	30/04/2010	231	4.500.000	Tali putus
HMC-002[001]	13/01/2012	31/01/2012	399	1.590.000	Engine break
HMC-002[002]	15/04/2012	20/04/2012	126	2.570.000	Engine break

B.3.1 Tabel Kinerja

Dari dua proses pembentukan tabel di atas yaitu tabel aset dan tabel perbaikan tabel 3.7 di atas menjadi kelompok data tersendiri. Pada tabel 3.7 di atas sekarang tersisa golongan data yang sudah satu golongan yang sama. *Variable* tersebut adalah *variable* yang berisikan data mengenai kinerja sebuah mesin.

Tabel 3.7 di atas masih memiliki sebuah *variable* yang dapat digunakan sebagai *primary key* sebuah tabel tersendiri. *Primary key* tersebut adalah *variable* *id_kinerja*. Dari PK tersebut *variabel* yang tersisa dari tabel 3.7 dan cocok dengan golongan data kinerja mesin adalah *variable* *periode_kinerja*, *total_waktu_up*, *total_waktu_down*, *availability*, *reliability*, *MTTR*, dan *MTBF*. Tabel tersebut adalah tabel kinerja yang sebenarnya sehingga dapat dibentuk tabel tersendiri yang berguna untuk menampung hasil perhitungan kinerja mesin. Tabel 3.13 di bawah merupakan tabel kinerja yang sudah melalui proses normalisasi 2 NF berikut terlihat bahwa semua *variable* yang ada bergantung penuh pada *primary key*nya yaitu *variable* *id_kinerja*.

Tabel 3. 13 Tabel Kinerja

<u>Id Kinerja</u>	Periode_Kinerja	Total_Waktu_up	Total_Waktu_down	Availa_bility	Reliab_ility	MTTR	MTBF
K-001[001]	651	567	84	79%	88%	84	325,5
K-001[002]	2520	2205	315	58%	88%	157,5	217
K-002[001]	7665	1491	399	45%	81%	399	325,5
K-002[002]	7665	3276	525	45%	87%	262,5	2117

B.3.4 Tabel Aktiva

Pada tabel 3.8 di atas terdapat data aset yang sebelumnya sudah dipisah menjadi tabel tersendiri. Oleh karena itu *variable* yang sama akan dikeluarkan dari tabel 3.8 dan tersisa beberapa *variable*. Terlihat pada tabel 3.10 bahwa dari *variable* di tabel 3.7 masih terdapat beberapa *primary key*. Salah satu *primary key* tersebut adalah *Id_aktiva*. Dengan masih adanya beberapa *primary key* dalam satu tabel maka harus dipisah berdasarkan golongan *variablenya* masing-masing.

Dari tabel tersebut dapat dikeluarkan sebuah *variable primary key*. *variable* tersebut adalah *id_aktiva*. Dari *id_aktiva* dapat dicari *variable* pada tabel 3.8 yang sesuai dengan golongan *id_aktiva* yang dapat dikelompokkan menjadi tabel aktiva. *Variable* tersebut antara lain adalah *akumulasi_susut*, *nilai_susut*, *nilai_buku_wajar*, *harga_setelah_tafsir*, *sisu_usia*, *usia*, dan *nilai_buku*. *Variable-variable* tersebut dapat dikeluarkan dari tabel 3.8 dan dibentuk sebuah tabel baru yang memiliki kesamaan kelompok data yaitu menjadi tabel aktiva. Tabel tersebut gunanya untuk menampung data nilai aktiva mesin. Tabel 3.14 di bawah merupakan tabel aktiva yang sudah melalui proses normalisasi 2 NF. Terlihat bahwa semua *variable* yang ada bergantung penuh pada *primary keynya* yaitu *variable id_aktiva*.

Tabel 3. 14 Tabel Aktiva

<u>Id_aktiva</u>	Akumulasi_Susut	Nilai_susut	Nilai_buku_wajar	Harga_setelah_tafsir	Sisa_usia	Usia	Nilai_buku
Atv-001	654.865.492	325.42 3.728	800.00 0.000	38.400. 000.000	118	2	37.745. 134.571
Atv-002	736.689.929	366.10 1.694	900.00 0.000	43.200. 000.000	118	2	42.463. 310.071

B.3.5 Tabel Rekening

Setelah kelompok data aset dan aktiva dikeluarkan dari tabel 3.8, tabel tersebut masih belum memenuhi persyaratan 2 NF. Hal tersebut dikarenakan pada tabel tersebut masih terdapat dua kelompok data yang berbeda. Hal tersebut dapat dilihat pada tabel 3.10. Pada tabel tersebut terlihat masih adanya dua *primary* yang berada dalam satu tempat. Salah satunya adalah *primary key no_rekening*. Dilihat dari masih terdapat *variable primary key* yang masih dapat dikeluarkan pada tabel 3.8 maka akan digolongkan kembali berdasarkan kelompok datanya.

Kelompok data yang sama dengan *primary key no_rekening* akan dikeluarkan dari tabel 3.8. *Variable* tersebut antara lain adalah *nama_rekening*, *tahun*, *saldo* dan *update_ts*. Kelompok data tersebut dikeluarkan dari tabel 3.8 dan dikelompokkan menjadi sebuah tabel tersendiri. Tabel tersebut adalah tabel rekening yang digunakan untuk menampung data rekening. Tabel 3.15 di bawah merupakan tabel rekening yang sudah melalui proses normalisasi 2 NF. Terlihat bahwa semua *variable* yang ada bergantung penuh pada *primary keynya* yaitu *variable no_rekening*.

Tabel 3. 15 Tabel Rekening

No rekening	Nama rekening	Tahun	Saldo	Update_ts
123456	BNI	2009	?????	Admin
789010	BCA	2008	?????	Admin

B.3.6 Tabel Transaksi

Setelah semua *variable* yang dapat dikelompokkan tersendiri berdasarkan kelompok datanya saat ini tabel 3.8 sudah menjadi satu kelompok data yang sama. Pada tabel tersebut masih terdapat sebuah *primary key* sehingga dapat dikelompokkan menjadi sebuah tabel tersendiri. *Primary key* tersebut adalah *variable* periode.

