

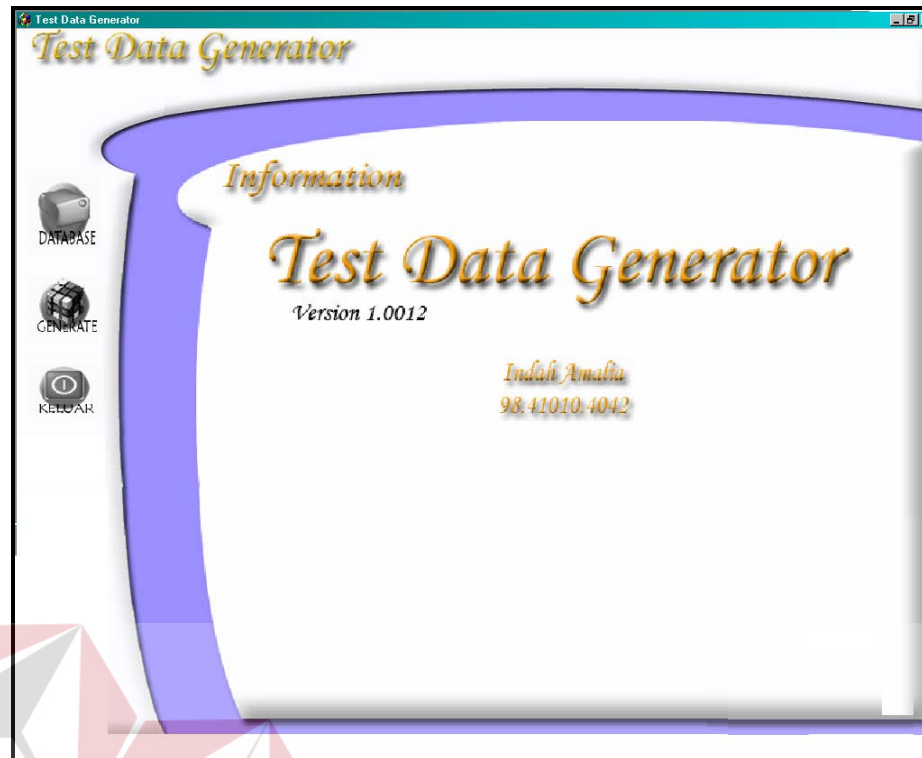
## BAB IV

### IMPLEMENTASI DAN PEMODELAN

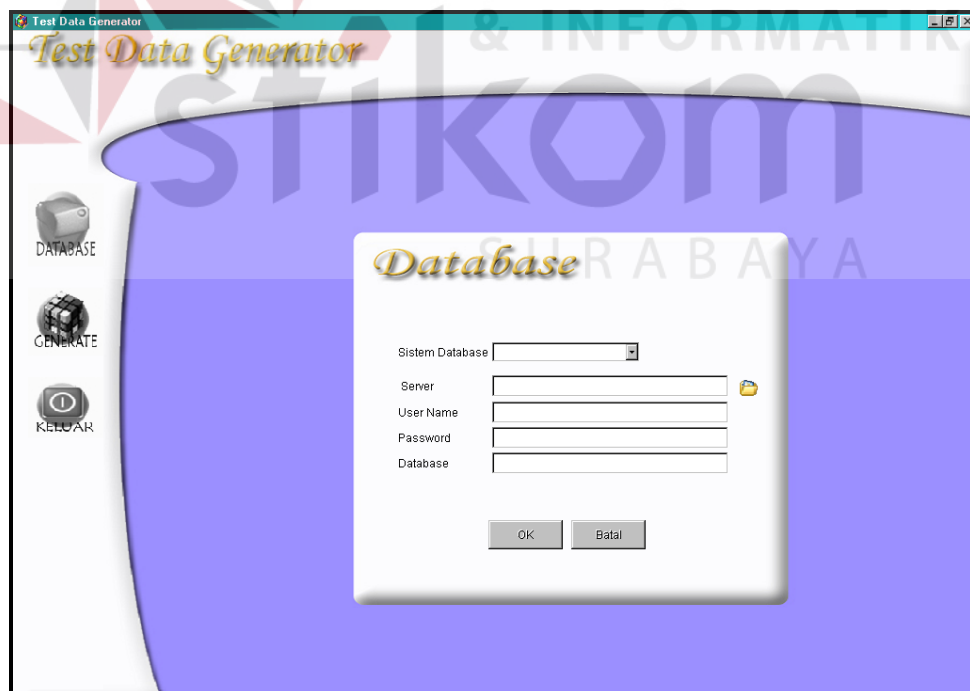
Pada bab ini membahas tentang implementasi program, peralatan yang digunakan, dan juga evaluasi *output*. Implementasi program menjelaskan tentang urutan kerja sistem dari pertama kali koneksi ke *database* sampai proses generasi data. Sedangkan evaluasi *output* menjelaskan tentang desain uji coba dan subyek coba, jenis data dan instrumen pengumpul data, serta analisis hasil uji coba.

