

BAB V

DESAIN, IMPLEMENTASI DAN TESTING

Pada bab ini akan dijelaskan mengenai perancangan SIMIN fase ketiga hingga kelima, yaitu *design*, *implementation*, dan pengujian. Fase pengujian meliputi uji verifikasi, uji validasi, dan uji *prototype*. Setelah perancangan sistem selesai dibuat, maka akan dihasilkan sebuah *prototype* SIMIN JTI.

5.1 Desain Sistem

Desain sistem adalah fase ketiga dari lima fase dalam siklus hidup pengembangan sistem. Tujuan dari desain sistem adalah untuk menciptakan model fisik dari sistem yang memenuhi persyaratan desain yang ditetapkan dalam fase analisa sebelumnya. Pada fase analisa sistem sudah dikembangkan sebuah model logis dari sistem yang baru. Pada fase ini akan dirancang desain fisik yang akan memenuhi spesifikasi yang dijelaskan dalam *system requirements document*. Fase desain sistem mencakup desain database, desain *user interface*, dan desain algoritma sistem.

5.1.1 Desain Database

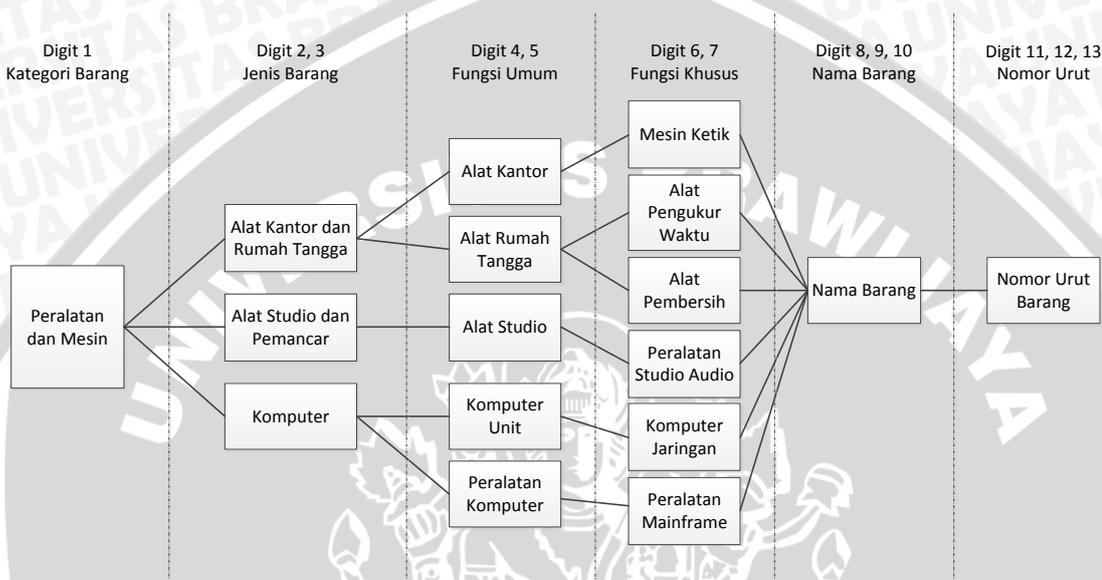
Untuk fase desain *database* secara umum, yang perlu dilakukan oleh analis adalah mengidentifikasi terlebih dahulu file-file yang diperlukan oleh sistem informasi. Langkah-langkah untuk mendesain database adalah sebagai berikut:

5.1.1.1 Peran *Group Technology* Dalam Desain Database

Group Technology (GT) merupakan suatu metode yang digunakan untuk menyelesaikan masalah - masalah yang mempunyai kemiripan yaitu dengan cara mengelompokkan masalah - masalah yang mirip menjadi satu sel sehingga pemecahan masalah tersebut dapat menghemat waktu dan upaya (Hadiguna, 2008). Konsep GT tersebut digunakan untuk mengklasifikasikan inventaris yang memiliki ciri yang sama.

Klasifikasi didefinisikan sebagai sebuah proses dari pengelompokkan bagian – bagian menjadi sebuah grup atau famili berdasarkan suatu aturan dan prinsip tertentu. Aplikasi klasifikasi konsep GT dalam SIMIN dilakukan dengan mengklasifikasikan inventaris berdasarkan kategori inventaris, jenis inventaris, fungsi umum, fungsi khusus, nama barang dan nomor barang seperti ditunjukkan pada gambar 5.1.

Selama ini GT dikembangkan untuk mengelompokkan *part* ataupun desain sebuah benda dengan cara mengelompokkannya sesuai kesamaan melalui kodefikasi. Dalam hal ini inventaris yang memiliki jenis yang sama akan dikelompokkan dengan memberikan kode inventaris pada inventaris tersebut. Kemudian inventaris yang memiliki *atribut* tahun juga akan diklasifikasikan berbeda untuk memberikan nomor yang berbeda – beda. Struktur *hybrid* dapat dilihat pada Gambar 5.1.



Gambar 5.1 Konsep *Hybrid* Pengelompokan Inventaris

Metode klasifikasi dan kodefikasi SIMIN menggunakan GT dengan sistem kodefikasi *hybrid* karena dapat dengan mudah dipahami dan diterapkan di JTI untuk memasukkan data sesuai dengan kondisi yang ada. Dalam melakukan *entry data* user juga akan dimudahkan dengan kodefikasi GT.

Sebelum membuat satu tabel data maka diperlukan suatu parameter yang digunakan dalam pengkodean dengan menggunakan konsep pengkodean *hybrid*. Berikut parameter yang digunakan pada pengkodean:

1. Kategori Inventaris (digit 1)

Kategori inventaris ini merupakan pengelompokan berdasarkan kategori yang dimiliki inventaris tersebut. Kategori inventaris dilambangkan dengan satu digit angka.

2. Jenis Inventaris (digit 2-3)

Jenis inventaris berisi 2 digit angka yang menjelaskan informasi jenis inventaris. Kode ini akan memudahkan pencarian inventaris berdasarkan jenis.

3. Fungsi umum (digit 4-5)

Fungsi umum ini merupakan pengelompokan inventaris berdasarkan fungsi umum yang dimiliki oleh inventaris tersebut. Contoh: fungsi umum inventaris berupa “alat rumah tangga” diwakili dengan kode 3.05.01.

4. Fungsi Khusus (digit 6-7)

Fungsi khusus ini merupakan kode untuk mengidentifikasi lebih detail mengenai informasi inventaris yang dimiliki. Fungsi khusus ini merupakan penjabaran dari fungsi umum.

5. Nama Inventaris (digit 8-10)

Nama inventaris merupakan kodifikasi terkait nama inventaris secara langsung. Nama inventaris ini diwakili dengan 3 angka untuk mengidentifikasi fungsinya.

6. Nomor Inventaris (digit 11-13)

Nomor inventaris merupakan kodifikasi yang berhubungan dengan nama inventaris. Nomor inventaris ini untuk memberikan spesifikasi lebih lanjut mengenai keterangan nama inventaris. Nomor inventaris diwakili dengan 3 angka untuk mengantisipasi jumlah barang yang banyak.

Untuk menganalisa kelompok yang sudah dibentuk maka digunakan tabel data yang memuat pengelompokan inventaris berdasarkan parameter diatas. Tabel 5.1 diharapkan mempermudah dalam menggambarkan penggolongan inventaris.

Tabel 5.1 Tabel Parameter dan *Attribute*

Parameter	Entity	Atribut	Kode	Keterangan
Hirarki Inventaris	Kategori Inventaris	Peralatan dan Mesin	3	Digit 1
	Jenis Inventaris	Alat kantor dan rumah tangga	05	Digit 2-3
		Alat studio dan pemancar	06	
		Komputer	10	
	Fungsi Umum	Alat kantor	01	Digit 4-5
		Alat rumah tangga	02	
		Alat Studio	01	
		Komputer unit	01	
		Peralatan komputer	02	
	Fungsi Khusus	Mesin ketik	01	Digit 6-7
		Alat pengukur waktu	01	
		Alat pembersih	02	
		Peralatan studio	01	
Komputer jaringan		01		
Peralatan mainframe		01		
Atribut Inventaris	Nama Inventaris	Nama inventaris yang terkait	001	Digit 8-10
	Nomor Inventaris	Nomor inventaris	001	Digit 11-13

Sehingga bila digambarkan secara tabel, maka contoh kodifikasi inventaris adalah sebagai berikut:

3	0	5	0	1	0	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---

Gambar 5.2 Contoh Kodifikasi Inventaris

Kode inventaris yang disebutkan pada gambar 5.2 adalah inventaris peralatan dan mesin, jenis alat kantor dan rumah tangga dengan detail alat kantor yang memiliki fungsi sebagai mesin ketik, nama inventaris tersebut dilambangkan dengan kode inventaris pertama dan memiliki jumlah inventaris sebanyak 1.