Kelompok data yang bergantung pada *primary key* periode antara lain adalah *variable* kode_bukti, tanggal_bukti, nomer_bukti, tanggal_posting, nomer_posting, uraian, akumulasi penyusutan, dan nilai_buku. Kelompok data tersebut dapat dijadikan tabel tersendiri. Tabel tersebut adalah tabel transaksi yang kelompok datanya berfungsi untuk mengelola data transaksi jurnal. Tabel 3,13 di bawah merupakan tabel transaksi yang sudah melalui proses normalisasi 2 NF. Terlihat bahwa semua *variable* yang ada bergantung penuh pada *primary key*nya yaitu *variable* Periode.

Tabel 3. 16 Tabel Transaksi

Periode	Kode_bukti	Tanggal_bukti	Nomer_bukti	Tanggal_posting	Nomer_posting	Uraian	Akumulasi_Penyusutan	Nilai_buku
Atv-001 [001]	CGG	31/02/ 2010				Periode Bulan : 02/2010	329.441. 764	38.870. 558. 236
Atv-001 [002]	CGG	31/03/ 2010				Periode Bulan : 03/2010	654.865. 492	37.745. 134. 571

<u>Periode</u>	<u>Kode_bukti</u>	<u>Tanggal_bukti</u>	<u>Nomer_bukti</u>	<u>Tanggal_posting</u>	<u>Nomer_posting</u>	<u>Uraian</u>	<u>Akumulasi_Penyusutan</u>	<u>Nilai_buku</u>
Atv-002 [001]	CGG	03/11/ 2011				Periode Bulan : 11/2011	370.588. 235	43.729. 411. 765
Atv-002 [002]	CGG	03/12/ 2011				Periode Bulan : 12/2011	736.689. 929	42.463. 310. 071

B.4 Normalisasi Tabel (3 NF)

Proses normalisasi berikutnya adalah proses normalisasi 3 NF. Untuk dapat memenuhi proses tersebut persyaratannya adalah *variable* yang ada tidak bergantung penuh terhadap *key* yang muncul. Pada proses normalisasi 3 NF akan muncul penghubung relasi antar tabel. Hal tersebut ditandai dengan adanya *foreign key* yang menyebabkan adanya *variable* yang tidak bergantung penuh pada *key* yang ada.

Relasi disini menunjukkan bahwa beberapa tabel tersebut saling membutuhkan dan dibutuhkan oleh tabel lainnya. Hal tersebut terlihat diawal bahwa kelompok data diantaranya saling mempengaruhi satu dan yang lainnya. Kelompok tersebut sudah terlihat di awal normalisasi 1 NF yaitu pada tabel 3.7 dan 3.8. Pada tabel 3.7 terlihat beberapa relasi yang terjadi antara lain adalah relasi antara tabel aset dengan tabel perbaikan dan tabel kinerja dengan tabel perbaikan dan tabel aset. Pada tabel 3.8 terlihat relasi antara tabel aset dengan aktiva, dan tabel aktiva dengan rekening serta tabel transaksi.

B.4.1 Tabel Aset

Tabel aset merupakan tabel yang berfungsi sebagai *master* data aset yang ada pada perusahaan. Dalam relasi antar tabel biasanya tabel

master adalah tabel yang dibutuhkan oleh tabel lainnya. Oleh karena itu pada umumnya *primary key* pada tabel aset ini biasanya akan menjadi *foreign key* pada tabel lainnya. Lihat pada gambar 3.16 di bawah :

PK				
Id_aset	Nama_aset	Harga_aset	Tanggal_beli	Status

Gambar 3 16 Gambar Tabel Aset

Pada gambar 3.16 di atas adalah tabel aset. Adapun kolom yang terdapat pada tabel tersebut adalah id_aset sebagai *Primary Key* (PK) pada tabel, nama_aset, harga_aset, tanggal_beli, status. Pada tabel aset tidak tampak adanya *key* lain. Dalam kasus ini tabel aset mengalami status yang tidak memenuhi kondisi 3 NF. Tidak terpenuhinya syarat tersebut antara lain syarat *variable* yang tidak bergantung penuh pada *key*nya. Karena pada tabel tersebut semua *variable* bergantung penuh pada id_aset. Hal ini membuat proses normalisasi akan berhenti hanya pada proses 3 NF.

B.4.2 Tabel Rekening

Tabel rekening merupakan tabel yang digunakan sebagai penampung data rekening perusahaan. Tabel berikut juga merupakan tabel master. Lihat pada gambar 3.17 di bawah :

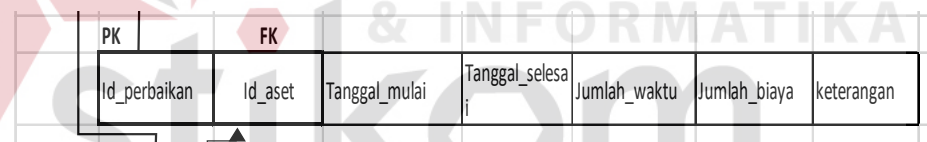
PK				
No_rekening	Nama_rekening	Tahun	Saldo	Update_TS

Gambar 3 17 Gambar Tabel Rekening

Pada gambar 3.17 di atas adalah tabel hasil normalisasi 3 NF rekening. Adapun kolom yang terdapat pada tabel tersebut adalah no_rekening sebagai *Primary Key* (PK) pada tabel, nama_rekening, tahun dan saldo. Dalam tabel tersebut memiliki kesamaan dengan tabel aset yaitu sama-sama tidak sepenuhnya memenuhi persyaratan 3 NF. Hal tersebut dikarenakan kedua tabel tersebut merupakan tabel master yang tidak membutuhkan tabel lainnya. Sehingga terlihat jelas bahwa kesemua *variablenya* bergantung penuh pada *primary key* no_rekening.

