

## **BAB III**

### **METODE PENELITIAN**

Pada desain sistem berbasis komputer, analisis memegang peranan yang penting dalam membuat rincian sistem baru. Analisis perangkat lunak merupakan langkah pemahaman persoalan sebelum mengambil tindakan atau keputusan penyelesaian hasil utama. Analisis sistem perangkat lunak adalah dokumen rincian atau dokumen sasaran.

Langkah selanjutnya dalam penyusunan sistem berbasis komputer adalah desain sistem yang bertujuan memberikan gambaran yang jelas kepada pemrogram komputer yang akan mengimplementasikan sistem.

#### **3.1. Model yang diambil/digunakan**

Pada penelitian ini model yang diambil adalah proyek pembuatan aplikasi *test data generator*. Tahap-tahap pembuatan aplikasi secara umum adalah sebagai berikut :

1. **Akusisi.** Digunakan untuk mengambil atau melakukan akusisi dan menganalisa kebutuhan sistem. *Outputnya* adalah diagram alur sistem dan spesifikasi umum sistem. Alat bantu yang digunakan adalah Visio Technical 4.1.
2. **Desain sistem.** Digunakan untuk melakukan desain sistem berdasarkan pada *output* tahap akusisi kebutuhan. *Output* pada tahap ini adalah diagram berjenjang, *Data Flow Diagram (DFD)*, *Entity Relational Diagram (ERD)*, struktur *database*, dan desain *user interface*. Alat bantu yang digunakan adalah Power designer 6 dan Visio Technical 4.1.

3. **Pembuatan aplikasi.** Berfungsi untuk membuat program berdasarkan desain sistem. *Outputnya* adalah *source code* (terlampir). Alat bantu yang digunakan adalah bahasa pemrograman Visual Basic 6.0, sedangkan sistem *database* menggunakan Microsoft Access dan SQL Server 7.
4. **Testing.** Bertujuan untuk memastikan bahwa program yang dibuat sesuai dengan desain kebutuhan sistem. Metode testing yang digunakan adalah *Blackbox Testing*.
5. **Implementasi.** Berfungsi untuk membuat *source code* menjadi aplikasi yang siap didistribusikan (akan dibahas pada bab 4). Alat bantu yang digunakan adalah Microsoft Studio *Installer*.

#### 3.1.1. Akusisi

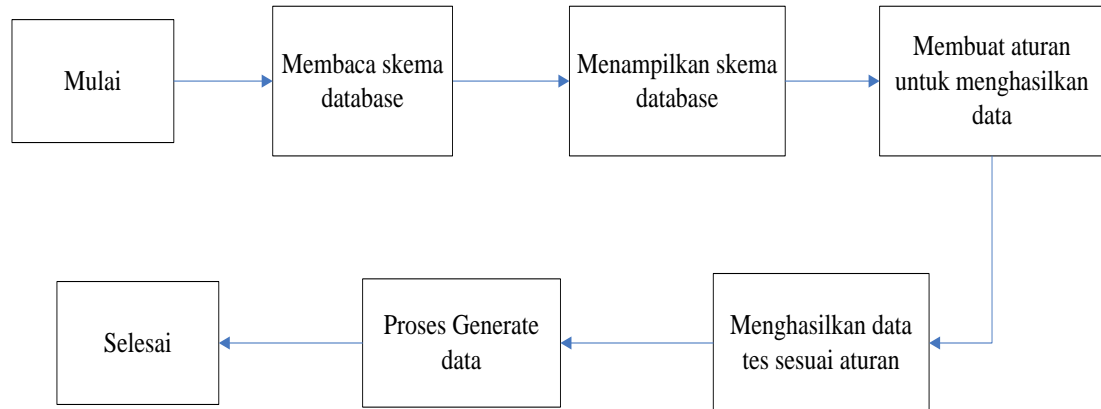
*Output* pada tahap ini adalah diagram alur sistem dan spesifikasi sistem.

Spesifikasi sistem yang *compatible* adalah sebagai berikut:

- a) Bahasa pemrograman *Microsoft Visual basic 6.0*.
- b) Obyek *database Microsoft Access 97, Microsoft Access 2000, Microsoft SQL Server 7.0, dan Microsoft SQL Server 2000*.

*Output* yang lain adalah diagram alur (*flow chart*). *Flow chart* merupakan suatu diagram yang digunakan untuk menunjukkan alur pekerjaan secara keseluruhan dari suatu sistem yang terdiri dari simbol-simbol sederhana yang dapat memudahkan dalam menemukan kesalahan-kesalahan dalam program.

Secara umum sistem yang dibangun tampak seperti gambar berikut :



Gambar 3.1. Gambaran umum sistem *Test Data Generator*.

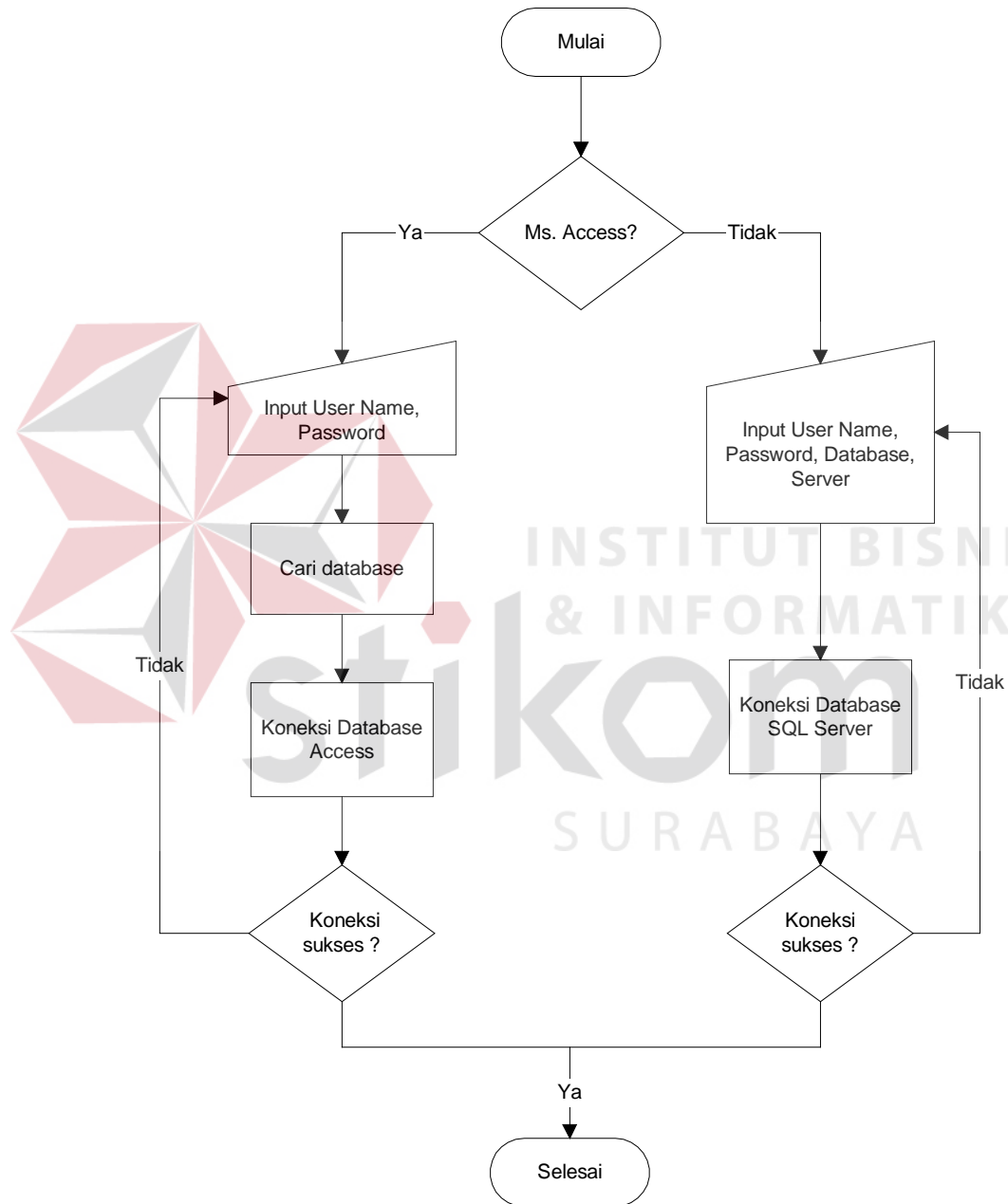
Gambaran umum dari sistem *tes data generator* (gambar 3.1) ini, dimulai dengan menghubungkan sistem dengan *database* yang dituju. Sistem ini memberikan pilihan koneksi dengan dua macam sistem *database*, antara lain (1) *Database Access*, dan (2) *Database SQL Server*.