5.1.1.2 Logical Model

Logical model dari sistem informasi lebih menjelaskan kepada *user* bagaimana nantinya fungsi-fungsi di sistem informasi secara logika akan bekerja. Logical model dapat digambar dengan menggunakan ERD (*Entity Relation Diagram*). Sebelum membuat ERD, sebaiknya dilakukan identifikasi entitas-entitas yang terlibat serta atribut dari masing-masing entitas untuk mempermudah membuat ERD. Tabel 5.2 berisi daftar entitas dan atribut ERD.

Tabel 5.2 Daftar Entitas dan Atribut ERD

Entitas	Keterangan	Atribut
Inventaris	Daftar inventaris milik JTI berupa barang yang dapat dipindahkan.	kode inventaris , nama_inventaris, kode_letak, merk, kode_kategori, thn_perolehan, harga, jumlah_inventaris, inventaris_layak, inventaris_rusak, inventaris_service
Peminjaman	Transaksi peminjaman inventaris JTI.	kode peminjaman , kode_inventaris, ID_peminjam, jumlah_pinjam, tgl_pinjam, tgl_kembali
Perawatan	Perawatan inventaris JTI.	kode perawatan , kode_inventaris, userID, jenis_kerusakan, biaya_perawatan, tgl_perawatan, tgl_selesai, jumlah_inventaris
User	Orang yang menjalankan sistem.	userID , username, no_telp, userpassword, status
Peminjam	Orang yang melakukan peminjaman.	ID peminjam , nama, no_telp
Kategori Inventaris	Jenis – jenis inventaris.	kode kategori , kategori_inventaris

Lanjutan Tabel 5.2 Daftar Entitas dan Atribut ERD

Entitas	Keterangan	Atribut
Letak Inventaris	Informasi lokasi inventaris yang terkait.	<u>kode letak</u> , letak_inventaris
Exception	Pengadaan inventaris yang dilakukan oleh <i>user</i> .	<u>Kode</u> , kode_letak, userID, nama_inventaris, harga, tgl_pengajuan, keterangan

5.1.1.3 Relasi dan Normalisasi

Langkah berikutnya setelah daftar entitas dan atribut yang terlibat dalam sistem ditentukan adalah membuat relasi dan normalisasi tabel. Relasi merupakan hubungan antar entitas dalam sistem yang memuat ketergantungan *primary key* dan *foreign key*. Sedangkan normalisasi adalah suatu bentuk yang menunjukkan bahwa tabel telah memenuhi kriteria untuk dilakukan normalisasi.

5.1.1.3.1 Kardinalitas Relasi

Setelah daftar entitas dan atribut dibuat berikutnya adalah menentukan jenis relasi pada masing-masing entitas yang terlihat pada Tabel 5.3.

Tabel 5.3 Tabel Identifikasi Relasi

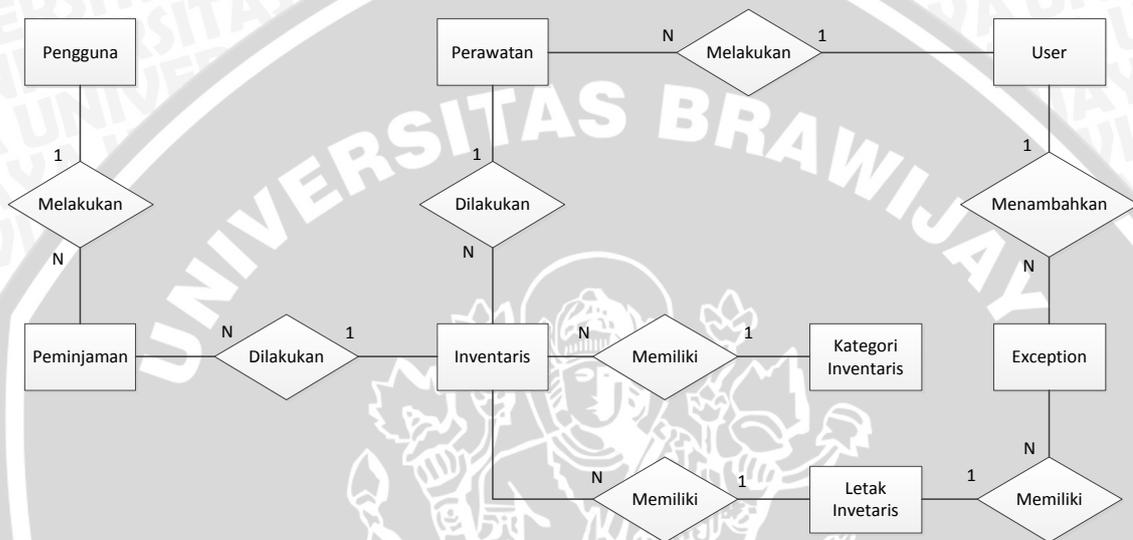
Entitas	Relasi	Entitas	Jenis Relasi
User	Melakukan	Perawatan	1:N
User	Menambahkan	Exception	1:N
Inventaris	Dilakukan	Perawatan	1:N
Inventaris	Memiliki	Letak Inventaris	N:1
Exception	Memiliki	Letak Inventaris	N:1
Inventaris	Memiliki	Kategori Inventaris	N:1
Inventaris	Dilakukan	Peminjaman	1:N
Pengguna	Melakukan	Peminjaman	1:N

Berdasarkan identifikasi relasi yang telah dilakukan, maka yang perlu dilakukan adalah mengidentifikasi hubungan antar masing-masing entitas yang terlibat sebagai berikut:

- Setiap *user* dapat melakukan perawatan pada banyak inventaris.
- Setiap *user* dapat menambahkan banyak data exception.
- Setiap inventaris dapat dilakukan perawatan berkali – kali.
- Beberapa jenis inventaris bisa terdapat pada satu lokasi yang sama.

- e. Beberapa data exception yang ditambahkan oleh *user* bisa terdapat pada satu lokasi yang sama.
- f. Beberapa inventaris bisa memiliki kategori yang sama.
- g. Setiap inventaris dapat dilakukan peminjaman berkali – kali.
- h. Setiap pengguna dapat melakukan peminjaman berkali – kali.

Berdasarkan identifikasi yang telah dilakukan, maka ERD selanjutnya dapat dibuat dengan jenis relasinya pada Gambar 5.3 berikut ini:



Gambar 5.3 Relasi ERD

Pada Gambar 5.3 semua entitas telah memenuhi bentuk normal tabel dan relasi antar entitas adalah *one to many* atau *many to one* (1-N atau N-1).

5.1.1.3.2 Normalisasi Tabel

Normalisasi adalah proses pengelompokan data ke dalam bentuk tabel untuk menyatakan entitas dan hubungan mereka sehingga terwujud satu bentuk database yang mudah untuk dimodifikasi. Selain itu, normalisasi bertujuan untuk menghilangkan kerangkapan data karena relasi *many to many* dan mengurangi kompleksitas data dan mempermudah modifikasi data. Berikut ini akan sedikit dijelaskan mengenai normalisasi tabel berdasarkan tabel peminjaman untuk menunjukkan bahwa tabel tersebut sudah memenuhi bentuk normal.

5.1.1.3.2.1 Bentuk Tidak Normal (*Unnormalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikukti format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data

dikumpulkan apa adanya sesuai dengan saat menginput. Tabel 5.4 contoh *unnormalized form*:

Tabel 5.4 Tabel Data Peminjaman Inventaris yang Tidak Normal

ID_ Pinjam	Kode_ inventaris	Nama_ Inventaris	Jumlah_ inventaris	Tgl peminjaman	Tgl Pengembalian
1234	3050201002001	Meja	3	10/05/2014	11/05/2014
	3060102135001	LCD	1	20/05/2014	20/05/2014
1111	3100102001001	Komputer	2	23/05/2014	24/05/2014

Dari Tabel 5.4 terdapat *multiple value* pada atribut ID_Pinjam 1234 yaitu pada kode_inventaris 3050201002001 dan 3060102135001. Untuk mentransformasikan tabel yang belum ternormalisasi di atas menjadi Tabel yang memenuhi kriteria 1NF adalah kita harus merubah seluruh atribut yang *multivalued* menjadi atribut *single value*, dengan cara menghilangkan *repeating group* pada tabel di atas.