B.4.3 Tabel Perbaikan

Tabel perbaikan merupakan tabel yang digunakan sebagai penampung data perbaikan mesin. Pada tabel terlihat bahwa tabel perbaikan membutuhkan tabel aset. Lihat pada gambar 3.18 di bawah :



PK	FK					
Id_perbaikan	Id_aset	Tanggal_mulai	Tanggal_selesai	Jumlah_waktu	Jumlah_biaya	keterangan

Gambar 3 18 Gambar Tabel Perbaikan


Pada gambar 3.18 di atas adalah tabel hasil normalisasi 3 NF perbaikan. Adapun kolom yang terdapat pada tabel tersebut adalah id_perbaikan sebagai *Primary Key* (PK) pada tabel, id_aset sebagai *Foreign Key* (FK), tanggal _mulai, tanggal_selesai, jumlah_waktu, jumlah_biaya dan keterangan. Pada tabel perbaikan terlihat jelas bahwa terdapat sebuah *variable* luar yang masuk kedalam tabel sebagai salah satu *key*. Hal tersebut menyebabkan *variable* lainnya akan secara tidak sepenuhnya bergantung pada *key* tersebut. *Key* tersebut adalah id_aset

sebagai *foreign key* pada tabel perbaikan. Oleh karena itu tabel perbaikan memenuhi kesemua persyaratan 3 NF.

B.4.4 Tabel Kinerja

Tabel Kinerja adalah tabel yang digunakan untuk menampung data perhitungan kinerja mesin HMC. Memiliki relasi terhadap dua buah tabel lainnya. Tabel kinerja membutuhkan tabel aset dan perbaikan. Lihat pada gambar 3.19 di bawah:

PK	FK	FK							
id_kinerja	id_aset	id_perbaikan	Periode_kin erja	Total_waktu_ up	Total_waktu_ down	Availability	Reability	MTTR	MTBF



Gambar 3 19 Gambar Tabel Kinerja

Pada gambar 3.19 di atas terdapat sebuah tabel hasil normalisasi 3 NF kinerja. Pada tabel tersebut terdapat beberapa kolom seperti, kolom *id_kinerja* sebagai PK (*Primary Key*), *id_aset* sebagai FK (*Foreign Key*) yang menghubungkan antara tabel kinerja dengan tabel aset, *id_perbaikan* sebagai FK (*Foreign Key*) yang menghubungkan antara tabel kinerja dengan tabel perbaikan, *periode_kinerja*, *total_waktu_up*, *total_waktu_down*, *MTTR*, *MTBF*, *availability* dan *reliability*. Terlihat jelas bahwa tabel kinerja memenuhi semua persyaratan 3 NF dengan adanya *key* lain yang membuat *variable* dalam tabel tersebut tidak bergantung penuh pada *key* yang ada. *Key* tersebut adalah *id_aset* dan *id_perbaikan* sebagai *foreign key*.

B.4.5 Tabel Aktiva

Tabel aktiva merupakan tabel yang digunakan sebagai penampung data hasil penilaian nilai ekonomis suatu aset. Tabel aktiva

membutuhkan relasi dengan tabel aset dan rekening. Lihat pada gambar 3.20 di bawah :

PK	FK	FK							
Id_aktiva	Id_aset	No_rekening	Akumulasi_susut	nilai_susut	Nilai_buku_wajar	Harga_setelah_tafsir	Sisa_usia	usia	nilai_buku

Gambar 3 20 Gambar Tabel Aktiva

Pada gambar 3.20 di atas adalah tabel hasil normalisasi 3 NF aktiva. Adapun kolom yang terdapat pada tabel tersebut adalah id_aktiva sebagai *Primary Key* (PK) pada tabel, id_aset sebagai *Foreign Key* (FK) yang berfungsi menghubungkan antara tabel aset dengan tabel aktiva, no_rekening sebagai *Foreign Key* (FK) yang menghubungkan antara tabel aktiva dengan tabel rekening, akumulasi_susut, nilai_susut, nilai_buku_wajar, harga_setelah_tafsir, usia(masa manfaat), sisa_usia dan nilai_buku. Adapun tabel aktiva juga memenuhi persyaratan 3 NF. Hal tersebut dapat terlihat dari tabel tersebut juga muncul adanya *key* lain yang membuat *variable* pada tabel aktiva tidak lagi bergantung penuh pada *key* yang ada. *Key* tersebut antara lain adalah id_aset dan no_rekening sebagai *foreign key*.

B.4.6 Tabel Transaksi

Tabel transaksi merupakan tabel yang berfungsi sebagai penampung nilai detil perhitungan nilai aktiva. Tabel transaksi jurnal ini jelas membutuhkan data dari data aktiva. Oleh karena itu aktiva harus terhubung dengan tabel transaksi dengan cara memberikan *primary keynya* sebagai *foreign key* pada tabel transaksi. Lihat pada gambar 3.21 di bawah:

PK	FK								
Periode	id_aktiva	Kode_bukti	Tanggal_bukti	Nomer_bukti	Tanggal_posting	Nomer_posting	Uraian	Akumulasi_susut	Nilai_buku

Gambar 3 21 Gambar Tabel Transaksi

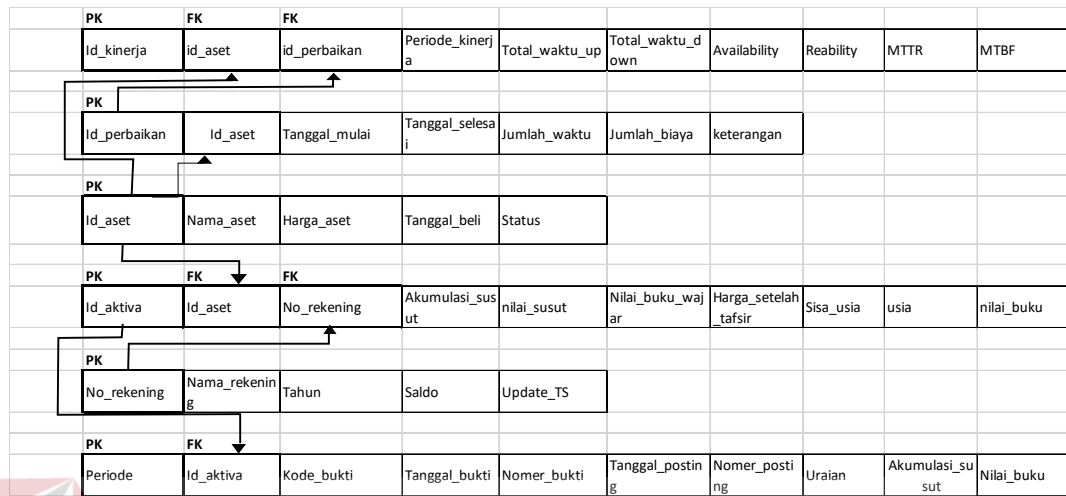
Pada gambar 3.21 di atas adalah tabel hasil normalisasi 3 NF transaksi. Adapun kolom yang terdapat pada tabel tersebut adalah periode sebagai *Primary Key* (PK) pada tabel, id_aktiva sebagai *Foreign Key* (FK) yang menghubungkan antara tabel transaksi dengan tabel aktiva, kode_bukti, tanggal_bukti, nomer_bukti, tanggal_posting, nomer_posting, uraian dan penyusutan. Dari hasil di atas dapat diketahui bahwa tabel transaksi juga memenuhi kesemua persyaratan 3 NF dengan adanya *primary key* milik tabel aktiva yang dijadikan salah satu *key* pada tabel transaksi. Dengan adanya relasi tersebut menyebabkan *variable* yang ada pada tabel transaksi tidak lagi bergantung penuh pada *key* yang ada,