Bila *database* berhasil dikoneksikan, maka sistem akan membaca skema *database* tersebut. *Database* akan dipetakan ke dalam bentuk skema, yaitu skema yang menghubungkan relasi-relasi antara tabel-tabel. Pada waktu membaca skema *database*, sistem membaca nama-nama tabel, *view*, kolom, tipe data pada tiap kolom serta level *field* yang berfungsi untuk melihat urutan table mana dulu yang akan di-*generate*.

*User* dapat membuat aturan pada waktu memasukkan data, misalnya data berisi karakter, numerik, tanggal/jam, atau multi *field*. Kemudian data tersebut yang ada dalam suatu tabel akan di-*generate* secara otomatis. Dan yang terakhir, sistem akan memproduksi data sesuai dengan aturan-aturan sebelumnya yang telah dibuat oleh *user*. Pada sistem ini digunakan untuk mengakses *database*.

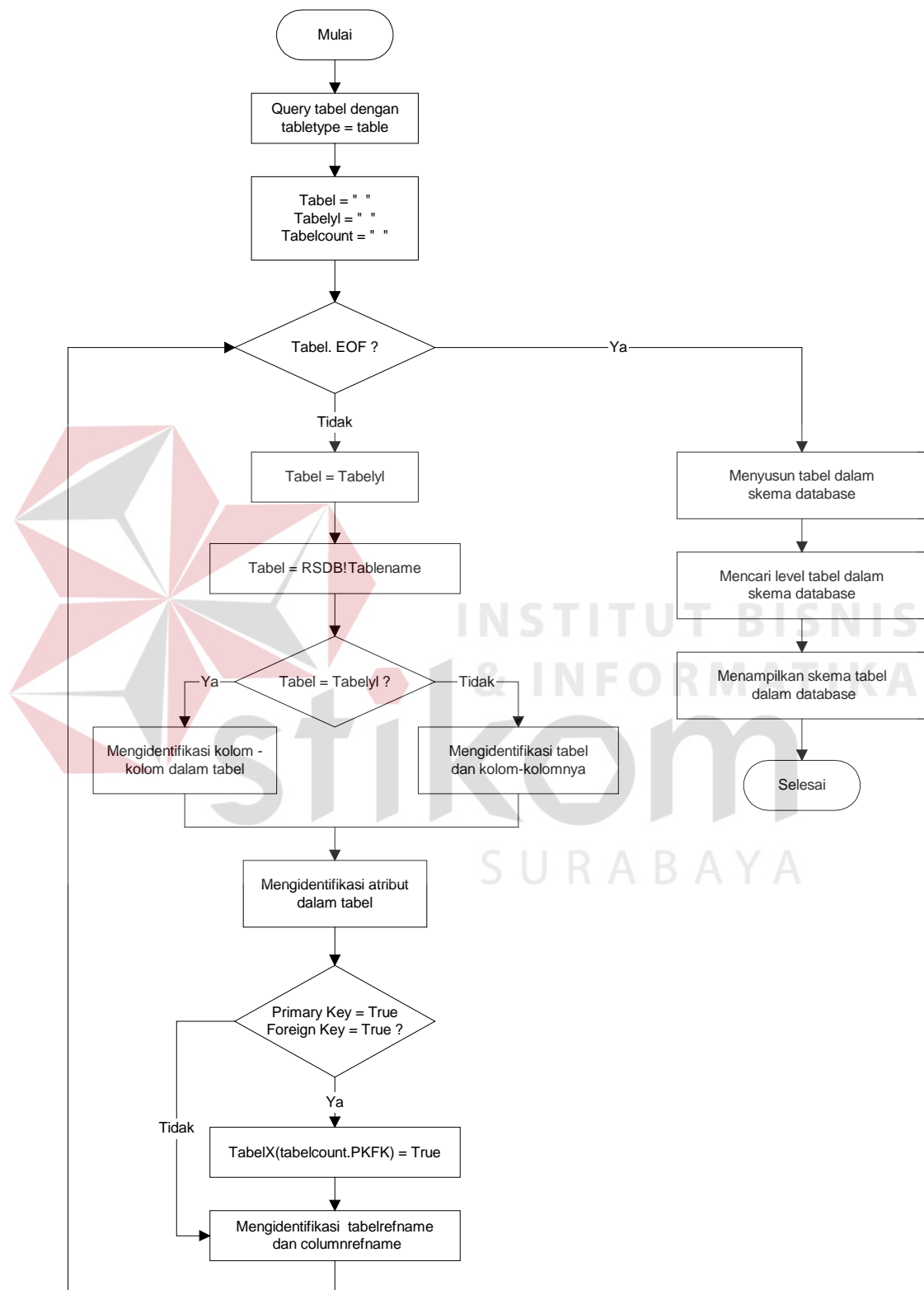
Adapun desain untuk beberapa bagian sistem dapat digambarkan dengan *flow chart* yang tampak sebagai berikut:

1. *Flow chart koneksi database*



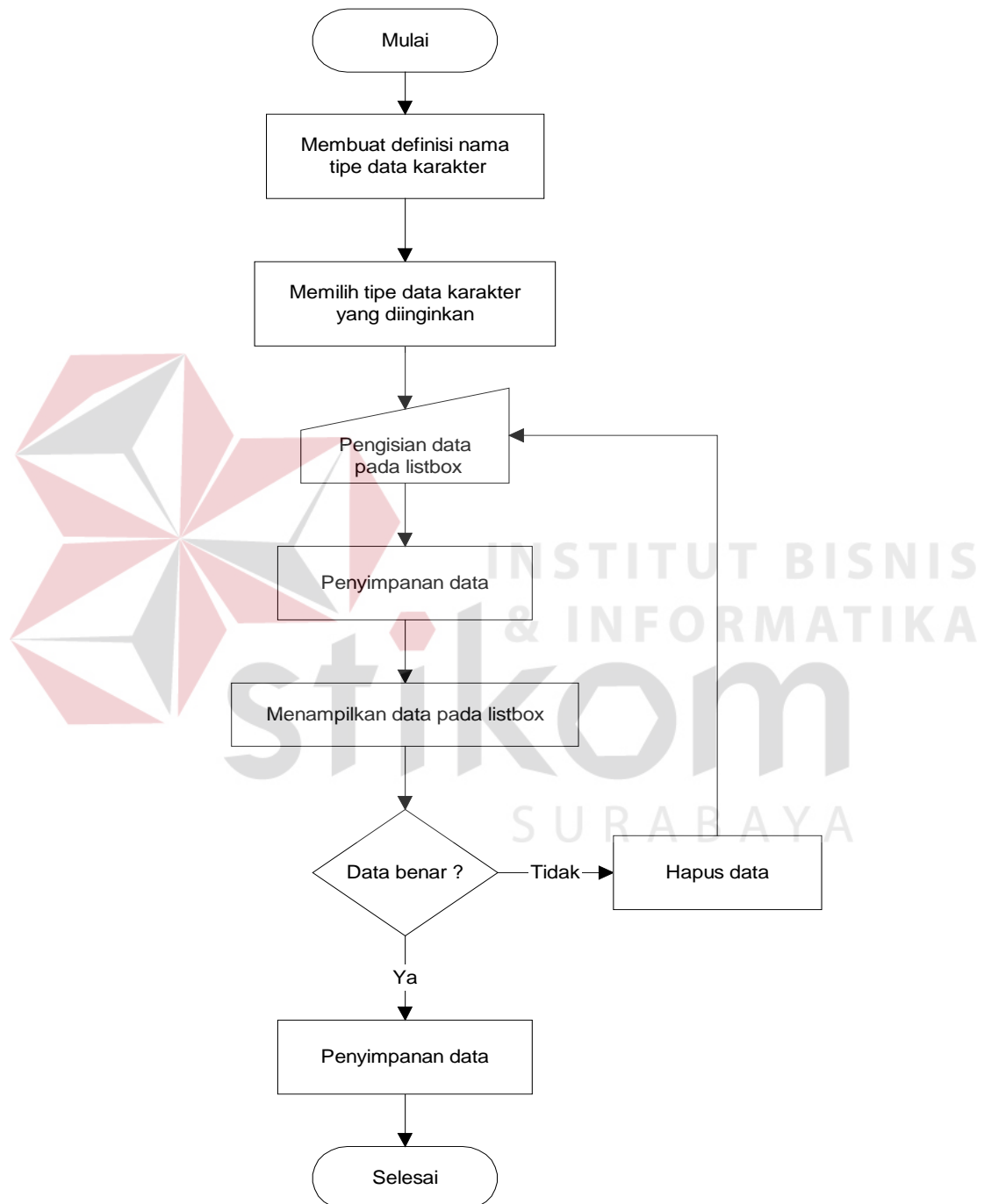
Gambar 3.2. *Flow chart koneksi database.*