5.1.1.3.2.2 Bentuk Normal Ke Satu (*First Normal Form / 1 NF*)

Pada tahap ini dilakukan penghilangan beberapa group elemen yang berulang agar menjadi satu harga tunggal yang berinteraksi di antara setiap baris pada suatu tabel, dan setiap atribut harus mempunyai nilai data yang atomic (bersifat atomic value). Atom adalah zat terkecil yang masih memiliki sifat induknya, bila terpecah lagi maka ia tidak memiliki sifat induknya.

Tabel 5.5 Tabel Data Peminjaman Inventaris 1NF

ID_ Pinjam	Kode_ inventaris	Nama_ Inventaris	Jumlah_ inventaris	Tgl peminjaman	Tgl Pengembalian
1234	3050201002001	Meja	3	10/05/2014	11/05/2014
1234	3060102135001	LCD	1	20/05/2014	20/05/2014

5.1.1.3.2.3 Bentuk Normal Ke Dua (*Second Normal Form / 2 NF*)

Syarat normal kedua (2-NF) sebagai berikut.

1. Bentuk data telah memenuhi kriteria bentuk normal kesatu.
2. Atribut bukan kunci (*non-key*) haruslah memiliki ketergantungan fungsional sepenuhnya (*fully functional dependency*) pada kunci utama / *primary key*. Dengan demikian untuk membentuk normal kedua haruslah sudah ditentukan *primary key*-nya. *Primary key* tersebut haruslah lebih sederhana, lebih unik, dapat mewakili

atribute lain yang menjadi anggotanya, dan lebih sering digunakan pada tabel / relasi tersebut. Bentuk *second normal form* akan ditunjukkan pada Tabel 5.6.

Tabel 5.6 Tabel Data Peminjaman Inventaris 2NF

Kode_ Pinjam	ID_ Pinjam	Kode_ inventaris	Nama_ Inventaris	Jumlah_ inventaris	Tgl peminjaman	Tgl Pengembalian
P0001	1234	3050201002001	Meja	3	10/05/2014	11/05/2014
P0002	1234	3060102135001	LCD	1	20/05/2014	20/05/2014

Pada Tabel 5.5 belum terdapat adanya atribut yang unik dan berbeda dengan tabel yang lain untuk mewakili nilai atribut Tabel tersebut. Oleh karena itu, pada Tabel 5.6 ditambahkan atribut kode_pinjam sebagai *primary key*, dalam hal ini dapat dikatakan menambah suatu tabel antara untuk memebentuk relasi antara Tabel Inventaris dan Tabel Peminjam. Pada Tabel 5.6 terlihat bahwa kode_inventaris memiliki ketergantungan pada kode_pinjam, tetapi id_pinjam tidak memiliki ketergantungan pada kode_pinjam.

5.1.1.3.2.4 Bentuk Normal Ke Tiga (*Third Normal Form / 3 NF*)

Walaupun relasi 2-NF memiliki redudansi yang lebih sedikit dari pada relasi 1-NF, namun relasi tersebut masih mungkin mengalami kendala bila terjadi (*update*) terhadap relasi tersebut. Misalkan kita akan melakukan update terhadap id_pinjam seperti kita harus melakukan update terhadap dua baris pada data peminjaman inventaris (lihat Tabel 5.6). Jika kita hanya mengupdate satu baris saja, sementara baris yang lainnya tidak, maka data di dalam database tersebut akan inkonsisten / tidak teratur. Anomaly update ini disebabkan oleh suatu ketergantungan transitif (*transitive dependency*). Kita harus menghilangkan ketergantungan tersebut dengan melakukan normalisasi ketiga (3-NF). Syarat dari bentuk 3-NF adalah sebagai berikut:

1. Bentuk data telah memenuhi kriteria bentuk normal kedua.
2. Atribut bukan kunci (*non-key*) harus tidak memiliki ketergantungan transitif, dengan kata lain suatu atribut bukan kunci (*non_key*) tidak boleh memiliki ketergantungan fungsional (*functional dependency*) terhadap atribut bukan kunci lainnya, seluruh atribut bukan kunci pada suatu relasi hanya memiliki ketergantungan fungsional terhadap *priamry key* di relasi itu saja.

Tabel 5.7 Tabel Data Pengguna 3NF

ID_ Pinjam	Nama	No_Telp
1234	Harry	081216
1111	Kurniawan	085731

Tabel 5.7 merupakan bentuk normal ke tiga dari Tabel Peminjaman, semua atribut tidak terdapat *multiple value* dan atribut *non-key* memiliki ketergantungan pada *primary key*.

5.1.1.4 Physical System

Bentuk *physical system* dapat mengaktualisasikan dari desain logis. Pada fase ini, aktifitas *coding* pada program dilakukan, dimana pembuatan database dan user interface dibuat dengan menggunakan *Microsoft Access 2013*.

Setelah menentukan daftar entitas dan membuat ERD maka selanjutnya adalah membuat rancangan tabel *database*. *Data type* yang dipilih pada desain tabel *database* berdasarkan pada operasi matematik yang akan diterapkan, ketelitian angka dan untuk menghemat memori.

1. Tabel Inventaris

Tabel 5.8 Desain Database Tabel Inventaris

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Keterangan</i>
Kode_inventaris	Short Text	20	Kode Inventaris	Primary Key
Nama_inventaris	Short Text	50	Nama Inventaris	
Kode_letak	Short Text	40	Kode Letak Inventaris	Foreign Key
Merk	Short Text	30	Merk Inventaris	
Kode_kategori	Short Text	20	Kode Kategori Inventaris	Foreign Key
Thn_perolehan	Number	Long Integer	Tahun Perolehan	
Harga	Currency		Harga Inventaris	
Jumlah_inventaris	Number	Long Integer	Jumlah Inventaris	
Inventaris_layak	Number	Long Integer	Jumlah Inventaris Layak	
Inventaris_rusak	Number	Long Integer	Jumlah Inventaris Rusak	
Inventaris_service	Number	Long Integer	Jumlah Inventaris Service	

2. Tabel Exception

Tabel 5.9 Desain Database Exception

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Keterangan</i>
Kode	Short Text	10	Kode Exception	Primary Key
Kode_letak	Short Text	5	Kode Letak Inventaris	Foreign Key
UserID	Short Text	10	ID Penanggungjawab	Foreign Key
Nama_inventaris	Short Text	30	Nama Inventaris	
Harga	Currency		Harga Inventaris	
Tgl_pengajuan	Date/Time		Tanggal Pengajuan Inventaris	
Keterangan	Yes/No		Keterangan pembelian inventaris	

3. Tabel Kategori Inventaris

Tabel 5.10 Desain Database Kategori Inventaris

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Keterangan</i>
Kategori_inventaris	Short Text	20	Kategori Inventaris	
Kode_kategori	Short Text	5	Kode Kategori Inventaris	Primary Key

4. Tabel Letak Inventaris

Tabel 5.11 Desain Database Letak Inventaris

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Keterangan</i>
letak_inventaris	Short Text	40	Letak Inventaris	
Kode_letak	Short Text	5	Kode letak Inventaris	Primary Key

5. Tabel Peminjaman

Tabel 5.12 Desain Database Peminjaman Inventaris

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Keterangan</i>
Kode_peminjaman	Short Text	20	Kode Peminjaman Inventaris	Primary Key
Kode_inventaris	Short Text	20	Kode Inventaris	Foreign Key
Jumlah_pinjam	Number	Long Integer	Jumlah peminjaman Inventaris	
Tgl_pinjam	Date/Time		Tanggal Peminjaman Inventaris	
Tgl_kembali	Date/Time		Tanggal Pengembalian Inventaris	
Id_peminjam	Short Text	255	Nomor identitas peminjam	Foreign Key

6. Tabel Pengguna

Tabel 5.13 Desain Database Pengguna

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Keterangan</i>
id_peminjam	Short Text	20	Nomor identitas peminjam	Primary Key
Nama	Short Text	20	Nama peminjam	
No_telp	Short Text	20	Nomor telp peminjam	