C. *SQL-Tables*

Setelah terdapat desain ER, dan melalui proses normalisasi sehingga memunculkan tabel normal dengan relasinya, maka dapat ditransformasi menjadi *SQL-Tables* yang nantinya akan dapat dilihat perubahan tabel yang dinormalisasikan. *SQL-Tables* merupakan susunan tabel sebelum diterapkan pada PDM (*Physical Data Model*) yang selanjutnya diterapkan menjadi *database* aplikasi.

Pada gambar 3.22 di bawah dapat dilihat bahwa kolom dari masing-masing tabel terdapat perubahan dari model ER sebelumnya. Dapat dilihat bahwa dalam gambar *SQL-Tables* terdapat tambahan FK (*Foreign Key*) pada

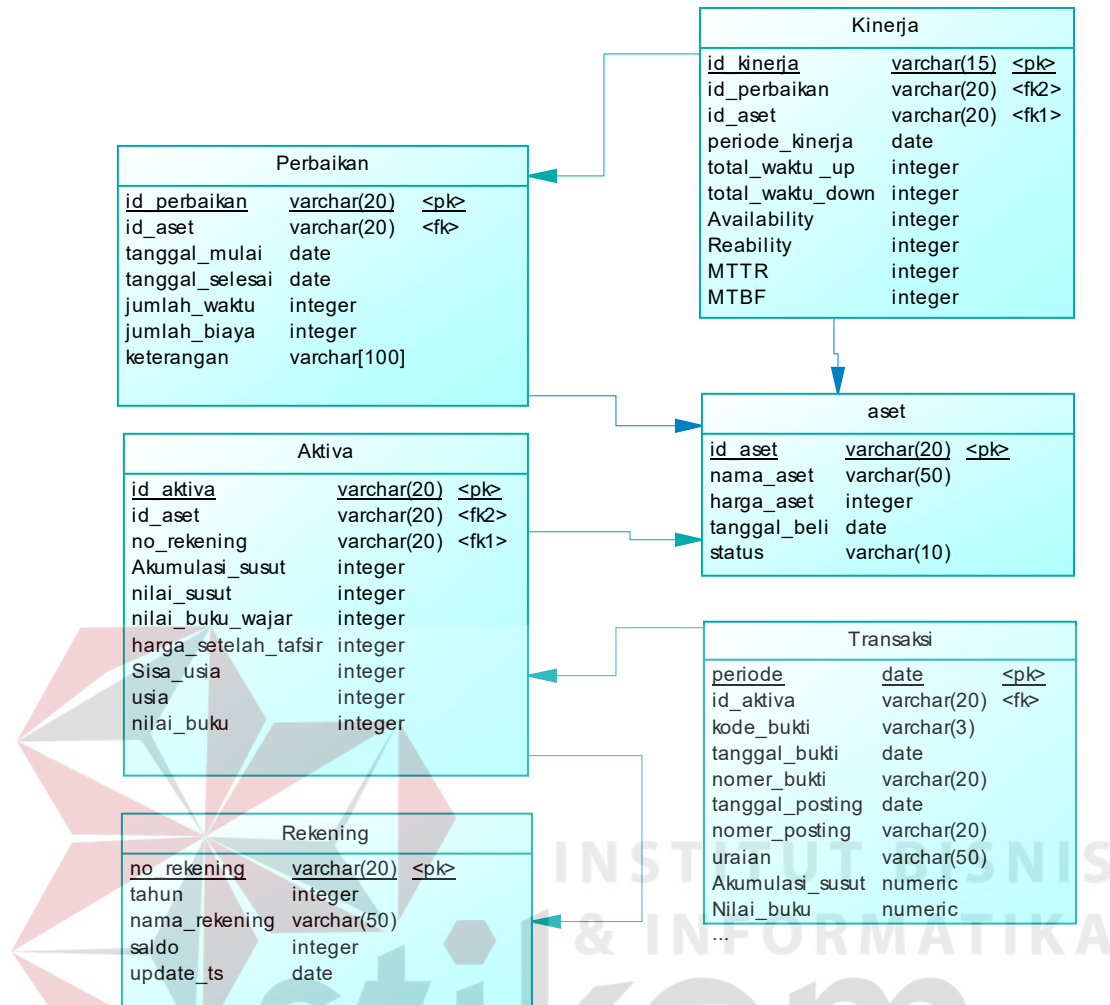
beberapa kolom yang tabelnya saling berkaitan. Hal tersebut menunjukkan relasi antar tabel yang terjadi di dalam *database*.



Gambar 3 22 SQL-Tables

D. *Physical Data Model (PDM)*

Physical Data Model adalah hasil normalisasi dari CDM yang nantinya PDM inilah yang dijadikan acuan desain *database* pada aplikasi. Pada umumnya PDM terdapat tabel tambahan berupa tabel detil apabila pada proses CDM terdapat relasi *many to many*.



Gambar 3 23 Physical Data Model

Pada gambar 3.23 tak terlalu tampak berbeda dari yang ada pada *SQL-Tables*. Karena pada rancangan *database* disini tidak memiliki relasi yang *many to many*. Dengan adanya hal tersebut proses normalisasinya tidak akan muncul tabel baru berupa tabel detail.