2. *Flow chart* untuk membaca dan menampilkan skema *database*



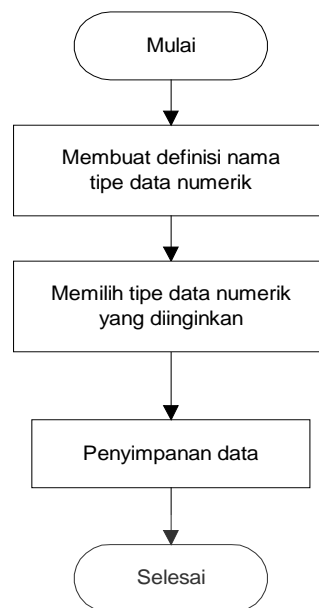
Gambar 3.3. *Flow chart* membaca dan menampilkan skema *database*.

3. *Flow chart* membuat aturan untuk menghasilkan data
- a) *Flow chart* membuat aturan tipe data karakter



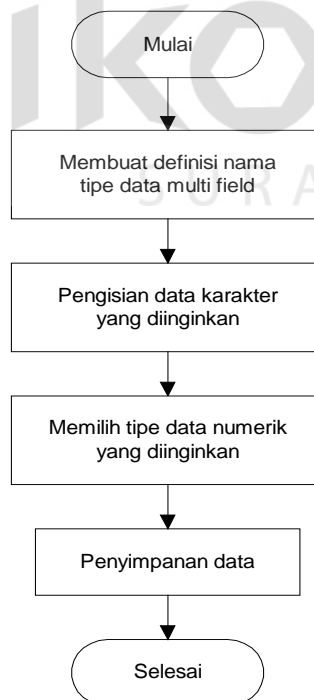
Gambar 3.4. *Flow chart* membuat aturan tipe data karakter.

b) *Flow chart* membuat aturan tipe data numerik



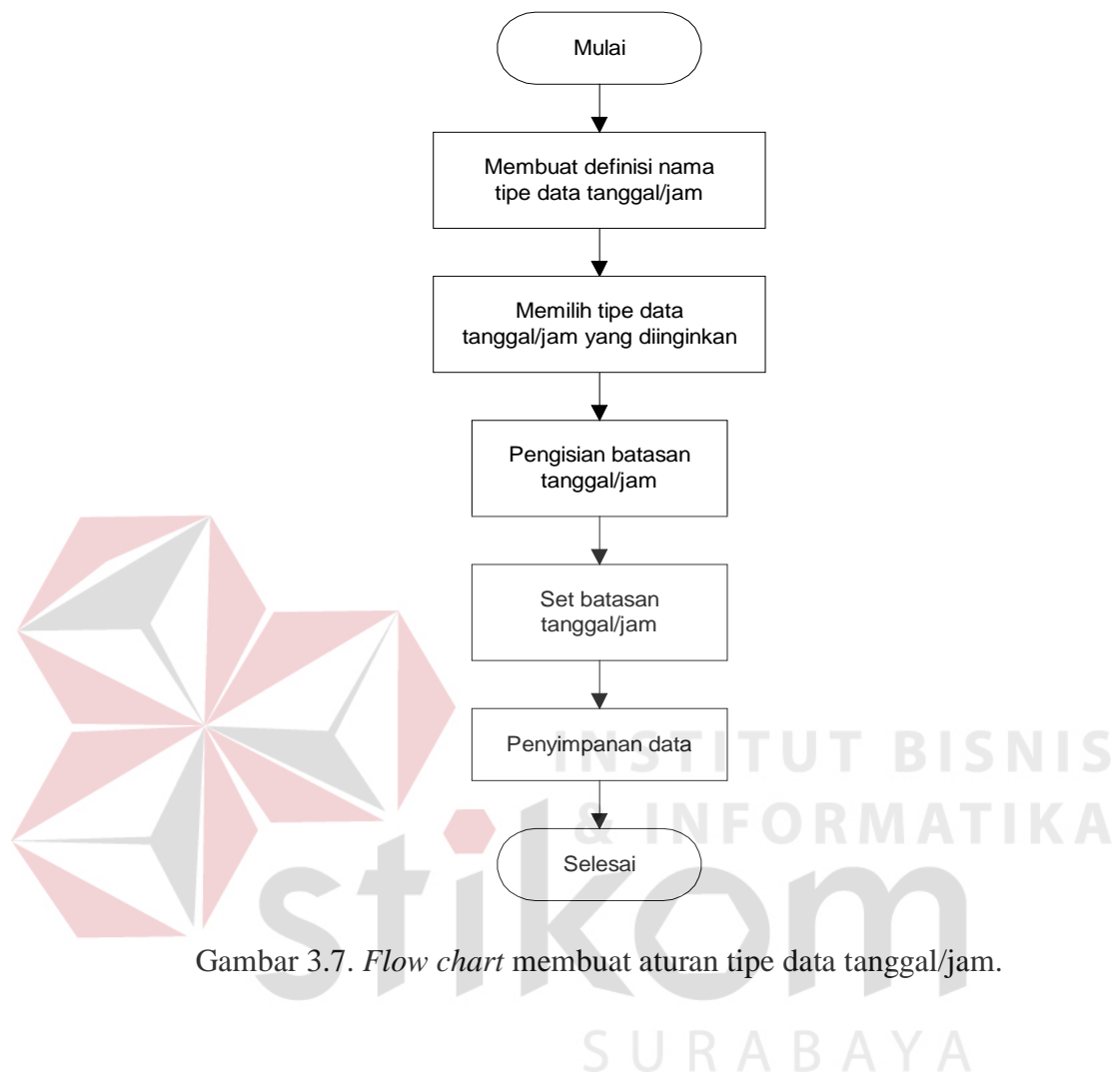
Gambar 3.5. *Flow chart* membuat aturan tipe data numerik.

c) *Flow chart* membuat aturan tipe data multi field



Gambar 3.6. *Flow chart* membuat aturan tipe data multi field.