#### 4.1. Implementasi Program

Pada waktu program pertama kali dijalankan, *form* yang pertama kali muncul adalah *form* judul (gambar 4.1) yang berisi judul, tombol *database*, tombol *generate*, dan tombol keluar. Tombol *database* untuk memilih sistem *database* yang akan digunakan (gambar 4.2), tombol *generate* digunakan untuk melakukan generasi data (gambar 4.22), sedangkan tombol keluar digunakan untuk keluar dari aplikasi.

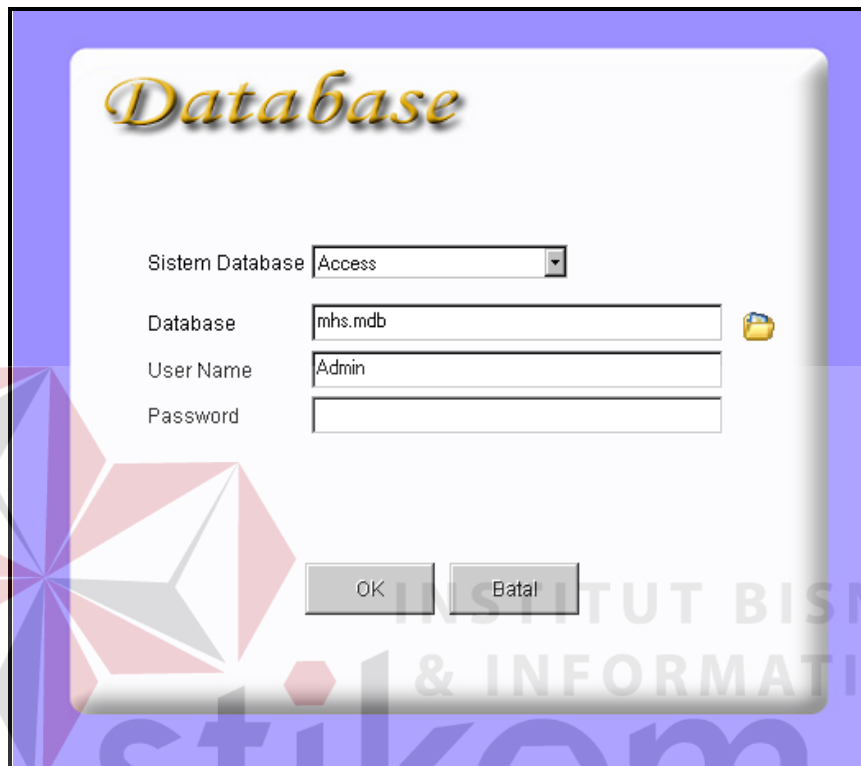


Gambar 4.1. Form judul.