E. Struktur Tabel

Struktur tabel di sini akan menjelaskan tentang tabel-tabel yang digunakan dalam aplikasi. Mulai dari nama kolom pada tabel, tipe data, *constraint* dan juga keterangan kegunaan dari kolom tersebut. Berikut adalah daftar tabel tabel tersebut :

E.1 Tabel Aset

Nama Tabel : Aset

Keterangan : Tabel yang digunakan untuk mencatat data mesin.

Tabel 3. 17 Struktur Tabel Aset

Nama Kolom	Tipe Data	Constraint	Keterangan
Id_aset	Varchar (20)	PK	Id pada mesin.
Nama_aset	Varchar (50)	-	Nama mesin.
Harga_aset	int	-	Harga mesin.
Tanggal_beli	date	-	Tanggal beli mesin.
Status	Varchar (10)	-	Status mesin.

E.2 Tabel Rekening

Nama Tabel : Rekening

Keterangan : Tabel yang mencatat data rekening.

Tabel 3. 18 Struktur Tabel Rekening

Nama Kolom	Tipe Data	Constraint	Keterangan
No_rekening	Varchar (20)	PK	Id rekening
Tahun	Int	-	Tahun pembuatan.
Nama_rekening	Varchar (50)	-	Nama rekening.
Saldo	int	-	Jumlah saldo pada rekening.
Update_ts	date	-	Tanggal terakhir update

E.3 Tabel Kinerja

Nama Tabel : Kinerja

Keterangan : Tabel untuk menyimpan data penilaian kinerja mesin.

Tabel 3. 19 Struktur Tabel Kinerja

Nama Kolom	Tipe Data	Constraint	Keterangan
Id_kinerja	Varchar (15)	PK	Id kinerja sebuah mesin pada periode tertentu
Id_aset	Varchar (20)	FK	Id pada mesin.
Total_waktu_up	Int	-	Jumlah waktu mesin dapat bekerja
Total_waktu_down	Int	-	Jumlah waktu mesin tidak dapat bekerja
<i>Availability</i>	Int	-	Ketersediaan mesin
<i>Reliability</i>	int	-	Keandalan mesin
MTTR	Int	-	Nilai MTTR mesin
MTBF	int	-	Nilai MTBF mesin

E.4 Tabel Perbaikan

Nama Tabel : Perbaikan

Keterangan : Tabel yang digunakan untuk mencatat data perbaikan suatu mesin.

Tabel 3. 20 Struktur Tabel Perbaikan

Nama Kolom	Tipe Data	Constraint	Keterangan
Id_perbaikan	Varchar (20)	PK	Id pada setiap perbaikan yang dicatat.
Id_aset	Varchar (20)	FK	Id pada mesin yang mengalami perbaikan.
Id_kinerja	Varchar (15)	FK	Id kinerja suatu mesin.
Tanggal_mulai	Date	-	Tanggal awal mesin rusak.
Tanggal_selesai	Date	-	Tanggal mesin selesai diperbaiki.

Nama Kolom	Tipe Data	Constraint	Keterangan
Jumlah_waktu	int	-	Jumlah waktu perbaikan mesin.
Jumlah_biaya	int	-	Jumlah biaya perbaikan mesin.
Keterangan	Varchar (100)	-	Keterangan kejadian perbaikan

E.5 Tabel Aktiva

Nama Tabel : Aktiva

Keterangan : Tabel yang mencatat nilai aktiva mesin.

Tabel 3. 21 Struktur Tabel Aktiva

Nama Kolom	Tipe Data	Constraint	Keterangan
Id_aktva	Varchar (20)	PK	Id aktiva mesin.
Id_aset	Varchar (20)	FK	Id pada mesin.
No_rekening	Varchar (20)	FK	Id Rekening.
Persen_susut	int	-	Persentase susut pada nilai ekonomis mesin.
Usia	int	-	Perkiraan usia mesin
Sisa_usia	Int	-	Sisa usia dari perkiraan usia.
Nilai_susut	int	-	Nilai susut mesin.
Akumulasi_susut	int	-	Jumlah total nilai susut suatu mesin selama beberapa waktu.
Nilai_buku_wajar	int	-	Nilai pengurangan harga pokok mesin
Harga_setelah_tafsir	int	-	Nilai harga pokok setelah dikurangi nilai buku wajar

Nama Kolom	Tipe Data	Constraint	Keterangan
Nilai_Buku	Int	-	Nilai Harga mesin setelah disusutkan

E.6 Tabel Transaksi

Nama Tabel : Transaksi

Keterangan : Tabel yang digunakan untuk mencatat hasil perhitungan aktiva.

Tabel 3. 22 Struktur Tabel Transaksi

Nama Kolom	Tipe Data	Constraint	Keterangan
Id_transaksi	Varchar (25)	PK	Id_transaksi
Periode	Date	-	Periode transaksi
Id_aktiva	Varchar (20)	FK	Id_aktiva yang mengalami transaksi.
Kode_bukti	Varchar (3)	-	Kode Transaksi.
Tanggal_bukti	Date	-	Tanggal bukti.
Nomer_bukti	Varchar (20)	-	Nomer bukti.
Tanggal_posting	Date	-	Tanggal posting transaksi.
Nomer_posting	Varchar (20)	-	Nomer posting
Uraian	Varchar (50)	-	Penjelasan transaksi.
Akumulasi_susut	numeric	-	Jumlah penyusutan terakhir.
Nilai_buku	numeric	-	Nilai buku terakhir

3.2.3 Perancangan Antar Muka

Setelah merancang *context diagram*, *DFD level* dan *entity relationship diagram* dan PDM maka dapat diperoleh struktur tabel. Setelah struktur tabel dibuat maka proses selanjutnya yaitu perancangan *interface*. Perancangan *interface* berfungsi agar pengguna dapat mengetahui formulir yang digunakan sebagai *input*

untuk dimasukkan pada aplikasi dan *output* yang dihasilkan oleh aplikasi. Disamping itu, pengguna dapat dengan mudah memahami alur sistem yang berjalan pada aplikasi yang berbasis *web*. Pada pembuatan rancangan *interface* ini dibagi menjadi dua bagian yaitu membuat desain *input output* dari aplikasi dan membuat *user interface* dari aplikasi.