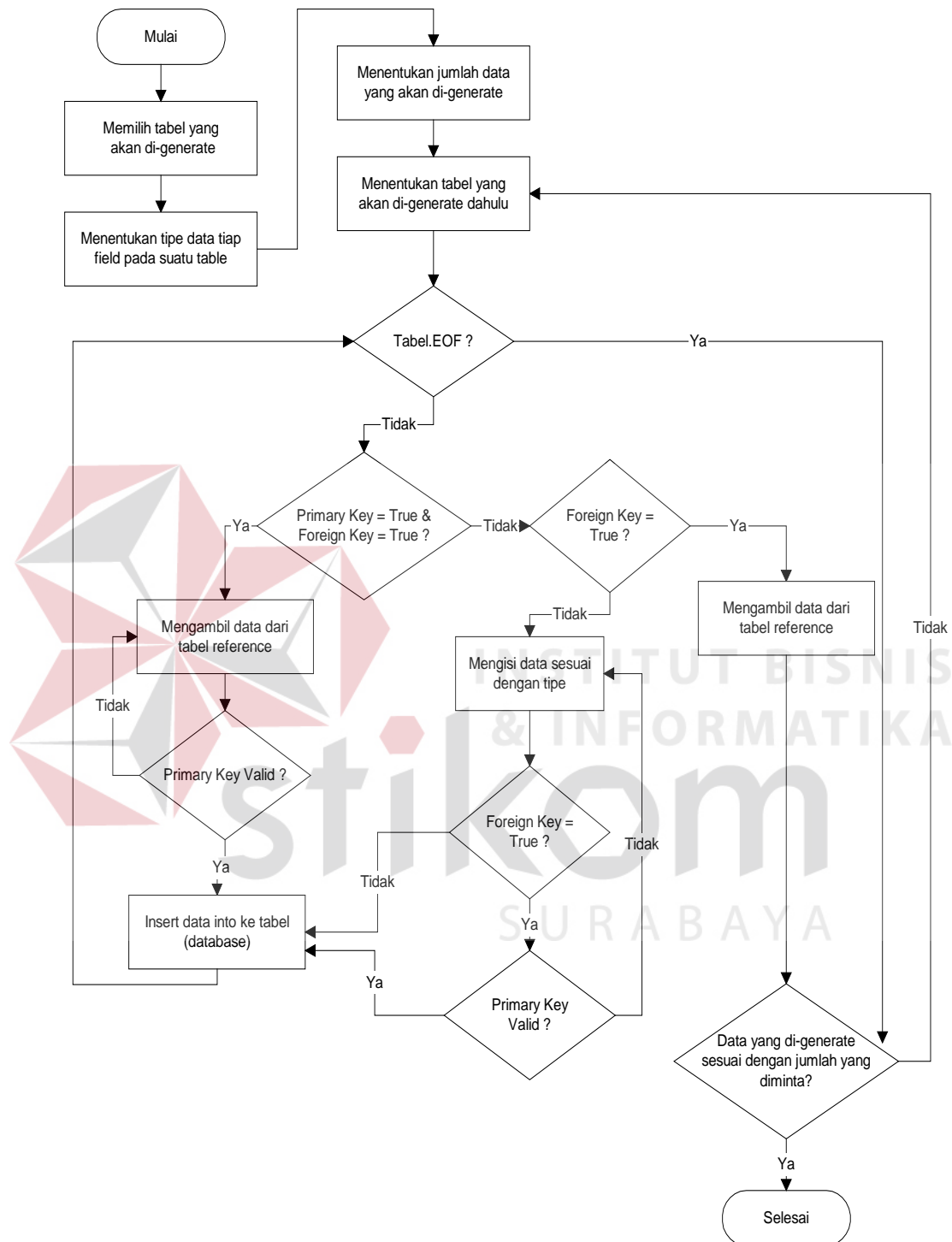
d) *Flow chart* membuat aturan tipe data tanggal/jam



Gambar 3.7. *Flow chart* membuat aturan tipe data tanggal/jam.



4. *Flow chart* menghasilkan data tes sesuai aturan



Gambar 3.8. *Flow chart* menghasilkan (generasi) data tes sesuai aturan.

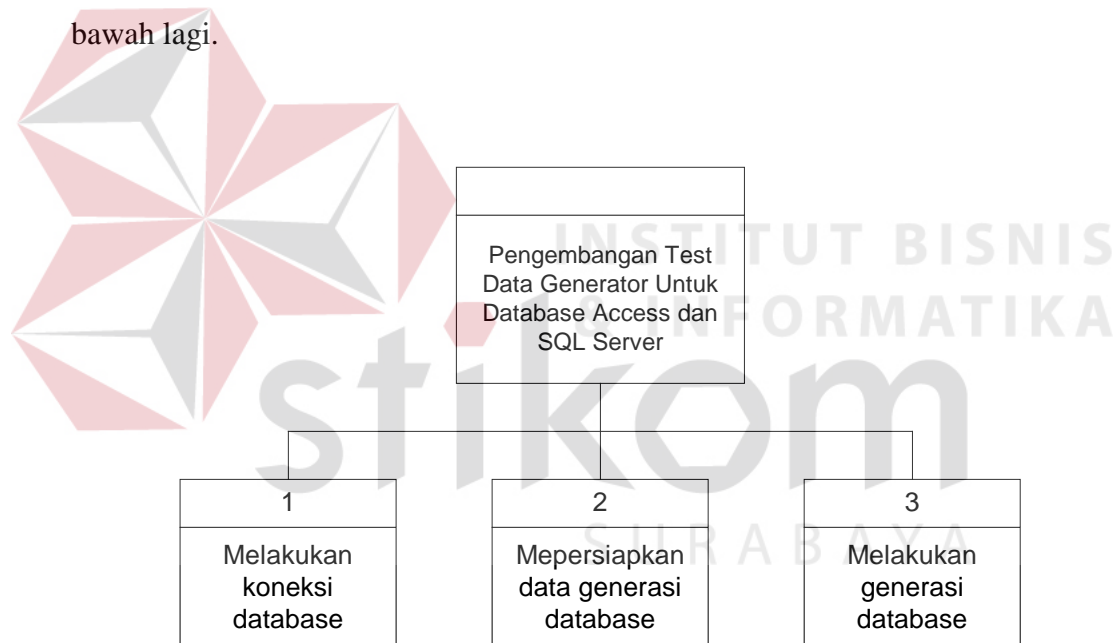
### 3.1.2. Desain Sistem

Untuk membuat suatu desain sistem digunakan model-model yang sudah umum digunakan dan tahap-tahap yang dilakukan dalam mendesain sistem ini, yaitu: Diagram Berjenjang, Data Flow Diagram (DFD), Entity Relational Diagram (ERD), struktur *database*, desain *user interface*.

#### A. Diagram Berjenjang

Diagram berjenjang digunakan untuk mempermudah dan mempersiapkan penggambaran Data Flow Diagram (DFD) ke tingkat-tingkat

bawah lagi.



Gambar 3.9. Diagram berjenjang Pembuatan Test Data Generator untuk Pra-Proses *Data-Driven Testing*.

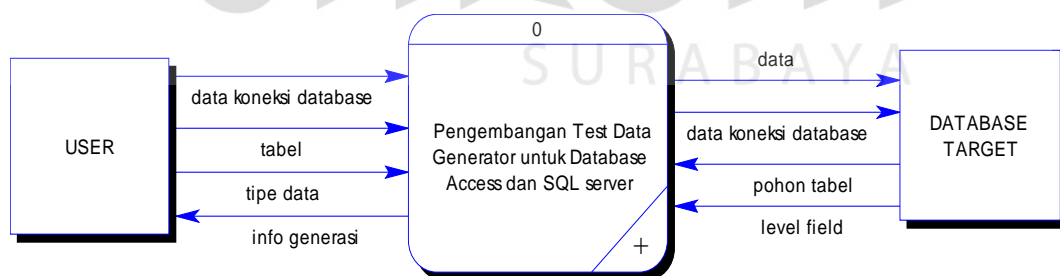
Pada gambar 3.9, tampak bahwa diagram berjenjang tersebut memiliki dua tingkat (*level*), yaitu:

- a) *Context Diagram* adalah *level* yang paling tinggi, dimana terdapat satu bagan sistem utama dari *test data generator*.
- b) *Level 0* diperoleh dari *Context Diagram* yang terdiri dari tiga bagan, yaitu: (1) Melakukan koneksi *database*, (2) Mempersiapkan data generasi *database*, dan (3) Melakukan generasi *database*.

## B. Data Flow Diagram

Data Flow Diagram (DFD) adalah suatu diagram yang bermanfaat untuk menganalisa suatu sistem yang ditinjau dari arus data yang ada dalam suatu sistem. Langkah pertama dalam pembuatan diagram alur ini adalah membuat *context diagram* untuk memberikan gambaran secara keseluruhan, kemudian sistem yang masih bersifat global dijelaskan hingga menjadi sub-sub yang lebih kecil dan terperinci.