7. Tabel Perawatan

Tabel 5.14 Desain Database Perawatan Inventaris

<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Keterangan</i>
Kode_perawatan	Short Text	10	Kode Perawatan Inventaris	Primary Key
Kode_inventaris	Short Text	255	Kode Inventaris	Foreign Key
userID	Short Text	255	Identitas User	Foreign Key
Jenis_kerusakan	Short Text	255	Jenis Kerusakan Inventaris	
Biaya_perawatan	Currency		Biaya Perawatan Inventaris	
Tgl_perawatan	Date/Time		Tanggal Perawatan Inventaris	
Tgl_selesai_perawatan	Date/Time		Tanggal Selesai Perawatan Inventaris	
Jumlah_inventaris_service	Number	Long Integer	Jumlah Inventaris yang diservis	

8. Tabel *User*Tabel 5.15 Desain *Database User*

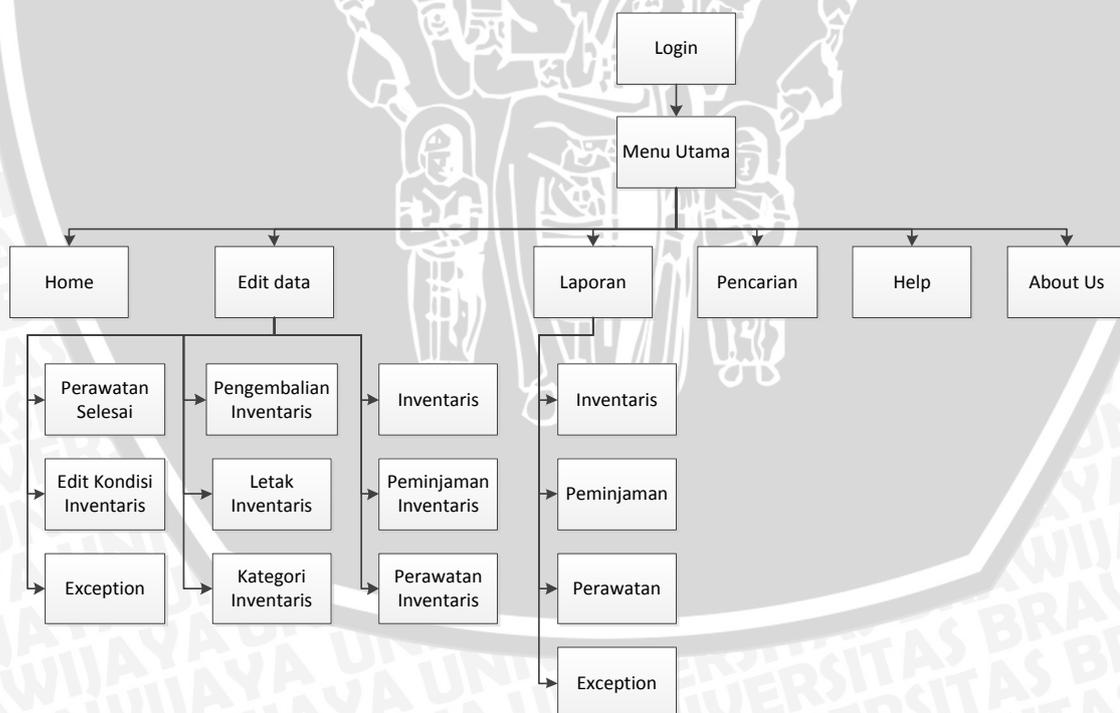
<i>Field</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Note</i>	<i>Keterangan</i>
userID	Short Text	25	Nomor identitas <i>user</i>	Primary Key
Username	Short Text	20	Nama <i>user</i>	
No_telp	Short Text	20	Nomor telp <i>user</i>	
userpassword	Short Text	25	Password <i>user</i>	
Status	Short Text	50	Status jabatan <i>user</i>	

5.2 Desain Antar Muka (*User Interface Design*)

Tampilan visual merupakan hal yang penting dalam interaksi pengguna dengan sistem. Sistem dalam tampilan visual harus memenuhi beberapa ketentuan yang telah dirancang, agar mudah mengerti dan digunakan oleh admin dan pengguna. Perancangan user interface akan ditunjukkan pada Gambar 5.4.

5.2.1 Bagan Hierarki Menu *User Interface*

Berdasarkan bab IV Tabel 4.1, maka dapat ditentukan rancangan hierarki menu *user interface* sesuai dengan kebutuhan *user* yang terdiri dari menu Home, Edit Data, Laporan, Pencarian, Help dan About Us yang akan dijelaskan pada Gambar 5.4.

Gambar 5.4 Bagan Hierarki Menu *User Interface*

Dari Gambar 5.4 dapat dilihat bahwa setelah *user* melakukan login terdapat menu utama dari SIMIN JTI. Menu utama SIMIN terdiri dari enam menu yaitu Home, Edit Data, Laporan, Pencarian, Help, dan About Us. Pada menu Edit Data dan Laporan

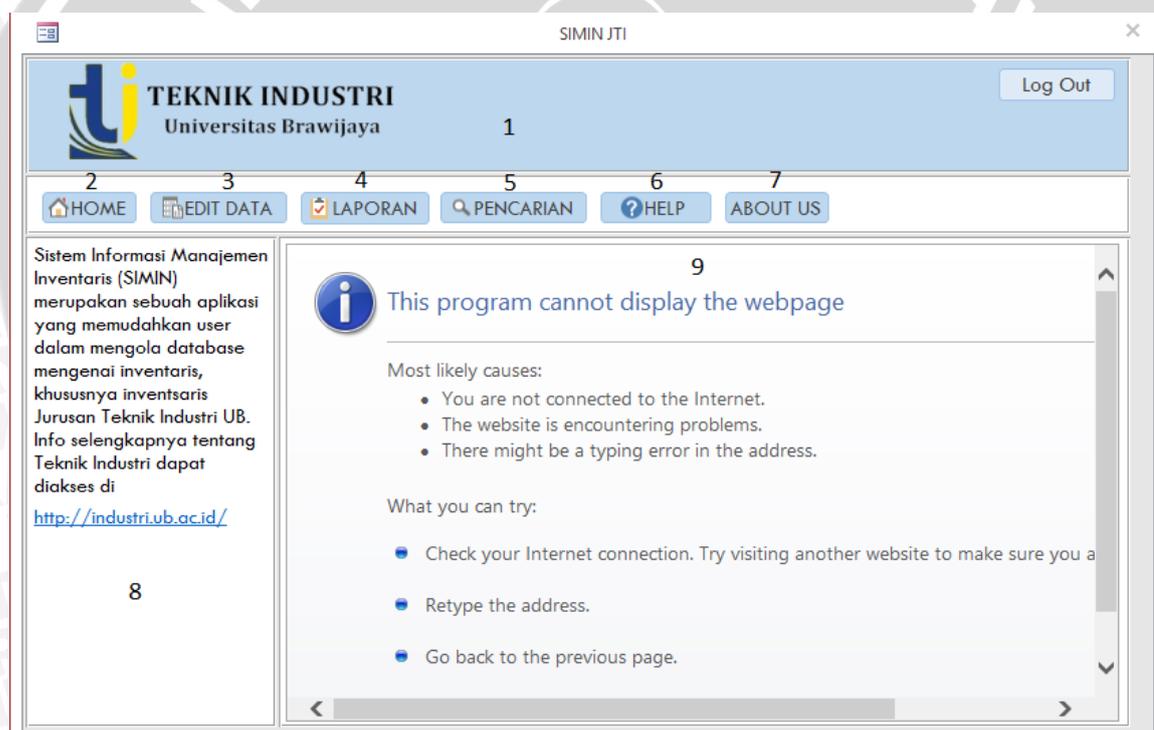
terdiri dari beberapa sub-menu. Sub-menu Edit Data berisi Inventaris, Peminjaman Inventaris, Perawatan Inventaris, Pengembalian Inventaris, Letak Inventaris, Kategori Inventaris, Perawatan Selesai, Edit Kondisi Inventaris dan Exception. Menu Laporan terdiri dari 4 submenu yaitu, Inventaris, Peminjaman, Perawatan dan Exception. Adapun penjelasan dari masing – masing fitur adalah:

1. Home : merupakan halaman awal SIMIN yang berfungsi untuk memperkenalkan *user* dengan aplikasi SIMIN.
2. Edit Data : fitur yang berfungsi untuk menambah, menghapus dan melihat data SIMIN. Dalam fitur Edit Data terdapat beberapa *submenu*, diantaranya:
 - a. Inventaris : untuk menambahkan Data Inventaris.
 - b. Peminjaman Inventaris : untuk menambahkan data Peminjaman Inventaris.
 - c. Pengembalian Inventaris : untuk mencatat pengembalian dan melihat daftar peminjaman inventaris.
 - d. Perawatan Inventaris : untuk menambahkan data perawatan inventaris.
 - e. Perawatan Selesai : untuk mencatat perawatan selesai dan melihat daftar perawatan inventaris.
 - f. Letak Inventaris : untuk menambahkan daftar letak inventaris.
 - g. Kategori Inventaris : untuk menambahkan jenis kategori baru dalam inventaris.
 - h. Edit Kondisi Inventaris : untuk mencatat daftar kondisi dalam inventaris.
 - i. Exception : untuk menambahkan daftar perencanaan inventaris baru yang akan dibeli.
3. Laporan : merupakan fitur SIMIN yang memberikan laporan tentang inventaris, sesuai dengan kebutuhan *user* sesuai dengan *output* yang diinginkan. *Submenu* yang terdapat dalam Laporan adalah:
 - a. Laporan Inventaris : memberikan informasi laporan *summary* (keseluruhan) tentang daftar inventaris.
 - b. Laporan Peminjaman : memberikan informasi laporan bulanan, tahunan, *summary* atau *costum report*(laporan bebas) mengenai daftar peminjaman.
 - c. Laporan Perawatan : memberikan informasi laporan bulanan, tahunan, *summary* atau *costum report*(laporan bebas) mengenai daftar perawatan.
 - d. Laporan Exception : memberikan informasi laporan bulanan, tahunan, *summary* atau *costum report*(laporan bebas) mengenai daftar exception.

4. Pencarian : *Menu* pencarian merupakan fitur SIMIN yang berfungsi memberikan informasi tentang inventaris kepada *user*.
5. Help : merupakan *user manual* yang menjelaskan langkah – langkah detail dalam penggunaan aplikasi SIMIN.
6. About Us : memberikan informasi tambahan mengenai SIMIN untuk membantu pengembangan aplikasi.

5.2.2 Desain Interface

Desain *interface* adalah perancangan tampilan SIMIN yang nantinya digunakan oleh *user*. Desain *interface* ini diatur sesuai dengan kebutuhan *user* pada hirarki menu *user interface*. Selain itu penataan letak dan warna juga harus dipertimbangkan agar *user* tidak kesulitan ketika menggunakan aplikasi SIMIN. Desain *interface* SIMIN JTI bisa dilihat pada Gambar 5.5 dengan penjelasan pada Tabel 5.16.



Gambar 5.5 Desain Interface

Keterangan:

- | | |
|--------------|-------------------|
| 1. Header | 6. Help |
| 2. Home | 7. About Us |
| 3. Edit Data | 8. SubMenu |
| 4. Laporan | 9. Detail SubMenu |
| 5. Pencarian | |

Tabel 5.16 Faktor Penentuan Ukuran *Interface*

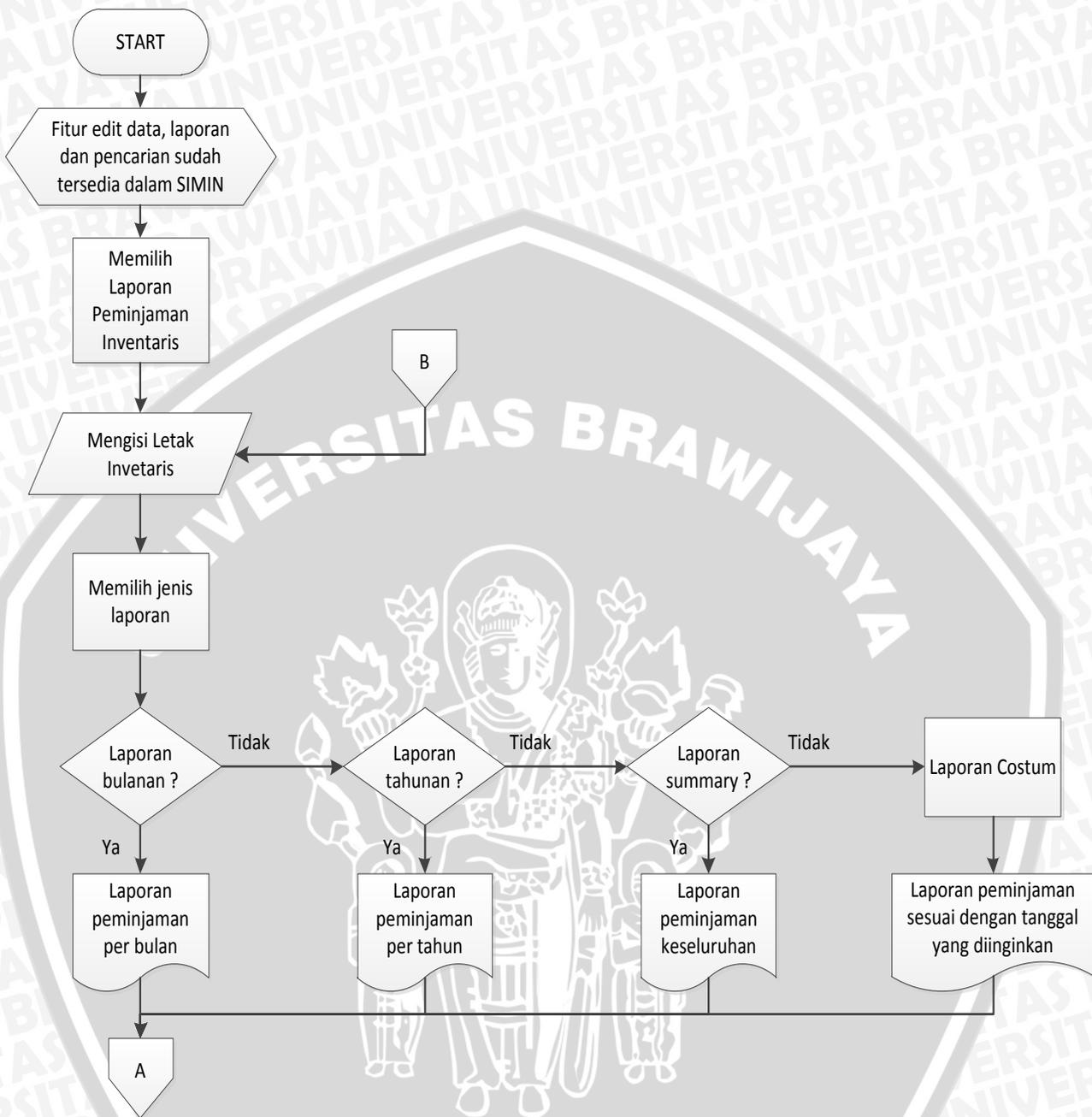
	Ukuran (panjang x lebar)	Warna	Font	Font size	Pertimbangan Faktor Ergonomis
<i>Header</i>	21.3 x 2.2 cm	Biru Muda			<p>a. <i>Header</i> merupakan kolom yang berfungsi untuk memperkenalkan aplikasi SIMIN.</p> <p>b. Warna yang dipilih adalah biru muda karena cerah tapi tidak menyilaukan sehingga tidak membuat mata cepat lelah.</p>
<i>Menu Container</i> (Home, Edit Data, Laporan, Pencarian, Help dan About Us)	21.2 x 1.1 cm	<i>Menu Container Color</i> : Putih <i>Menu Color</i> : Biru Muda	Tw Cen MT (Detail)	11	<p>a. Menu merupakan penghubung untuk mengakses semua fitur SIMIN.</p> <p>b. <i>Menu Color</i> yang dipilih adalah biru muda karena cerah tapi tidak menyilaukan sehingga tidak membuat mata cepat lelah. Sedangkan <i>Menu Container Color</i> yang dipilih adalah putih karena tidak contrast dengan biru muda sehingga mata lebih mudah menyesuaikan dengan layar komputer.</p> <p>c. Font yang dipilih sebesar 11 dengan format <i>uppercase</i> untuk membedakan Menu dengan <i>submenu</i>. Jenis font</p> <p>d. yang digunakan adalah Tw Cent MT karena tidak menghabiskan banyak tempat tapi tetap bisa dibaca.</p>

Lanjutan Tabel 5.16 Faktor Penentuan Ukuran *Interface*

	Ukuran (panjang x lebar)	Warna	Font	Font size	Pertimbangan Faktor Ergonomis
					e. Ukuran <i>menu container</i> disesuaikan dengan besarnya form maksimal, yaitu 21.2 x 1.1 cm. Sedangkan masing – masing menu ukurannya disesuaikan dengan besar font masing – masing.
<i>Submenu Container</i>	4.7 x 9.5 cm	Putih			Merupakan <i>container submenu</i> . Warna yang background yang dipilih adalah putih karena disesuaikan dengan jenis warna dari <i>submenu</i> .
<i>Submenu detail</i>	16.4 x 9.4 cm	Biru Muda	Tw Cen MT (Detail)	11	a. <i>Submenu detail</i> merupakan penjelasan singkat untuk beberapa <i>submenu</i> , sehingga ukurannya disesuaikan dengan detail dari <i>submenu</i> . b. Warna yang dipilih adalah putih karena merupakan <i>container</i> dari beberapa <i>submenu</i> .