A. Rancangan *Input Output*

Desain *input output* adalah rancangan *form* yang digunakan untuk membantu alur berjalannya sistem dengan cara memberikan antarmuka kepada pengguna secara nyata berupa dokumen kertas. Desain *input* merupakan dokumen yang digunakan oleh pengguna sebagai media sementara yang nantinya akan disalin kedalam aplikasi yang ada. Desain *output* yaitu dokumen yang dihasilkan oleh aplikasi, misalnya nota pembayaran, laporan, dan lain-lain.

A.1 Rancangan *Input*

Desain *input* merupakan dokumen yang digunakan oleh bagian keuangan sebagai *input* data aset. Dokumen ini berisi tentang data aset yang baru dimiliki perusahaan..

A.2 Rancangan *Output*

Desain *output* laporan merupakan desain *output* yang digunakan sebagai rancangan *output* aktiva aset.

B. *User Interface*

Pada sub bab ini menjelaskan tentang tampilan antar muka pengguna dengan aplikasi. *User interface* merupakan tampilan yang dibuat oleh peneliti

sebagai acuan bagi pengguna untuk mengetahui isi *vield* yang akan digunakan pada aplikasi. Tampilan ini hampir sama dengan *form* yang akan dibuat pada aplikasi. Aplikasi dibuat berbasis *website* sehingga tampilan tersebut dapat digunakan oleh semua pengguna.

B.1 *Form Login*

Gambar 3.24 di bawah merupakan gambaran *interface* aplikasi nantinya pada bagian halaman *login*. Halaman ini nantinya yang akan menjadi halaman masuk hak akses *user* untuk dapat menggunakan aplikasi ini.

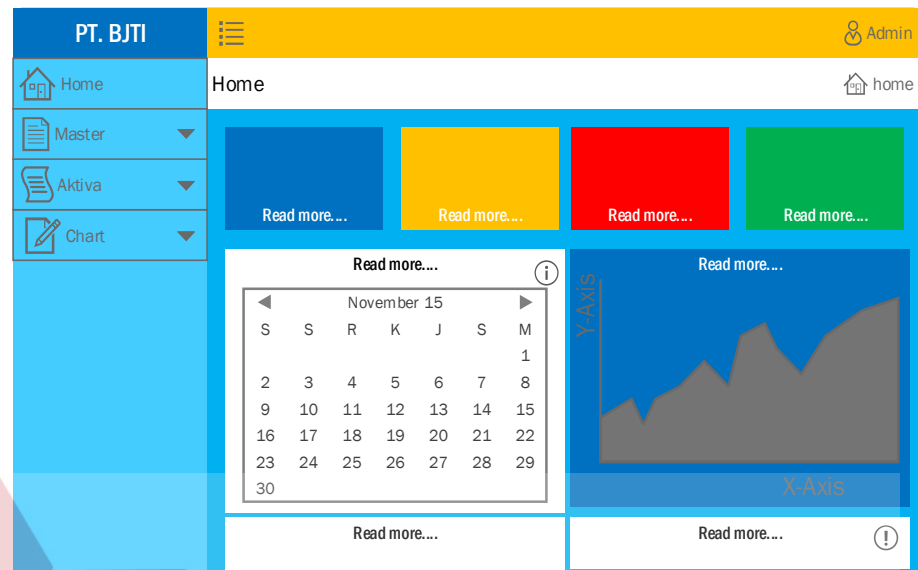


Gambar 3 24 *User Interface Login*

B.2 *Form Dashboard*

Gambar 3.25 di bawah merupakan gambaran *interface* aplikasi nantinya pada bagian halaman *dashboard*. Halaman ini nantinya yang akan menjadi halaman depan atau *home* pada aplikasi. Fungsi dari halaman ini nantinya adalah untuk menunjukkan informasi secara garis besar kepada *user* tentang kondisi aset dari berbagai penilaian seperti

kinerja aset, ketersediaan aset bahkan dari segi ekonomi aset yang akan ditampilkan dalam bentuk grafik agar mudah dibaca oleh pihak *user*.



Gambar 3 25 *User Interface Dashboard*

B.3 *Form Master Aset*

Gambar 3.26 di bawah merupakan gambaran *interface* aplikasi nantinya pada bagian halaman *master* aset. Halaman ini nantinya yang akan menjadi halaman untuk menginputkan data aset.

Gambar 3 26 *User Interface Master Aset*

B.4 *Form Transaksi Aktiva*

Gambar 3.27 di bawah merupakan gambaran *interface* aplikasi nantinya pada bagian halaman transaksi aktiva. Halaman ini nantinya yang akan menjadi halaman yang digunakan untuk memproses nilai aktiva setiap periodenya. Halaman ini digunakan untuk menjalankan transaksi aktiva dari keseluruhan aktiva yang dimiliki pada periode yang sesuai dengan *inputan user*. Halaman ini akan mempermudah tugas *user* untuk memproses penyusutan nilai aktiva setiap periodenya. Nantinya *user* tidak perlu lagi melakukan perhitungan transaksi tersebut satu persatu lagi. Karena dengan halaman ini proses tersebut akan selesai dengan satu klik proses saja.

Gambar 3 27 *User Interface* Transaksi Aktiva

B.5 *Form Master Rekening*

Gambar 3.28 di bawah merupakan gambaran *interface* aplikasi nantinya pada bagian halaman *master* rekening. Halaman ini nantinya

yang akan menjadi halaman yang digunakan *user* untuk menginputkan data rekening ke dalam *database*.

Gambar 3 28 *User Interface Master Rekening*

B.6 *Form Master Perbaikan*

Gambar 3.29 di bawah merupakan gambaran *interface* aplikasi nantinya pada bagian halaman perbaikan aset. Halaman ini nantinya yang akan menjadi halaman yang digunakan untuk proses pencatatan perbaikan suatu aset. Nantinya pada halaman ini akan terdapat proses perhitungan nilai kinerja suatu aset.

Gambar 3 29 *User Interface Master Perbaikan*

B.7 *Form Export Excel*

Gambar 3.30 di bawah merupakan gambaran aplikasi nantinya pada bagian halaman *export excel*. Halaman ini nantinya yang akan menjadi halaman yang berfungsi sebagai halaman yang dapat memilih aset dan periodenya yang akan di*export* ke dalam format *excel*.