### B.1. Context Diagram

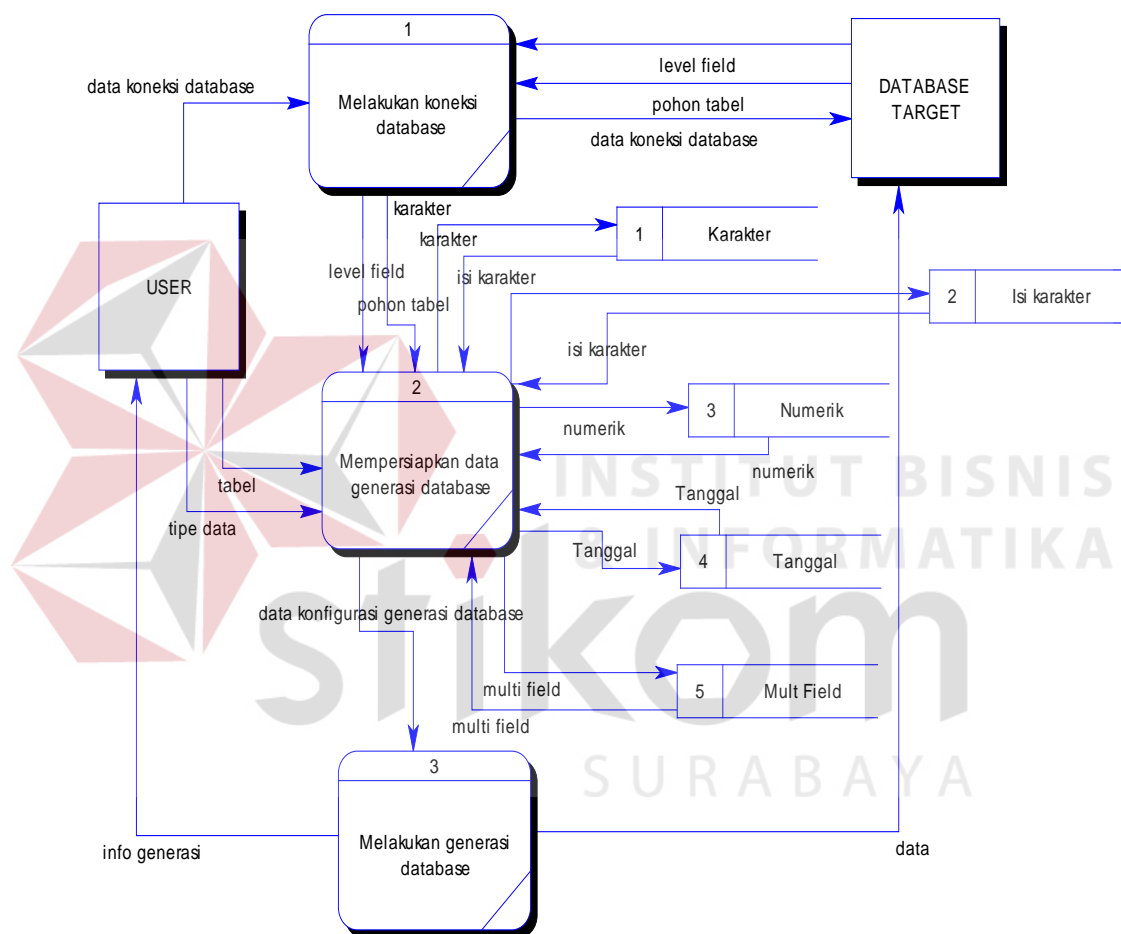


Gambar 3.10. *Context Diagram* Pembuatan Test Data Generator untuk Pra-Proses *Data-Driven Testing*.

Pada gambar 3.10 terdapat satu proses (Pengembangan Test Data Generator untuk *Database Access* dan *SQL Server*) dengan dua *entity*, yaitu:

User, dan *Database* target. Dalam gambar tersebut menggambarkan sistem secara global.

## B.2. DFD Level 0



Gambar 3.11. DFD Level 0 Pembuatan Test Data Generator untuk Pra-Proses *Data-Driven Testing*.

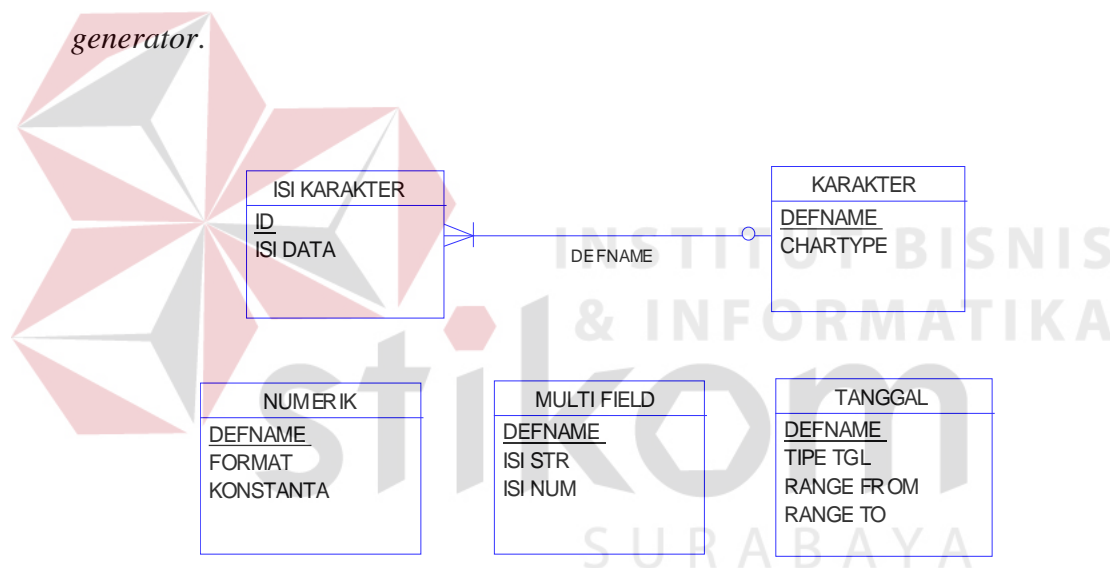
Gambar 3.11 merupakan detail dari proses Pengembangan Test Data Generator Untuk *Database Access* dan *SQL Server*. Dalam gambar tersebut terdapat tiga proses untuk menjelaskan detail dari *context diagram*, yaitu proses

- (1) Melakukan koneksi *database*,
- (2) Mempersiapkan data generasi *database*,
- (3) Melakukan generasi *database*.

### C. Entity Relational Diagram

*Entity Relational (ER) Diagram* digunakan untuk menginterpretasikan, mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan *database*. ER-Diagram menyediakan bentuk untuk menunjukkan struktur keseluruhan kebutuhan data dari *user*.

Berikut ini merupakan ER-Diagram yang digunakan dalam *test data generator*.



Gambar 3.12. ER-Diagram Test Data Generator.

### D. Struktur Database

Dalam pembuatan *test data generator* tidak terlepas dari penggunaan *database* untuk mengambil dan menyimpan data dari hasil proses *generate* maupun data yang dibutuhkan. Di dalam *database* ini terdapat beberapa table diantaranya sebagai berikut:

### D.1. Tabel Karakter

Nama Tabel : Karakter  
 Primary Key : DefName  
 Fungsi : Menyimpan data karakter

Tabel 3.1. Tabel Karakter.

No.	Key	Nama Field	Tipe Data	Lebar Field
1.	Primary Key	DefName	Text	50
2.		CharType	Text	50

Tabel karakter (tabel 3.1), digunakan untuk menyimpan *database* dari tipe-tipe data karakter yang ada. *Field* DefName merupakan definisi nama dari suatu karakter. Sedangkan *field* CharType merupakan tipe karakter dari *field* DefName.

### D.2. Tabel Isi Karakter

Nama Tabel : Isi Karakter  
 Primary Key : Id  
 Fungsi : Menyimpan data isi karakter

Tabel 3.2. Tabel Isi Karakter.