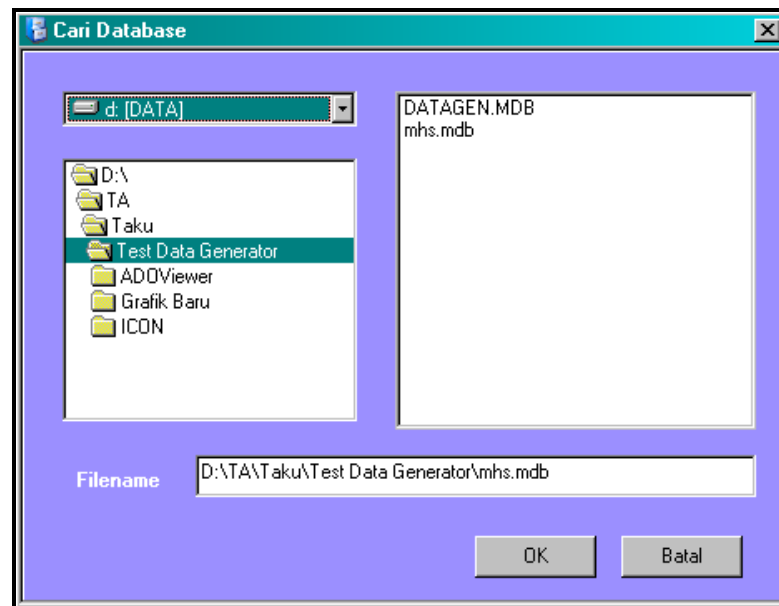
Gambar 4.2. Form koneksi database

Untuk melakukan koneksi *database* user dapat memilih sistem *database* yang akan digunakan. Sistem *database* ini terdapat dua pilihan, yaitu *database Access*, dan *SQL Server* seperti yang ditunjukkan gambar di bawah ini.



Gambar 4.3. *Form koneksi database Access.*

Pada sistem *database Access*, *user* dapat memilih *database* dengan cara meng-klik ikon yang ada disamping inputan nama *database* sehingga muncul *form* cari *database* (gambar 4.4) di bawah ini, yang berfungsi untuk mencari *file databasenya*.



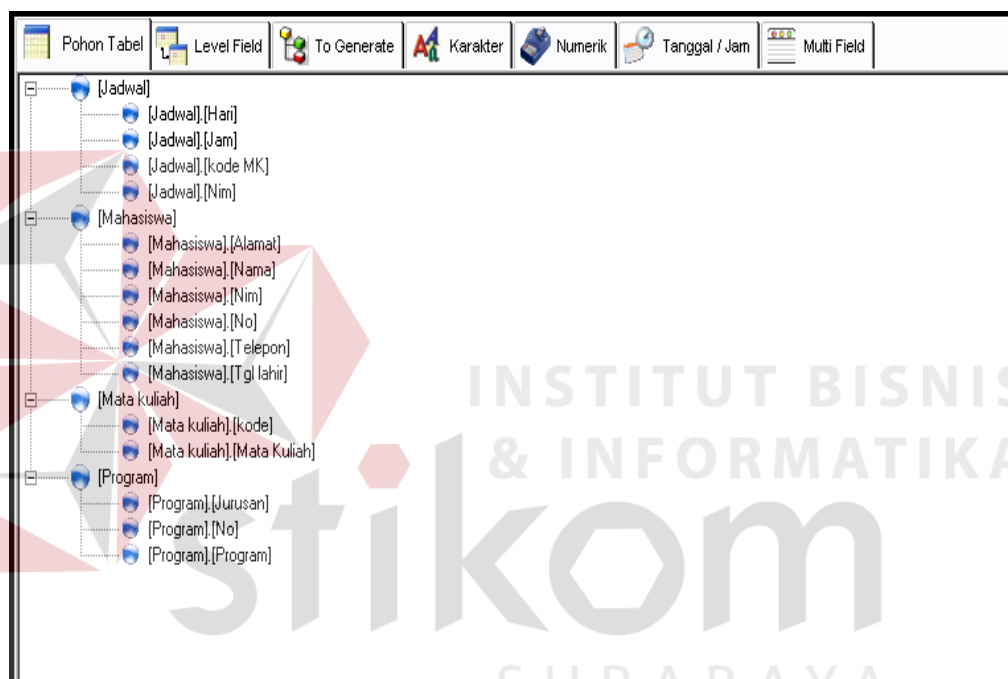
Gambar 4.4. *Form cari database.*

Sedangkan pada sistem *database SQL Server 7*, *database* diisi secara manual dan password dapat dikosongi.

Gambar 4.5. *Form koneksi database SQL Sever 7.*

Setelah *database* dipilih atau diisi, maka klik tombol OK dan kemudian menuju ke *form* lembar kerja pohon tabel seperti gambar 4.6.

Pada *form* lembar kerja pohon tabel (*treeview*). Skema *database* langsung tersusun sesuai dengan tabel yang ada dalam *database* yang dipilih. Begitu juga dengan level *field*, yang akan muncul secara otomatis pada lembar kerja level *field* seperti gambar 4.7.