Gambar 3 30 *User Interface Export Excel*

B.8 Form Aktiva

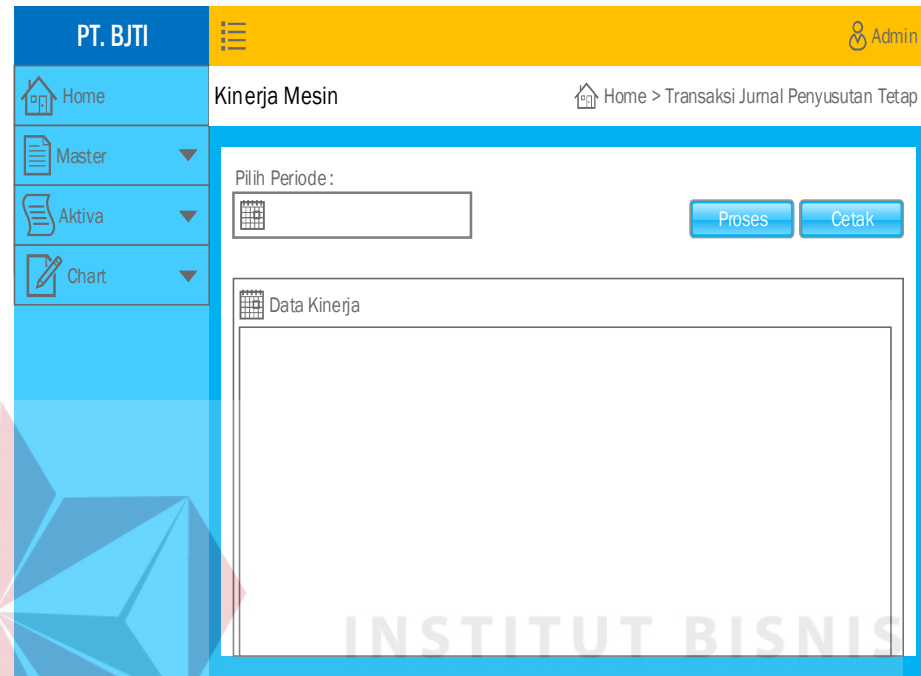
Gambar 3.31 di bawah merupakan gambaran aplikasi nantinya pada bagian halaman *master* aktiva. Halaman ini nantinya yang akan menjadi halaman yang berfungsi sebagai halaman *input* data-data aktiva dari sebuah aset. Dari *inputan* yang terdapat pada gambar di bawah nantinya akan menjadi acuan dalam mencari nilai-nilai yang dapat digunakan untuk menentukan nilai aktiva sebuah aset.

Gambar 3 31 *User Interface Aktiva*

B.9 Form Kinerja

Gambar 3.32 di bawah merupakan gambaran aplikasi nantinya pada bagian halaman kinerja. Halaman ini nantinya yang akan menjadi halaman yang berfungsi sebagai halaman yang dapat memonitor

kondisi kinerja sebuah aset yang telah tercatat. Halaman ini nantinya dapat melihat secara detil masing-masing aset dalam bentuk grafik dan fitur lainnya.



Gambar 3.32 User Interface Login

B.10 Form Dashboard Evaluasi

Gambar 3.33 di bawah merupakan gambaran aplikasi nantinya pada bagian halaman evaluasi. Halaman ini nantinya yang akan menjadi halaman berfungsi sebagai halaman yang dapat menunjukkan kondisi suatu aset berdasarkan beberapa penilaian. Penilaian tersebut antara lain diberi dari segi penilaian kinerja dan penilaian nilai ekonomis aset tersebut. Nantinya pada halaman ini akan terdapat fitur persentase kondisi kinerja suatu aset dengan perbandingan keterangan sesuai standar yang ada. Terdapat fitur penilaian ekonomis yang dapat menunjukkan bentuk pengeluaran perawatan aset tersebut sudah masuk

kategori merugikan atau tidak ditinjau dari biaya perawatan dibandingkan dengan perkiraan penghasilan kinerja mesin tersebut.

Gambar 3 33 *User Interface* Evaluasi

B.11 Laporan Kinerja

Gambar 3.34 di bawah merupakan gambaran laporan nantinya pada bagian laporan kinerja. Laporan ini nantinya yang akan menjadi laporan yang digunakan untuk menunjukkan kondisi suatu aset.

PT BJTI		Laporan Ketersediaan dan keandalan Kinerja Mesin HMC			
	Availability	Standart 1	Reability	Standart 2	Penilaian
Mesin 1					
Mesin 2					
Mesin 3					
Mesin 4					
Mesin 5					
Mesin 6					

Gambar 3 34 *User Interface* Laporan Kinerja

B.12 Laporan Aktiva

Gambar 3.35, 3.36, dan 3.37 di bawah merupakan gambaran laporan aktiva nantinya pada bagian laporan aktiva. Laporan ini nantinya yang akan menjadi laporan yang digunakan untuk menunjukkan kondisi nilai aktiva suatu aset.

NO	KODE REK	NAMA AKTIVA	UNIT	MASA	MASA	NILAI PEROLEHAN	TAKSIRAN NILAI BUKU WAJAR	NILAI PEROLEHAN BETAH TAKSIRAN	NILAI	SISA MASA MANFAAT
				PEROLEHAN	MANFAAT				PENYUSUTAN PER BULAN	

Gambar 3 35 *User Interface* Laporan Aktiva (A)

PENYUSUTAN												
Jan-13	Feb-13	Mar-13	Apr-13	Mei-13	Jun-13	Jul-13	Agu-13	Sep-13	Okt-13	Nov-13	Des-13	TOTAL TAHUN 2013

Gambar 3 36 *User Interface* Laporan Aktiva (B)

AKUMULASI PENYUSUTAN		NILAI BUKU	SISA MANFAAT	KODE AKUN PUSAT	KODE AKUN JENIS
S/D TAHUN SEBELUMNYA	S/D TAHUN BERJALAN				

Gambar 3 37 *User Interface* Laporan Aktiva (C)

3.2.4 Perencanaan Uji Coba Sistem

Setelah melakukan perancangan dan desain sistem informasi evaluasi kinerja mesin, maka tahap selanjutnya adalah melakukan perencanaan atas uji coba sistem informasi yang akan dilakukan setelah sistem informasi selesai dibangun. Uji coba ini dilakukan untuk mengetahui apakah sistem informasi yang dibuat telah sesuai dengan kebutuhan pihak PT BJTI. Uji coba ini dilakukan dengan subjek uji coba perorangan dan juga dilakukan uji coba dengan *black box testing*.