No.	Key	Nama Field	Tipe Data	Lebar Field
1.	Primary Key	Id	Auto Number	Long Integer

Tabel 3.2. Tabel Isi Karakter (lanjutan)

No.	Key	Nama Field	Tipe Data	Lebar Field
2.		IsiData	Text	50
3.		DefName	Text	50

Tabel isi karakter (tabel 3.2), digunakan untuk menyimpan *database* dari isi karakter yang terdapat suatu karakter. *Field* Id merupakan nomor urut yang otomatis pada waktu menyimpan di *database*. *Field* IsiData merupakan isi data dari Defname, sedangkan *field* DefName merupakan definisi nama dari suatu isi karakter.

### D.3. Tabel Numerik

Nama Tabel : Numerik  
 Primary Key : DefName  
 Fungsi : Menyimpan data numerik

Tabel 3.3. Tabel Numerik.

No.	Key	Nama Field	Tipe Data	Lebar Field
1.	Primary Key	DefName	Text	50
2.		Format	Text	50
3.		Konstanta	Number	Long Integer

Tabel Numerik (tabel 3.3), digunakan untuk menyimpan *database* dari tipe-tipe data numerik yang ada. *Field* DefName merupakan nama dari suatu data

numerik. *Field* Format merupakan format dari data numerik. Sedangkan *field* Konstanta merupakan tipe numerik yang sumber datanya bukan *test data type*.

#### D.4. Tabel Tanggal/Jam

Nama Tabel : Tanggal/Jam

Primary Key : Id

Fungsi : Menyimpan data tanggal/jam

Tabel 3.4. Tabel Tanggal/Jam.

No.	Key	Nama Field	Tipe Data	Lebar Field
1.	Primary Key	DefName	Text	50
2.		TipeTgl	Text	50
3.		RangeFrom	Text	15
4.		RangeTo	Text	15

Tabel Tanggal/Jam (tabel 3.4), digunakan untuk menyimpan *database* dari tipe-tipe data tanggal atau jam yang ada. *Field* DefName merupakan nama dari suatu data tanggal atau jam. *Field* Tipe Tgl merupakan format dari tanggal atau jam. *Field* RangeFrom merupakan batas awal tanggal/jam yang diinginkan, sedangkan *field* RangeTo merupakan batas akhir tanggal/jam yang diinginkan.

#### D.5. Tabel Multi *Field*

Nama Tabel : Multi *Field*

Primary Key : DefName

Fungsi : Menyimpan data multi *field*



Tabel 3.5. Tabel Multi *Field*.

No.	Key	Nama Field	Tipe Data	Lebar Field
1.	Primary Key	DefName	Text	50
2.		IsiStr	Text	50
3.		IsiNum	Text	50

Tabel Multi Field (tabel 3.5), digunakan untuk menyimpan *database* dari tipe-tipe data yang *multi field* (tipe data campuran). *Field* DefName merupakan nama dari suatu data *multi field*. *Field* IsiStr merupakan format isi data string, sedangkan *field* IsiNum merupakan format isi data numerik.

### E. Desain *User Interface*

Desain *user interface* merupakan suatu gambaran tentang bentuk tampilan yang dibuat pada program *test data generator*.

Pada desain *form* koneksi *database* (gambar 3.13), *field-field*nya tidak harus ada. Pada koneksi *database Access* hanya ada *field input* nama *database*, *input* nama *user*, *input password*, *input* nama *server* tidak ada. Pada koneksi *database SQL Server* semua *field* ada tetapi tombol pencarian *database* tidak ada.

Judul : Merupakan judul dari tugas akhir ini.

*Database* : Merupakan tombol untuk koneksi ke *database*.

*Generate* : Merupakan tombol untuk *generate database*.

Keluar : Merupakan tombol untuk keluar dari sistem.

Daftar Koneksi : Merupakan daftar pilihan dari nama-nama *database* yang akan dipilih.

Tombol Cari *Database* : Merupakan tombol untuk mencari *database* yang akan digunakan.

Input Nama *Server* : Merupakan *entry* text nama *server*.

Input Nama *User* : Merupakan *entry* text nama *user*.

Input *Password* : Merupakan *entry* text *password*.

Input Nama *Database* : Merupakan *entry* text nama *database*.

Tombol OK : Merupakan tombol untuk melakukan koneksi ke *database*.

Tombol Cancel : Merupakan tombol untuk membatalkan koneksi ke *database*.

The diagram illustrates the layout of a database connection form. It features a main window with a title bar labeled 'Judul'. On the left side of the window, there are three vertically stacked buttons: 'Tombol Database', 'Tombol Generate', and 'Tombol Keluar'. The central area of the window is titled 'Database' and contains several elements: a 'Daftar Pilihan Koneksi' button at the top left, followed by four input fields stacked vertically: 'Input Nama Server', 'Input Nama User', 'Input Password', and 'Input Nama Database'. To the right of the 'Daftar Pilihan Koneksi' button is a 'Tombol Cari Database' button. At the bottom of the 'Database' section, there are two buttons: 'Tombol OK' and 'Tombol Batal'.

Gambar 3.13. Desain *form* Koneksi *Database*.

Gambar 3.14. Desain *form* Cari *Database*.

Di bawah ini merupakan keterangan dari gambar desain *form* cari *database* (gambar 3.14).

- Daftar Pilihan Drive : Merupakan daftar nama-nama *drive* yang ada.
- List nama *Directory* : Merupakan list nama-nama *directory* yang terdapat pada sistem.
- List Nama File : Merupakan list nama-nama file yang ada pada sistem.
- Nama File : Nama file *database* yang akan di koneksikan.
- Tombol OK : Merupakan tombol untuk menyetujui nama *database* yang akan dikoneksikan.
- Tombol Cancel : Merupakan tombol untuk membatalkan nama *database* yang akan dikoneksikan.

Tombol Pohon Tabel	Tombol Level Field	Tombol Generate	Tombol Karakter	Tombol Numerik	Tombol Tanggal/Jam	Tombol Multi Field
<div style="display: flex; justify-content: space-between; align-items: center; height: 200px;"> <div style="text-align: center; flex-grow: 1;"> <p>Lembar Kerja</p> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Tombol Baru</div> <div style="border: 1px solid black; padding: 5px;">Tombol Hapus</div> </div>						

Gambar 3.15. Desain *form* Lembar Kerja (*form* utama).

Di bawah ini merupakan keterangan dari gambar desain *form* lembar kerja (gambar 3.15) yang juga merupakan *form* utama.

Judul : Merupakan judul dari tugas akhir ini.

Tombol *Database* : Merupakan tombol untuk koneksi ke *database*.

Tombol *Generate* : Merupakan tombol untuk *generate database*.

Tombol Keluar : Merupakan tombol untuk keluar dari sistem.

Tombol Pohon Tabel : Merupakan tombol untuk menampilkan tabel-tabel dalam bentuk pohon tabel (*tree*).

Tombol Level *Field* : Merupakan tombol untuk menampilkan level (urutan tabel) mana yang akan di-*generate* dulu.

Tombol *To Generate* : Merupakan tombol untuk membuat tabel yang akan di-*generate*.

Tombol Karakter : Merupakan tombol untuk membuat tipe tes data karakter.

Tombol Tanggal : Merupakan tombol untuk membuat tipe tes data tanggal.

Tombol Multi Field : Merupakan tombol untuk membuat tipe tes data yang terdiri dari beberapa tipe tes data.

Lembar kerja : Merupakan halaman *form* yang di aktifkan dari tombol-tombol pada menu utama.

Pada saat tombol *To Generate*, Karakter, Numerik, Tanggal, dan Multi Field ditekan satu per satu, maka akan muncul tombol Baru dan tombol Hapus.

Tombol Baru : Merupakan tombol untuk membuat tipe tes data baru.

Tombol Hapus : Merupakan tombol untuk menghapus tipe tes data.

Jika ingin menambahkan karakter baru maka tekan tombol baru dan muncul *form* Tipe Data Karakter. Demikian juga dengan Tipe Data Numerik, Tipe Data Tanggal, dan Tipe Data Multi *field*.

Di bawah ini merupakan keterangan dari gambar desain *form* tipe data karakter (gambar 3.16).

Judul : Merupakan Judul dari *form*.

Input Definisi Karakter : Merupakan *entry* definisi karakter.