Gambar 4.6. *Form* lembar kerja pohon tabel.

kode	Mata Kuliah
2174	Kalkulus
17814	Aljabar
18501	Manajemen
32178	Alpro
21185	Alpro

Gambar 4.7. *Form kembar kerja level field.*

Seperti halnya lembar kerja pohon tabel, *form* lembar kerja level *field* juga menampilkan skema *database*, tetapi untuk *form* level *field* menampilkan urutan nama tabel mana yang harus di-generate terlebih dahulu. Jika nama tabel pada level *field* di-klik maka akan muncul data-data yang ada pada tabel tersebut.

Gambar 4.8. *Form* lembar kerja tipe data karakter.

Pada lembar kerja tipe data karakter (gambar 4.8) di atas, berfungsi untuk menampilkan aturan tipe data karakter yang dibuat. Jika tombol baru di klik maka akan muncul *form* tipe data karakter seperti gambar 4.9 berikut.

Gambar 4.9. *Form* tipe data karakter.

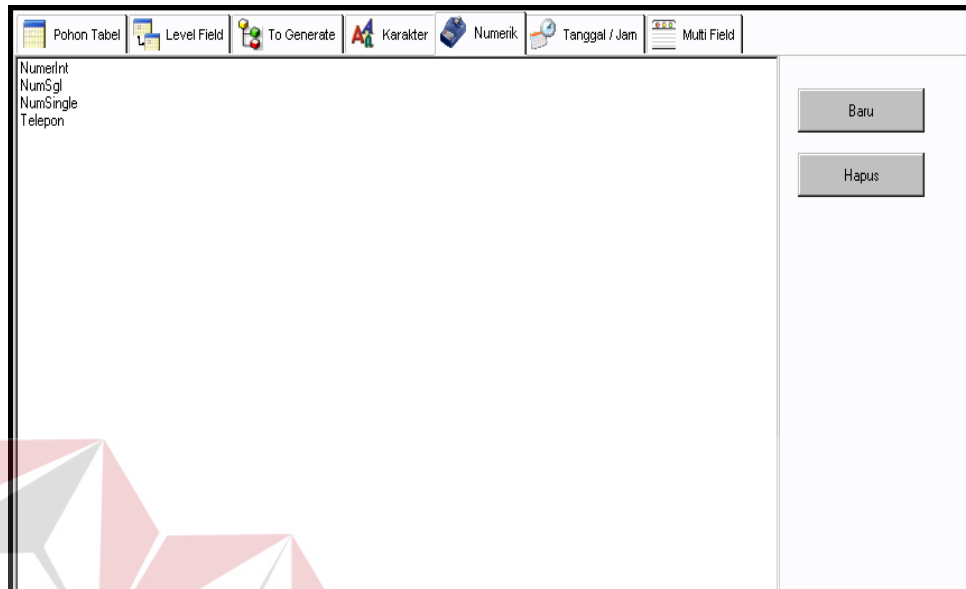
*Form* tipe data karakter (gambar 4.9) berfungsi untuk membuat aturan baru untuk tipe data karakter. Definisi merupakan nama karakter yang dibuat, tipe karakter berisi nama bisnis, nama orang, dan alamat. Sedangkan isi datanya inputkan pada inputan yang ada di sebelah tombol tambah yang kemudian akan ditambahkan pada *list* yang ada di bawahnya.

Gambar 4.10. *Form* propertis tabel dengan *field* yang bertipe data karakter.

*Form* propertis tabel dengan *field* yang bertipe data karakter (gambar 4.10) berfungsi untuk mengisi sumber data dan isi data *field-field* yang ada pada tabel yang akan di-*generate* pada lembar kerja *to generate* (gambar 4.20). Pilih dulu salah satu atribut tabel yang ada, kemudian pilih sumber data yang diinginkan (konstanta atau *test data type*). Jika sumber data adalah *test data type* maka pilih isi data yang ada, dan jika sumber data konstanta maka isi sesuai keinginan. Sedangkan keterangan tipe merupakan tipe dari *field* yang dipilih, keterangan *primary key* (PK) merupakan keterangan apakah *field* tersebut PK atau



bukan, dan keterangan *foreign key* merupakan keterangan apakah *field* tersebut mengacu ke *field* yang ada pada tabel lain (FK).