A. Perencanaan Uji Coba Subjek Perorangan

Perencanaan uji coba subjek perorangan ini dilakukan agar sistem informasi yang dibuat telah sesuai dengan kebutuhan pengguna dan telah dapat diterima oleh pengguna. Subjek uji coba yang diambil adalah pada PT BJTI Surabaya, perencanaan uji coba dengan subjek perorangan ini secara lebih jelasnya dapat dilihat pada Tabel 3.20.

Tabel 3. 23 Rencana Uji Coba Subjek Perorangan

No	Subjek	Rencana Testing	Hasil Yang Diharapkan
1	Manager Operasional (1 Orang)	Manager operasional PT BJTI melakukan uji coba aplikasi evaluasi kinerja mesin (melakukan pengecekan dan validasi bahwa aplikasi telah sesuai dengan yang diinginkan dan telah dapat membantu menyelesaikan permasalahan).	Sistem telah sesuai dengan apa yang diharapkan dan mampu memberikan hasil evaluasi kinerja mesin dengan baik.
2	Bag, Keuangan	Bag. Keuangan melakukan uji coba penilaian ekonomis suatu mesin dengan menggunakan data yang sudah ada, apakah terdapat hasil yang tidak sesuai atau aplikasi sudah bekerja dengan baik.	Sistem telah sesuai dengan apa yang diharapkan dan mampu memberikan perhitungan aktiva yang akurat dan lebih cepat.

No	Subjek	Rencana Testing	Hasil Yang Diharapkan
3	Bag, Mekanik	Bag. Mekanik melakukan uji coba penilaian kinerja suatu mesin dengan menggunakan data yang sudah ada, apakah terdapat hasil yang tidak sesuai atau aplikasi sudah bekerja dengan baik. Dengan error yang masih dibawah batas.	Sistem telah sesuai dengan apa yang diharapkan dan mampu memberikan perhitungan kinerja mesin yang akurat dan lebih cepat.

B. Perencanaan Uji Coba *System*

Setelah melakukan rancang bangun sistem informasi evaluasi kinerja mesin, maka harus dilakukan uji coba untuk menguji fungsionalitas dari sistem informasi yang telah dibangun.

B.1 Uji Coba Halaman *Login*

Rancangan berikut berfungsi untuk mengetahui kesesuaian *login* dari masing-masing anggota berdasarkan *username* dan *password* yang telah ditentukan sebelumnya. Uji coba ini juga berfungsi untuk mengetahui kesesuaian aplikasi dengan harapan yang akan dicapai. Rancangan uji coba form login dapat dilihat pada tabel 3.21.

Tabel 3. 24 Rencana Uji Coba *Form Login*

No	Tujuan	Masukan	Keluaran yang diharapkan
1	Respon saat <i>username</i> salah	Data sembarang	Menampilkan bahwa data yang dimasukkan tidak benar
2	Respon saat <i>password</i> salah	data sembarang	Menampilkan bahwa data yang dimasukkan tidak benar
3	Respon saat <i>username</i> dan <i>password</i> benar	<i>Username</i> dan <i>password</i>	Membuka halaman <i>dashboard</i> sesuai jabatan

B.2 Uji Coba Halaman Perbaikan

Rancangan uji coba *form* perbaikan berfungsi untuk mengetahui fungsi aplikasi berjalan sesuai dengan harapan yang akan dicapai. Rancangan uji coba *form* perbaikan dapat dilihat pada tabel 3.22.

Tabel 3. 25 Rencana Uji Coba *Form* Perbaikan

No	Tujuan	Masukan	Keluaran yang diharapkan
1	Filter data kondisi mesin berjalan dengan baik pada pilihan awal perbaikan.	<i>Combo box</i> mesin	Data mesin yang hanya memiliki status baik.
2	Mengetahui fungsi awal perbaikan berjalan dengan baik.	Data awal perbaikan	Menyimpan data awal perbaikan mesin yang telah dipilih. Status mesin menjadi perbaikan.
3	Filter data kondisi mesin berjalan dengan baik pada pilihan selesai perbaikan.	<i>Combo box</i> mesin	Data mesin yang hanya memiliki status perbaikan.
4	Mengetahui fungsi selesai perbaikan berjalan dengan baik	Data perbaikan	Mengupdate data perbaikan dan membuat status mesin dalam keadan baik.

B.3 Uji Coba Halaman Kerja

Rancangan uji coba *form* kerja berfungsi untuk mengetahui fungsi aplikasi berjalan sesuai dengan harapan yang akan dicapai. Rancangan uji coba *form* kerja dapat dilihat pada tabel 3.23.

Tabel 3. 26 Rencana Uji Coba *Form Kerja*

No	Tujuan	Masukan	Keluaran yang diharapkan
1	Filter data status mesin berjalan dengan baik pada pilihan kerja.	Tanggal, id_mesin	Data mesin yang dapat dipilih hanya yang berstatus != bekerja, pada tanggal tersebut.

B.4 Uji Coba Halaman *Dashboard*

Rancangan uji coba *form dashboard* berfungsi untuk mengetahui fungsi aplikasi berjalan sesuai dengan harapan yang akan dicapai. Rancangan uji coba *form dashboard* dapat dilihat pada tabel 3.24.

Tabel 3. 27 Rencana Uji Coba *Form Dashboard*

No	Tujuan	Masukan	Keluaran yang diharapkan
1	Filter data kondisi mesin berjalan dengan baik pada pilihan awal perbaikan.	<i>Combo box</i> mesin	Data mesin yang hanya memiliki status baik.
2	Mengetahui fungsi awal perbaikan berjalan dengan baik.	Data awal perbaikan	Menyimpan data awal perbaikan mesin yang telah dipilih. Status mesin menjadi perbaikan.
3	Filter data kondisi mesin berjalan dengan baik pada pilihan selesai perbaikan.	<i>Combo box</i> mesin	Data mesin yang hanya memiliki status perbaikan.
4	Mengetahui fungsi selesai perbaikan berjalan dengan baik	Data perbaikan	<i>Mengupdate</i> data perbaikan dan membuat status mesin dalam keadan baik.