Daftar Pilihan Tipe Karakter : Merupakan daftar pilihan tipe data karakter yang ada.

Tombol Baru : Merupakan tombol untuk menambah tipe data karakter.

- Tombol Batal : Merupakan tombol untuk membatalkan penambahan tipe data karakter dan keluar dari *form*.
- Input Nama : Merupakan *entry* nama pada definisi karakter baru.
- Tombol Tambah : Merupakan tombol untuk menambahkan data dari input tambah ke list tambah.
- Tombol Hapus : Merupakan tombol untuk menghapus data dari list tambah.
- List Nama : Merupakan daftar nama-nama pada definisi karakter.

Judul

Input Definisi Karakter Tombol Baru

Daftar Pilihan Tipe Karakter Tombol Batal

Input Nama Tombol Tambah Tombol Hapus

List Nama

Gambar 3.16. Desain *form* Tipe Data Karakter

Gambar 3.17. Desain *form* Tipe Data Numerik

Di bawah ini merupakan keterangan dari gambar desain *form* tipe data numerik (gambar 3.17).

Judul : Merupakan judul dari *form*.

Input Definisi Numerik : Merupakan *entry* definisi numerik.

Daftar Pilihan Format Numerik : Merupakan daftar pilihan format numerik yang ada.

Tombol Baru : Merupakan tombol untuk menambah format data.

Tombol Batal : Merupakan tombol untuk membatalkan penambahan format data dan keluar dari *form* ini.

Gambar 3.18. Desain *form* Tipe Data Tanggal/Jam

Di bawah ini merupakan keterangan dari gambar desain *form* tipe data tanggal/jam (gambar 3.18).

- Judul : Merupakan judul dari *form*.
- Input Definisi Tanggal : Merupakan *entry* nama defnisi tanggal.
- Daftar Pilihan Format Tanggal : Merupakan list dari format tanggal yang ada.
- Input Tanggal Awal : Merupakan *entry* tanggal mulai yang diinginkan.
- Input Tanggal Akhir : Merupakan *entry* tanggal akhir yang diinginkan.
- Tombol Baru : Merupakan tombol untuk menambahkan format tanggal.
- Tombol Batal : Merupakan tombol untuk membatalkan penambahan atau pengubahan format tanggal dan keluar dari *form* ini.



Gambar 3.19. Desain *form* Tipe Data Multi *Field*

Di bawah ini merupakan keterangan dari gambar desain *form* tipe data multi *field* (gambar 3.19).

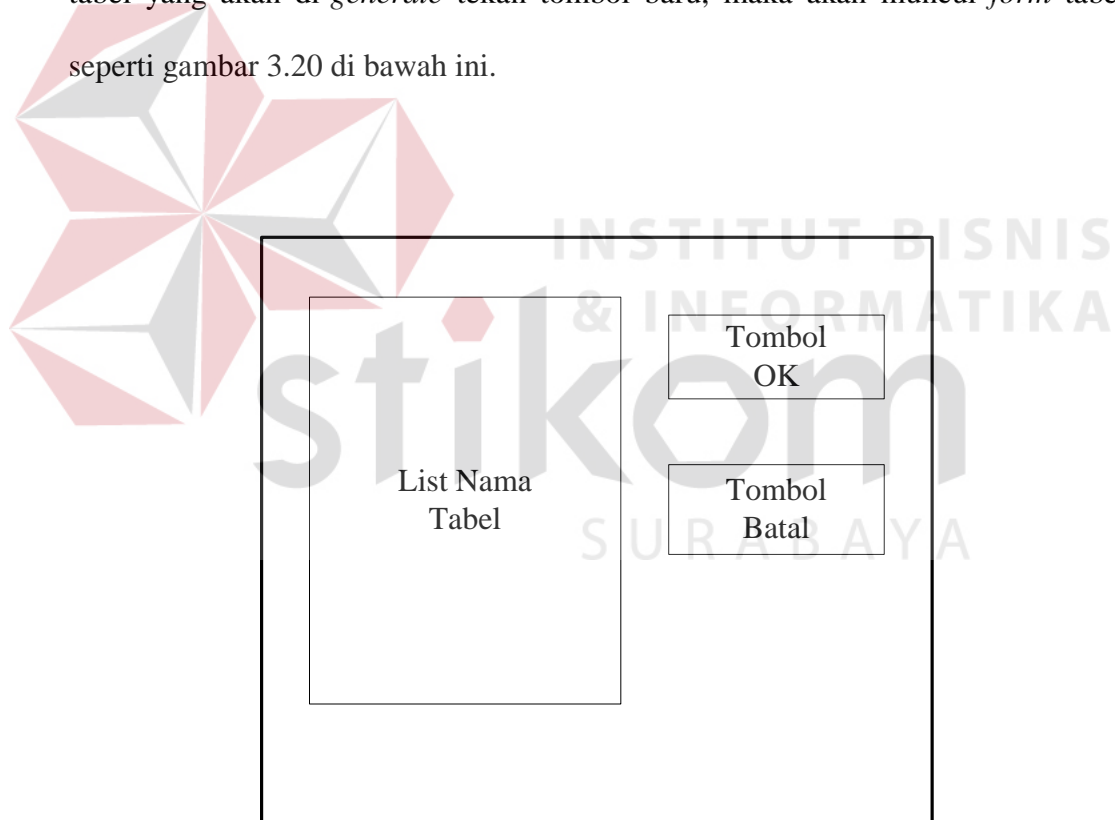
- Judul : Merupakan judul dari *form*.
- Input Definisi Multi Field : Merupakan *entry* nama definsi multi *field*.
- Input Isi String : Merupakan *entry* string yang akan disimpan untuk *database*.
- Daftar Pilihan Isi Numerik : Merupakan list dari isi numerik yang ada.
- Tombol Baru : Merupakan tombol untuk menambahkan atau mengubah format tanggal.
- Tombol Batal : Merupakan tombol untuk membatalkan penambahan tipe data multi *field* dan keluar dari *form* ini.

Jika tombol Karakter, Numerik, Tanggal, dan Multi *Field* ditekan satu per satu, maka pada masing-masing lembar kerja akan muncul nama dari masing-

masing tipe yang baru. Dan bila nama tipe tersebut diklik dua kali, maka akan muncul *form* masing-masing tombol (Karakter, Numerik, Tanggal, dan Multi *Field*) tetapi tombol baru yang ada dalam *form* tersebut akan berubah menjadi tombol Ubah.

Tombol Ubah : Merupakan tombol untuk mengubah isi yang ada dalam *form* tersebut.

Pada waktu menekan tombol *To Generate*, pada lembar kerja terdapat keterangan nama tabel, ada join, dan jumlah *field*. Jika ingin mengisikan nama tabel yang akan di-*generate* tekan tombol baru, maka akan muncul *form* tabel seperti gambar 3.20 di bawah ini.



Gambar 3.20. Desain *form* Tabel.

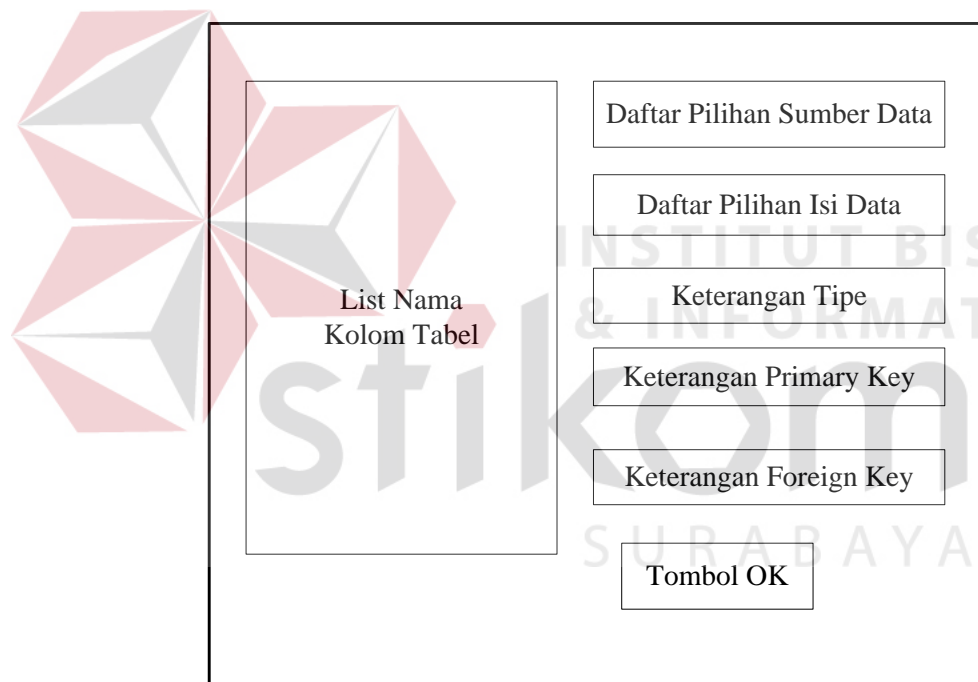
Di bawah ini merupakan keterangan dari gambar desain *form* tabel (gambar 3.20).