Gambar 4.11. *Form* lembar kerja tipe data numerik.

Sama seperti *form* lembar kerja tipe data karakter (gambar 4.8), pada *form* lembar kerja tipe data numerik (gambar 4.11) di atas berfungsi untuk menampilkan aturan tipe data numerik yang dibuat. Jika tombol baru di klik maka akan muncul *form* tipe data numerik seperti gambar 4.12 berikut.

Gambar 4.12. *Form tipe data numerik.*

*Form* tipe data numerik (gambar 4.12) ini, berfungsi untuk membuat aturan baru untuk tipe data numerik. Definisi merupakan nama tipe data numerik yang dibuat, sedangkan format numerik berisi format data numerik yaitu *integer*, *long integer*, *single*, *double*.

Gambar 4.13. *Form propertis tabel dengan field yang bertipe data numerik.*

Seperti pada *form* properti tabel dengan field yang bertipe data karakter, *form* properti tabel dengan field yang bertipe data numerik (gambar 4.13) berfungsi untuk mengisi sumber data dan isi data *field-field* yang ada pada tabel yang akan di-*generate* pada lembar kerja *to generate*. Pilih dulu salah satu atribut tabel yang ada, kemudian pilih sumber data yang diinginkan (konstanta atau *test data type*). Jika sumber data adalah *test data type* maka pilih isi data yang ada, dan jika sumber data konstanta maka isi sesuai keinginan.



Gambar 4.14. *Form* lembar kerja tipe data tanggal/jam.

Sama seperti *form* lembar kerja sebelumnya, pada *form* lembar kerja tipe data tanggal/jam (gambar 4.14) di atas berfungsi untuk menampilkan aturan tipe data tanggal/jam yang dibuat. Jika tombol baru di klik maka akan muncul *form* tipe data tanggal/jam seperti gambar 4.15 berikut.

Gambar 4.15. *Form* tipe data tanggal/jam.

*Form* tipe data tanggal/jam (gambar 4.15) ini, berfungsi untuk membuat aturan baru untuk tipe data tanggal/jam. Definisi merupakan nama tipe data tanggal/jam yang dibuat, sedangkan format tanggal/jam berisi format data tanggal/jam yaitu *general date*, *long date*, *short date*, *long time*, dan *short time*. Tentukan batas tanggal yang diinginkan yang akan ditampilkan pada *database* yang akan di-generate.

*Form* properti tabel dengan *field* bertipe data tanggal/jam (gambar 4.16) berfungsi untuk mengisi sumber data dan isi data *field-field* yang ada pada tabel yang akan di-generate pada lembar kerja *to generate*. Pilih dulu salah satu atribut tabel yang ada, kemudian pilih sumber data yang diinginkan (konstanta atau *test data type*). Jika sumber data adalah *test data type* maka pilih isi data yang ada, dan jika sumber data konstanta maka isi sesuai keinginan.

The screenshot shows a window titled '[Mahasiswa]'. On the left, a list of fields is displayed: [Alamat], [Nama], [Nim], [No], [Telepon], and [Tgl lahir]. The field [Tgl lahir] is selected. On the right, the 'Sumber Data' dropdown is set to 'Test Data Type'. The 'Isi Data' dropdown is set to 'Tanggalku'. Below these, the 'Tipe' is 'Waktu', 'Primary Key' is 'False', and 'Foreign Key' is 'False'. At the bottom are 'OK' and 'Hapus' buttons.

Gambar 4.16. *Form* properti tabel dengan *field* bertipe data tanggal.

The screenshot shows a window with a toolbar containing icons for 'Pohon Tabel', 'Level Field', 'To Generate', 'Karakter', 'Numerik', 'Tanggal / Jam', and 'Multi Field'. The 'Multi Field' icon is active. The main area displays 'MFku' and 'Suara'. On the right, there are 'Baru' and 'Hapus' buttons. A large watermark for 'stikom SURABAYA' is visible in the background.

Gambar 4.17. *Form* lembar kerja tipe data multi *field*.

Sama seperti *form* lembar kerja sebelumnya, pada *form* lembar kerja tipe data multi *field* (gambar 4.17) di atas berfungsi untuk menampilkan definisi nama aturan tipe data multi *field* yang dibuat. Jika tombol baru di tekan maka akan muncul *form* tipe data multi *field* seperti gambar 4.18 berikut.