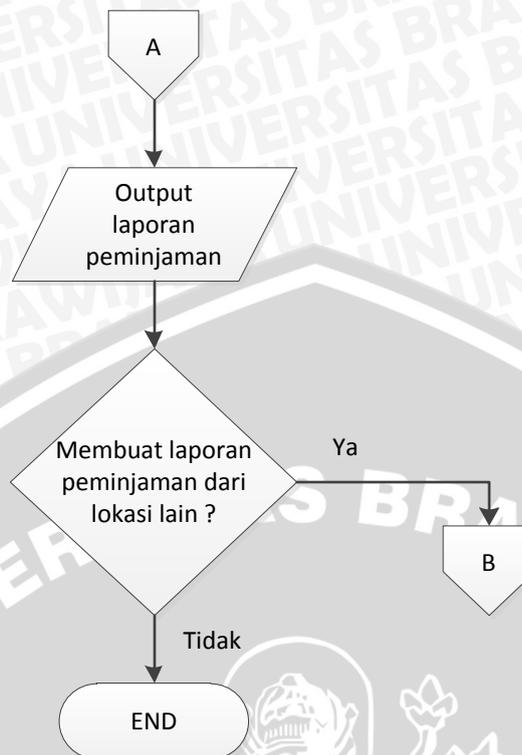
5.3 Desain Algoritma

Algoritma adalah otak dari software, yang melukiskan langkah demi langkah yang harus dilakukan untuk mencapai tujuan. Algoritma pada umumnya berupa *flowchart* atau *pseudocode*. Berikut ini merupakan algoritma dari menu laporan peminjaman pada dalam bentuk *flowchart*.



Gambar 5.6 Flowchart Laporan Peminjaman Inventaris



Gambar 5.7 Lanjutan *Flowchart* Laporan Peminjaman Inventaris

5.4 Implementasi

Pada fase ini semua perancangan yang telah dilakukan akan diaplikasikan ke dalam *software*. Implementasi bertujuan agar *user* dapat menggunakan program yang telah dianalisa dan dirancang sebelumnya. Pada fase ini terdapat dua bagian, yaitu implementasi *database* dan implementasi *user interface*.

5.4.1 Implementasi *Database*

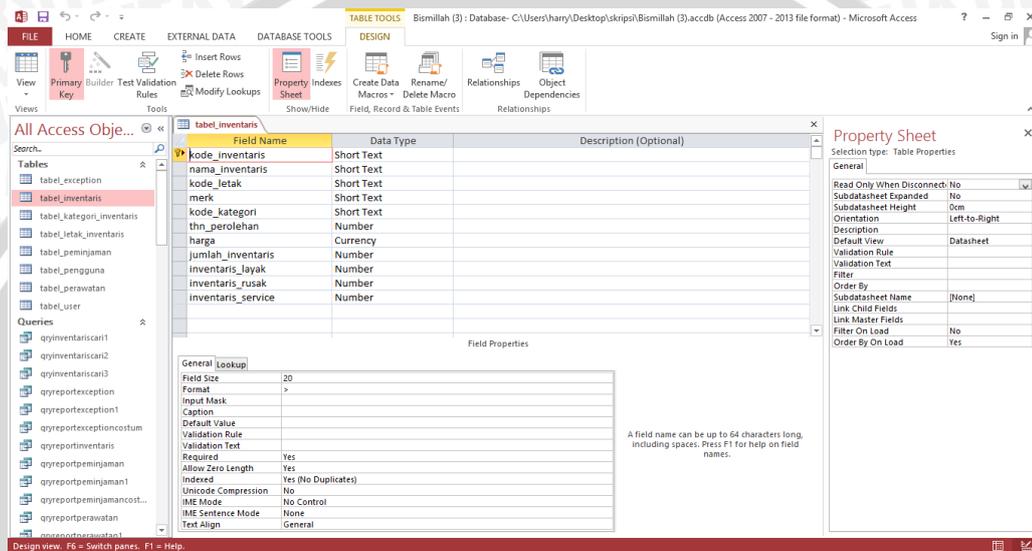
Pembuatan *database* ini diperlukan untuk memasukkan, menghapus, mengubah, memanipulasi, dan memperoleh data atau informasi seluruh *content* yang ada di dalam SIMIN nantinya. Pembuatan *database* pada aplikasi SIMIN JTI dengan menggunakan *Microsoft Access* dapat dilakukan dengan cara sebagai berikut:

5.4.1.1 Pembuatan *Database*

Setelah entitas dan relasi ditentukan, langkah selanjutnya adalah membuat Tabel tersebut. Berikut merupakan tabel inventaris, tabel peminjaman dan tabel perawatan. Sedangkan untuk tabel lainnya akan ditampilkan pada lampiran.

1. Tabel Inventaris

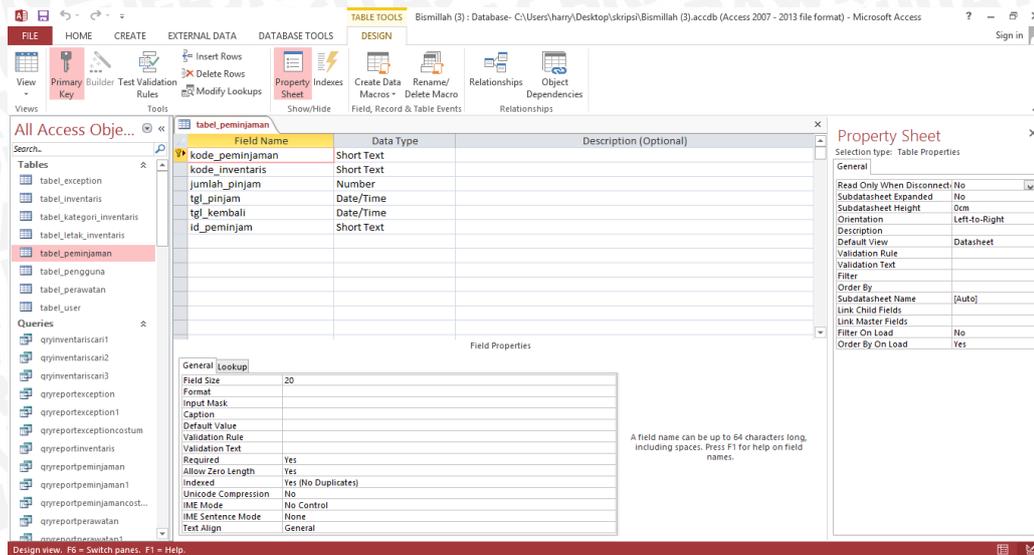
Tabel inventaris merupakan database untuk menyimpan data mengenai inventaris. Dalam tabel inventaris terdapat data mengenai kode_inventaris, nama_inventaris, kode_letak, merk, kode_kategori, thn_perolehan, harga, dan jumlah_inventaris. Detail mengenai rincian pembuatan tabel inventaris terdapat pada Tabel 5.8. Sedangkan implementasi tabel database inventaris ditunjukkan pada gambar 5.8.