List Nama Tabel : Merupakan daftar nama tabel yang akan di-*generate*.

Tombol OK : Merupakan tombol untuk menyetujui nama tabel yang telah dipilih yang akan di-*generate*.

Tombol Batal : Merupakan tombol untuk membatalkan pemilihan tabel yang akan di-*generate* dan keluar dari *form* ini.

Kemudian muncul *form field setting* yang berisi list nama kolom dari tabel, sumber data, isi data, dan keterangan tipe. Keterangan tipe ini antara lain: *Primary Key*, dan *Foreign Key* yang muncul secara otomatis.



Gambar 3.21. Desain *form Field Setting*

Di bawah ini merupakan keterangan dari gambar desain *form field setting* (gambar 3.21).

List Nama Kolom Tabel : Merupakan list nama kolom pada tabel yang akan di-*generate*.

- Daftar Pilihan Sumber Data : Merupakan daftar pilihan sumber data yang berupa konstanta atau *test data type*.
- Daftar Pilihan Isi Data : Merupakan daftar isi data yang ada yang sesuai dengan tipe-tipe kolom pada suatu tabel.
- Tombol OK : Merupakan tombol untuk menyetujui isi *form* properti yang telah dibuat dan yang akan di-*generate*.

Apabila nama tabel yang akan di-*generate* sudah muncul di lembar kerja, maka tekan tombol *Generate* untuk men-*generate* data pada tabel tersebut sehingga muncul *form Generate* seperti di bawah ini.

Gambar 3.22. Desain *form Generate Data*.

Di bawah ini merupakan keterangan dari gambar desain *form generate* data (gambar 3.22).

- Jumlah Data yang akan di-*generate* : Merupakan jumlah data yang akan di-*generate* pada suatu tabel.
- Tombol *Generate* : Merupakan tombol untuk men-*generate* data pada suatu tabel.

Tombol Batal : Merupakan tombol untuk membatalkan proses *generate* data pada suatu tabel.

### 3.1.3. Pembuatan Aplikasi

Pada pembuatan aplikasi “Pengembangan Data Test Generator untuk Database Access dan SQL Server“ ini menggunakan bahasa pemrograman Visual Basic 6.0 yang hasil *Outputnya* adalah *source code* dapat dilihat pada lampiran.

### 3.1.4. Testing

Untuk menunjang sistem ini, metode yang digunakan adalah *blackbox testing*. Metode *blackbox testing* ini berfungsi untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi tanpa pengetahuan detail struktur internal dari sistem atau komponen yang di tes. *Blackbox testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.

*Functional analysis* dan *risk based testing* dimana fungsi-fungsi pada aplikasi tidak semuanya akan di-tes, namun dianalisa terlebih dahulu berdasarkan prioritas penggunaan dan resiko dari fungsi-fungsi tersebut.

## 3.2. Rancangan Evaluasi

Pada sistem ini fungsi-fungsi yang mendapatkan prioritas adalah sebagai berikut:

### a) Koneksi *database Microsoft Access* dan *SQL Server*

Pada *test case* (uji coba) koneksi *database* ini bertujuan untuk memastikan sistem dapat melakukan koneksi ke sistem *database Access* dan *SQL Server*. Dimana masing-masing menggunakan *database* Biblio untuk *Microsoft Access*, dan *database* Nothwind untuk *SQL Server*.

b) Pembacaan pohon tabel

Pada *test case* pembacaan pohon tabel ini bertujuan untuk memastikan bahwa pohon tabel yang dibuat sesuai dengan *database* target (tujuan). Pembacaan pohon tabel menggunakan uji komparasi, dimana pohon tabel yang digunakan oleh sistem akan dibandingkan dengan Erwin 4.1. *Database* yang digunakan adalah *database* Bibilio untuk Access, sedangkan *database* Northwind untuk SQL Server.

c) Generate data

Pada *test case* generate data bertujuan untuk memastikan bahwa data yang di-generate sesuai dengan yang data di-entry ke dalam model masukan. *Test case* ini menggunakan metode *equivalence partitioning* yang masing-masing *database* akan diuji generasi data terhadap empat macam model masukan (karakter/string, numerik, tanggal/jam, dan *multifield*). *Database* yang digunakan adalah Bibilio untuk Microsoft Access dan Northwind untuk SQL Server.

Tabel di bawah ini merupakan rincian dari beberapa *test case* yang dilakukan oleh sistem.

Tabel 3.6. Rincian evaluasi *test case* pada system.

No.	Tujuan	Input	Output yang diharapkan	Keterangan
1.	Memastikan sistem dapat melakukan koneksi ke sistem <i>database</i> Access.	Data koneksi <i>database</i> Access (Bibilio).	Pohon tabel Bibilio.mdb yang ada pada <i>form</i> lembar kerja pohon tabel.	<i>Test case</i> koneksi <i>database</i> Access.

Tabel 3.6. Rincian *test case* pada sistem (lanjutan).

No.	Tujuan	Input	Output yang diharapkan	Keterangan
2.	Memastikan sistem dapat melakukan koneksi ke sistem <i>database SQL Server</i> .	Data koneksi <i>database SQL Server</i> (Northwind).	Pohon tabel Northwind.ldb yang ada pada <i>form</i> lembar kerja pohon tabel.	<i>Test case</i> koneksi <i>database SQL Server</i> .
3.	Memastikan bahwa pohon tabel yang dibuat sesuai dengan <i>database target</i> (tujuan).	Tabel yang ada pada <i>database Access</i> (Bibilio).	Pohon tabel Bibilio.mdb yang ada pada <i>form</i> lembar kerja pohon tabel (gambar 4.25).	<i>Test case</i> pembacaan pohon tabel <i>database Access</i> .
4.	Memastikan bahwa pohon tabel yang dibuat sesuai dengan <i>database target</i> (tujuan).	Tabel yang ada pada <i>database SQL server</i> (Northwind).	Pohon tabel Nothwind.ldb yang ada pada <i>form</i> lembar kerja pohon tabel (gambar 4.26).	<i>Test case</i> pembacaan pohon tabel <i>database SQL Server</i> .
5.	Memastikan bahwa data yang di-generate sesuai dengan yang data di-entry ke dalam model masukan.	Data-data model masukan karakter, isi karakter, numerik, tanggal/jam, dan multi <i>field</i> (tabel 3.1 - tabel 3.5) .	Data generasi sesuai dengan model masukan karakter, isi karakter, numerik, tanggal/jam, dan multi <i>field</i> (gambar 4.29).	<i>Test case</i> generate data pada <i>database Access</i> .

Tabel 3.6. Rincian *test case* pada sistem (lanjutan).

No.	Tujuan	Input	Output yang diharapkan	Keterangan
6.	Memastikan bahwa data yang di-generate sesuai dengan yang data di-entry ke dalam model masukan.	Data-data model masukan karakter, isi karakter, numerik, tanggal/jam, dan multi <i>field</i> (tabel 3.1 - tabel 3.5).	Data generasi sesuai dengan model masukan karakter, isi karakter, numerik, tanggal/jam, dan multi <i>field</i> (gambar 4.30).	<i>Test case</i> generate data pada <i>database</i> SQL Server.