Gambar 4.18. *Form* tipe data multi field.

*Form* tipe data *multifield* (gambar 4.18) berfungsi untuk membuat aturan baru untuk tipe data *multifield*. Definisi merupakan nama aturan tipe data *multifield* yang dibuat, isi string merupakan isi dari tipe data karakter dan isi numerik merupakan isi dari tipe data numerik yang diinginkan.

Gambar 4.19. *Form* properti tabel dengan *field* bertipe data multi field.

Seperti pada *form* propertis tabel dengan *field* bertipe data multi *field*, *form* propertis tabel tipe data multi *field* (gambar 4.19) berfungsi untuk mengisi sumber data dan isi data *field-field* yang ada pada tabel yang akan di-*generate* pada lembar kerja *to generate*. Pilih dulu salah satu atribut tabel yang ada, kemudian pilih sumber data yang diinginkan (konstanta atau *test data type*). Jika sumber data adalah *test data type* maka pilih isi data yang ada, dan jika sumber data konstanta maka diisi sesuai keinginan.

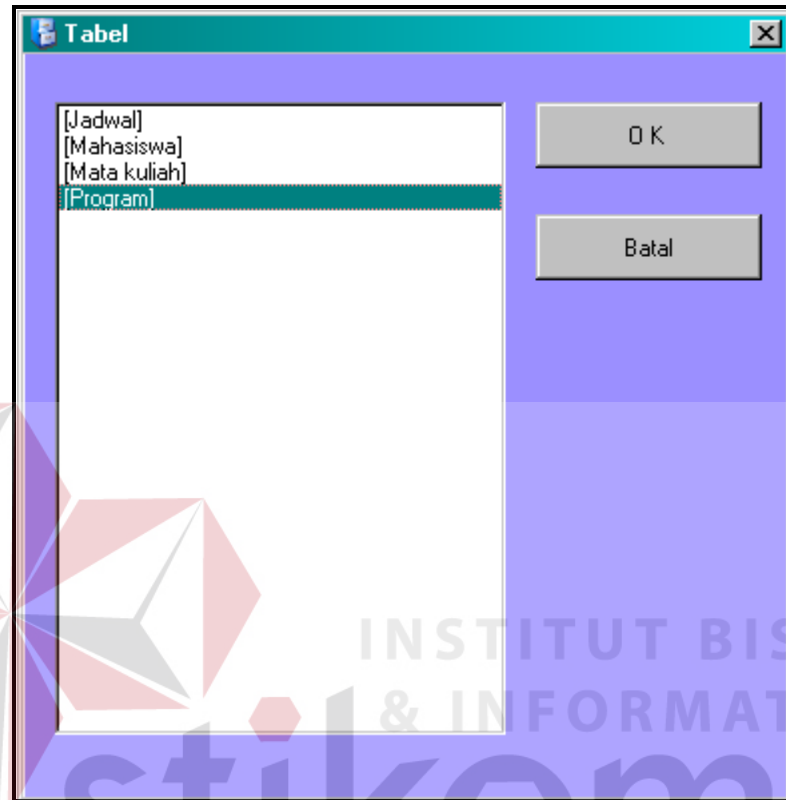
Sama seperti *form* lembar kerja sebelumnya, pada *form* lembar kerja *to generate* (gambar 4.20) berfungsi untuk menampilkan nama tabel yang akan di-*generate*, keterangan ada join dengan tabel lain atau tidak, dan keterangan jumlah *field* yang ada pada tabel tersebut.

Pohon Tabel	Level Field	To Generate	Karakter	Numerik	Tanggal / Jam	Multi Field
Nama Tabel		Ada Join	Jumlah Field			
[Program]		Tidak	3			
						Baru