Gambar 5.8 *Printscreen* Tabel Database Inventaris pada Microsoft Access 2013

2. Tabel Peminjaman

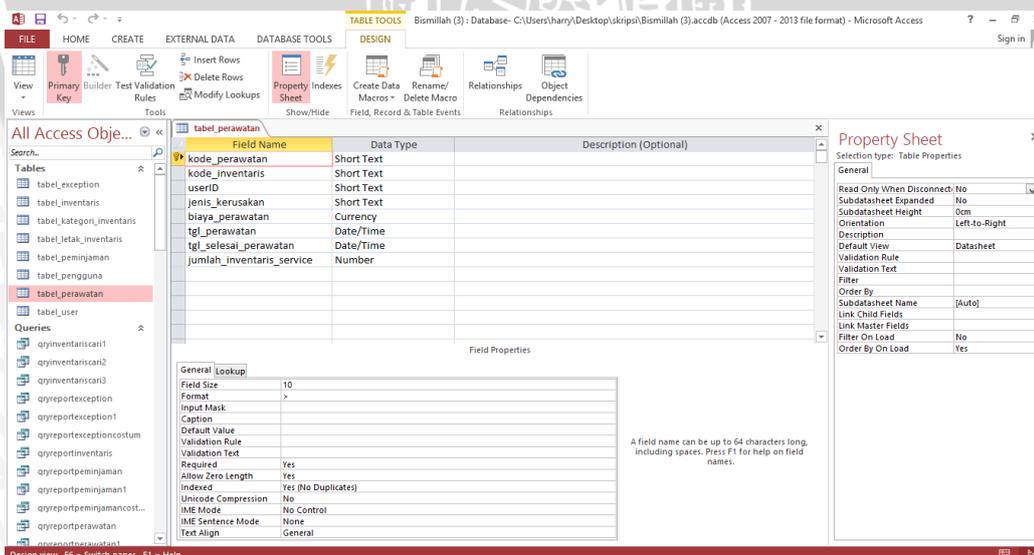
Tabel peminjaman merupakan database untuk menyimpan data mengenai peminjaman inventaris. Dalam tabel peminjaman terdapat data mengenai kode_peminjaman, kode_inventaris, jumlah_pinjam, tgl_pinjam, tgl_kembali, dan id_peminjaman. Detail mengenai rincian pembuatan tabel peminjaman terdapat pada tabel 5.12. Sedangkan implementasi tabel database peminjaman inventaris ditunjukkan pada gambar 5.9.



Gambar 5.9 *Printscreen* Tabel Peminjaman pada *Microsoft Access 2013*

3. Tabel Perawatan

Tabel perawatan merupakan database untuk menyimpan data mengenai perawatan inventaris. Dalam tabel perawatan terdapat data mengenai kode_perawatan, kode_inventaris, userID, jenis_kerusakan, biaya_perawatan, tgl_perawatan, tgl_selesai_perawatan, dan jumlah_inventaris_service. Detail mengenai rincian pembuatan tabel peminjaman terdapat pada tabel 5.14. Sedangkan implementasi tabel database peminjaman inventaris ditunjukkan pada gambar 5.10.



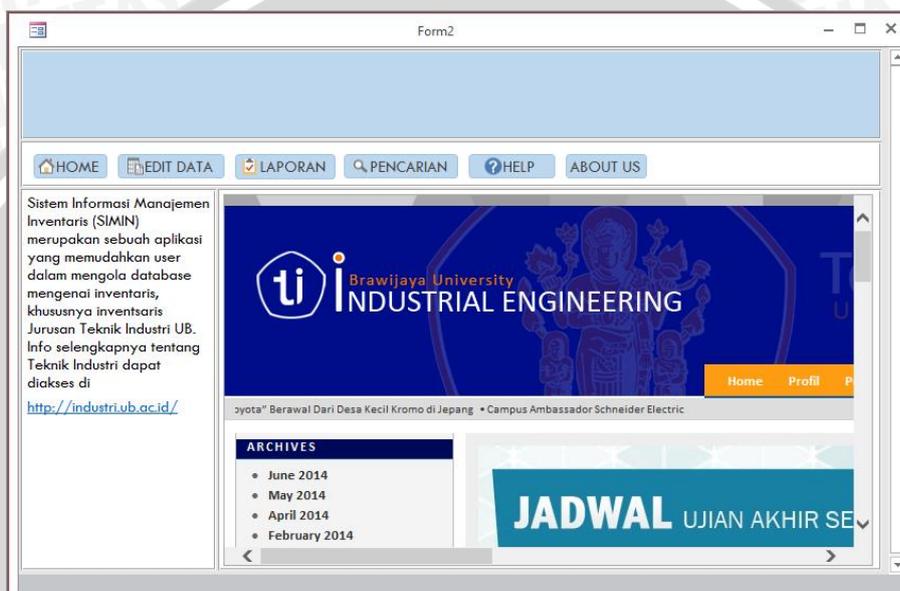
Gambar 5.10 *Printscreen* Tabel Perawatan pada *Microsoft Access 2013*

5.4.2 Implementasi *User Interface*

Implementasi ini didasarkan atas analisa dan desain yang telah dibuat pada fase sebelumnya. Berikut ini merupakan implementasi *User Interface Main Menu*, Edit Data dan Tambah Data Inventaris, sedangkan untuk *User Interface* yang lainnya akan ditampilkan pada lampiran.

1. Implementasi *User Interface Main Menu*

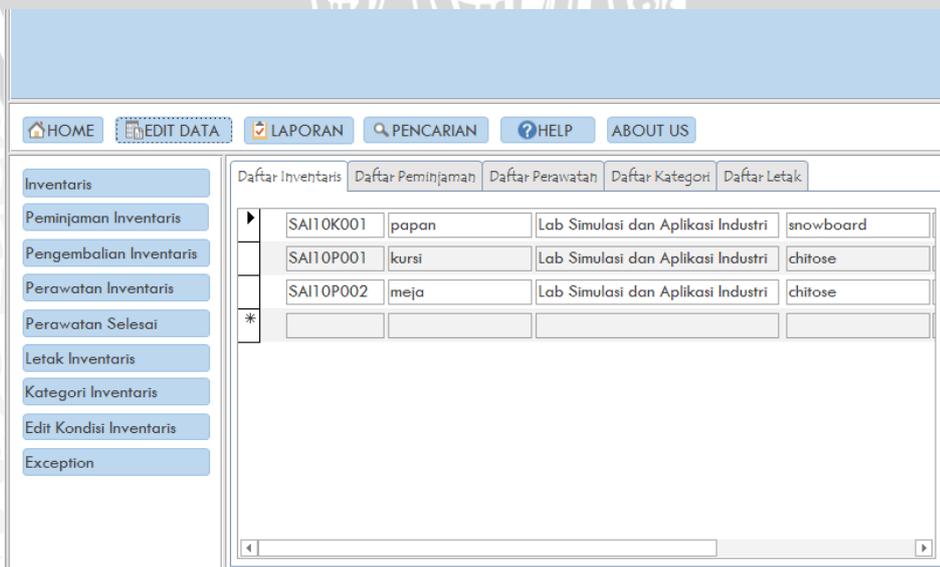
Implementasi *user interface Main Menu* untuk mengakses semua fitur SIMIN JTI.



Gambar 5.11 *Print screen Main Menu SIMIN*

2. Implementasi *User Interface Edit Data*

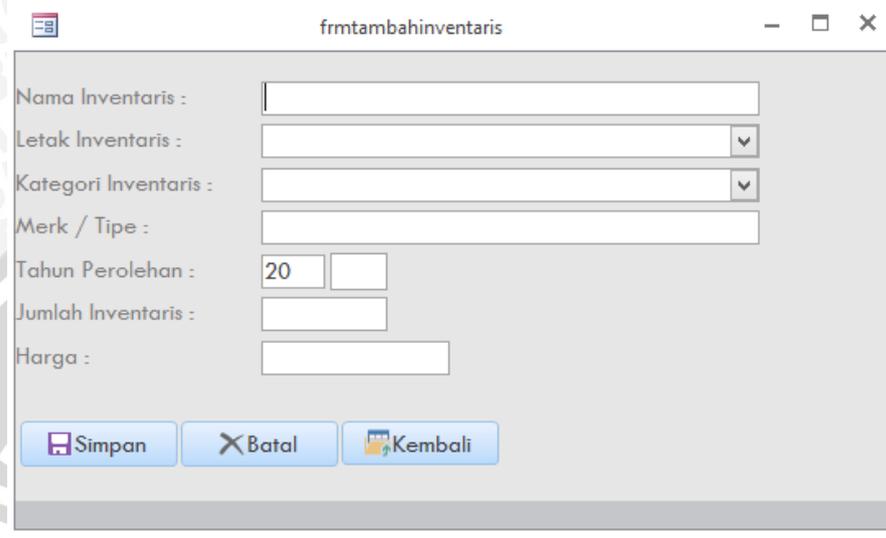
Implementasi *User Interface Edit Data* untuk melakukan penambahan data, menghapus data dan melihat data.



Gambar 5.12 *Print screen Edit Data SIMIN*

3. Implementasi *User Interface* Tambah Data Inventaris

Implementasi *User Interface* Tambah Data Inventaris untuk melakukan penambahan data Inventaris.



Gambar 5.13 *Print screen* Tambah Data Inventaris SIMIN

5.4.3 Pembuatan Kode Program (*Source Code*)

Berikut merupakan penggalan *source code* dari form tambah data inventaris.

Untuk *source code* lainnya akan ditampilkan di lampiran.

```
Option Explicit
Dim RS As Recordset

'memberi kode inventaris menggunakan group teknologi
Sub kode()
Dim kode As String
Dim kd As String
Me.RecordSource = "Tabel_inventaris"
Set RS = Me.RecordsetClone
RS.FindLast " kode_letak = '" & txtkdletak.Value & "'"
RS.FindFirst " kode_kategori = '" & txtkdktg.Value & "'"
If RS.NoMatch Then
txtkdinv.Value = txtkdletak.Value + txtthn.Value + txtkdktg.Value +
"001"
Else
kd = Val(Right(RS(0), 2)) + 1
Select Case Len(kd)
```

```

Case 1
txtkdinv.Value = txtkdletak.Value + txtthn.Value + txtkdktg.Value +
"00" + Trim(Str(kd))
Case 2
txtkdinv.Value = txtkdletak.Value + txtthn.Value + txtkdktg.Value +
"0" + Trim(Str(kd))
End Select
End If
End Sub

'menyimpan data
Private Sub cmdsimpan_Click()
Dim RS3 As Recordset
If (IsNull(txtinv) Or IsNull(cmbletak) Or IsNull(cmbktg) Or
IsNull(merk) Or IsNull(txtthn) Or IsNull(txtjmlh) Or IsNull(txtharga))
Then
    Beep
    MsgBox "Silahkan Lengkapi Data Terlebih Dahulu !",
vbInformation, "SIMIN JTI"
Else
Me.RecordSource = "Tabel_inventaris"
Set RS3 = Me.RecordsetClone
RS3.AddNew
RS3(0) = txtkdinv.Value
RS3(1) = txtinv.Value
RS3(2) = txtkdletak.Value
RS3(3) = merk.Value
RS3(4) = txtkdktg.Value
RS3(5) = txtthnperolehan.Value
RS3(6) = txtharga.Value
RS3(7) = txtjmlh.Value
RS3.Update
MsgBox "Data anda telah tersimpan!", vbInformation, "SIMIN JTI"
bersih
Me.Refresh
kode
End If
End Sub

```

5.5 Pengujian (*Testing*)

Tahapan terakhir setelah sistem sudah menjadi *prototype* adalah *testing* (pengujian). Langkah pengujian ini penting untuk melihat apakah *prototype* yang telah dibuat sudah sesuai dengan 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.

5.5.1 Uji Verifikasi

Verifikasi adalah proses pemeriksaan desain dan ketelitian antara logika operasional model (program komputer) dengan logika diagram alur. Verifikasi dari suatu model ini memeriksa penerjemahan model matematis konseptual (diagram alur dan asumsi) ke dalam bahasa pemrograman secara benar. Uji verifikasi tersebut dilakukan dengan pertanyaan-pertanyaan kunci seperti pada Tabel 5.17.

Tabel 5.17 Uji Verifikasi

No	Jenis Pengujian	Fungsi Uji	Hasil Uji
1.	Menu “Log in”	Menekan tombol Log in	Sukses, <i>user</i> berhasil masuk dan membuka halaman awal SIMIN.
2.	Menu “Home”	Menekan tombol Home	Sukses, mengembalikan ke tampilan awal program “Home”
3.	Menu “Edit Data”	Menekan tombol Edit Data	Sukses, <i>user</i> dapat membuka menu “Edit Data”
4.	Menu “Laporan”	Menekan tombol Laporan	Sukses, <i>user</i> dapat membuka menu “Laporan”
5.	Menu “Pencarian”	Menekan tombol Pencarian	Sukses, <i>user</i> dapat membuka menu “Pencarian”
6.	Menu “Help”	Menekan tombol Help	Sukses, <i>user</i> dapat membuka menu “Help”
7.	Menu “About Us”	Menekan tombol About Us	Sukses, <i>user</i> dapat membuka menu “About Us”
8.	Simpan Data	Menekan tombol Simpan pada <i>submenu</i> Edit Data	Sukses, menyimpan data
9.	Pencarian Data	Memilih jenis pencarian pada <i>submenu</i> Pencarian	Sukses, melakukan pencarian data.

5.5.2 Uji Validasi

Uji validasi bertujuan untuk melihat dan memeriksa apakah proses yang telah dirancang setelah verifikasi sesuai dengan kebutuhan *user*. Uji validasi admin dilakukan oleh manajemen JTI. Melalui Tabel 5.18 dijelaskan berbagai kebutuhan tiap pengguna terhadap sistem yang dibuat.

Tabel 5.18 Uji Validasi

Pengguna	Kebutuhan Pengguna
User	Mengelola <i>database</i> dengan baik.
	Melakukan pencarian data dengan cepat.
	Membuat Laporan Inventaris sesuai dengan kebutuhan.
	Sistem yang mudah digunakan.

5.5.3 Uji Prototype

Uji *prototype* dilakukan dengan tujuan untuk mengetahui apakah *prototype* yang dibuat telah memenuhi kebutuhan *user*. Pada pengujian ini mendiskripsikan mengenai kelebihan sistem baru dibanding sistem lama yang akan dijelaskan pada Tabel 5.19.

Tabel 5.19 Uji Prototype

Jenis Atribut	Sistem Lama	Sistem Baru
<i>Performance</i>	<ul style="list-style-type: none"> - Banyaknya rekaman data yang diperoleh pada kurun waktu tertentu tergantung pada kemampuan <i>user</i> dalam mengolah data. - Pencarian data tidak dilakukan secara otomatis dan dilakukan secara manual, sehingga membutuhkan waktu yang cukup lama. 	<ul style="list-style-type: none"> - Kemudahan dalam melakukan rekap data sesuai kebutuhan. - Pencarian data dapat dilakukan secara otomatis.
<i>Information</i>	<ul style="list-style-type: none"> - Informasi yang disampaikan oleh <i>user</i> masih memungkinkan terjadi kesalahan. - Memungkinkan adanya data yang <i>redundant</i>. - Sistem informasi yang ada saat ini menggunakan <i>microsoft excel</i> yang kurang informatif. - Data tidak disusun dengan baik. - Kesesuaian informasi yang disampaikan dengan kondisi yang ada juga masih belum tentu akurat sepenuhnya. 	<ul style="list-style-type: none"> - Informasi yang disampaikan oleh <i>user</i> tidak terjadi kesalahan. - Data yang disimpan memiliki atribut yang berbeda – beda sehingga tidak ada data yang memiliki nilai sama. - Sistem informasi yang ada saat ini menggunakan <i>microsoft access</i>. - Data yang disimpan terstruktur dengan baik. - Informasi yang disimpan dan informasi yang disampaikan tepat sama.
<i>Economy</i>	<ul style="list-style-type: none"> - Biaya yang digunakan oleh <i>user</i> dalam mengoperasikan data dan menjalankan aplikasi masih belum diketahui. 	<ul style="list-style-type: none"> - <i>User</i> tidak memerlukan biaya dalam menjalankan aplikasi.
<i>Control</i>	<ul style="list-style-type: none"> - Data yang <i>redundant</i> disimpan secara tidak konsisten, karena disimpan dalam file yang berbeda – beda. 	<ul style="list-style-type: none"> - Tidak terdapat <i>redundant</i> data karena semua data memiliki atribut yang berbeda.
<i>Efficiency</i>	<ul style="list-style-type: none"> - Penyampaian informasi mengenai inventaris masih kurang efisien karena terdapat data yang <i>redundant</i>. 	<ul style="list-style-type: none"> - Penyampaian informasi mengenai inventaris lebih terstruktur.

Jenis Atribut	Sistem Lama	Sistem Baru
<i>Service</i>	<ul style="list-style-type: none"> - Sistem yang ada dapat menghasilkan <i>output</i> yang tidak konsisten. - Pencarian inventaris yang dilakukan secara mendadak dengan informasi tertentu juga sulit dilakukan mengingat pencarian yang dilakukan secara manual. 	<ul style="list-style-type: none"> - Output sistem memiliki data yang konsisten. - Dapat melakukan pencarian inventaris dengan mudah dan cepat.

Berdasarkan pada Tabel 5.19, menunjukkan bahwa adanya sistem yang baru lebih baik dari segi *performance, information, economy, control efficiency* dan *service*. Maka tujuan dari rekayasa Sistem Informasi Manajemen Inventaris JTI dapat dikatakan telah terpenuhi.