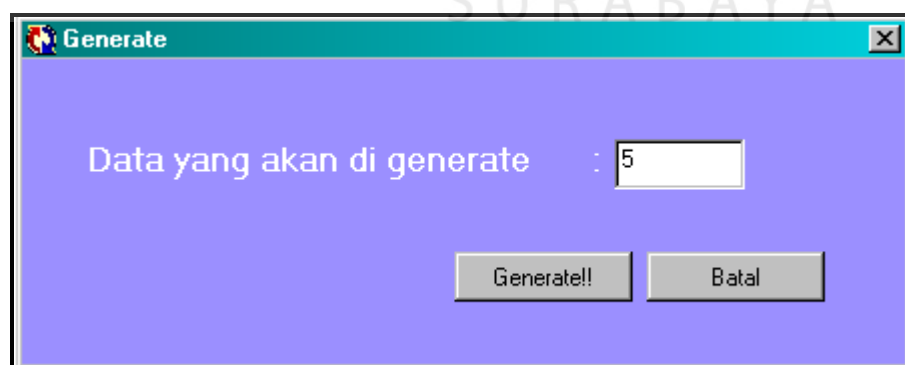
Gambar 4.20. *Form* lembar kerja *to generate*.

Pada *form* lembar kerja *To generate* terdapat tombol baru, jika tombol baru ini di klik maka akan muncul *form* tabel seperti gambar 4.21. *Form* tabel (gambar 4.21) berfungsi untuk memilih salah satu tabel yang akan di-*generate*.

Setelah itu akan muncul properti tabel yang dipilih seperti yang telah dijelaskan di atas.



Gambar 4.21. *Form tabel.*



Gambar 4.22. *Form generate.*



*Form generate* (gambar 4.22) ini berfungsi men-*generate* data yang telah dipilih pada *form* lembar kerja *to generate* dengan cara memasukkan jumlah data yang ingin di-*generate*.

#### 4.2. Peralatan yang Digunakan

Adapun *hardware* yang digunakan dalam pembuatan sistem ini adalah:

1. Intel Pentium III 668 MHz
2. Hardisk 10 Gb
3. Monitor VGA 15"
4. CDRW / DVD Samsung 24x16 Max
5. Keyboard
6. Printer Canon Bjc-1000 SP
7. Mouse

Sedangkan *software* yang digunakan adalah:

1. Microsoft SQL Server 7.0
2. Microsoft Access 2003
3. Microsoft Visual Basic 6.0
4. Visio Technical 4.1
5. Power Designer 6 32-bit
6. Allfusion Erwin data modeler 4.1

### 4.3. Evaluasi Output

Untuk mengukur validasi sistem *test data generator* diperlukan tiga evaluasi, yaitu : 1) Desain uji coba dan subyek coba 2) Jenis data dan instrumen pengumpul data 3) Analisis hasil uji coba.

#### 4.3.1. Desain Uji Coba dan Subyek Coba

Dalam desain uji coba dan subyek coba, ada tiga *test case* (uji coba) yang dilakukan sistem, yaitu: (1) Koneksi *database*, (2) Pembacaan pohon tabel, dan (3) Generasi data. Ketiga *test case* tersebut telah dibahas pada bab 3 (sub bab evaluasi). Berikut ini merupakan hasil *test case* pada sistem.

##### A. Koneksi *database*

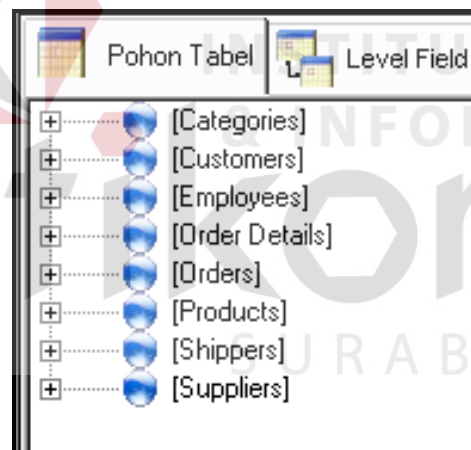
Pada koneksi *database*, jika sistem dapat terhubung ke *database* maka akan terbentuk pohon tabel pada *form* pohon tabel. *Test case* koneksi *database* ada dua yaitu : *test case* koneksi *database* Access dan *test case* koneksi *database* SQL Server. Jika koneksi *database* berhasil, maka akan ditandai dengan munculnya pohon tabel yang terbentuk dari tabel-tabel yang ada pada *database*, dan pohon tabel masih dalam keadaan tertutup (kolom-kolom yang ada dalam tabel tersebut masih belum tampak).

1. Koneksi *database* dengan menggunakan Access 2000



Gambar 4.23. *Test case* koneksi *database* Bibilio pada Access 2000

2. Koneksi *database* dengan menggunakan SQL Server 7.0

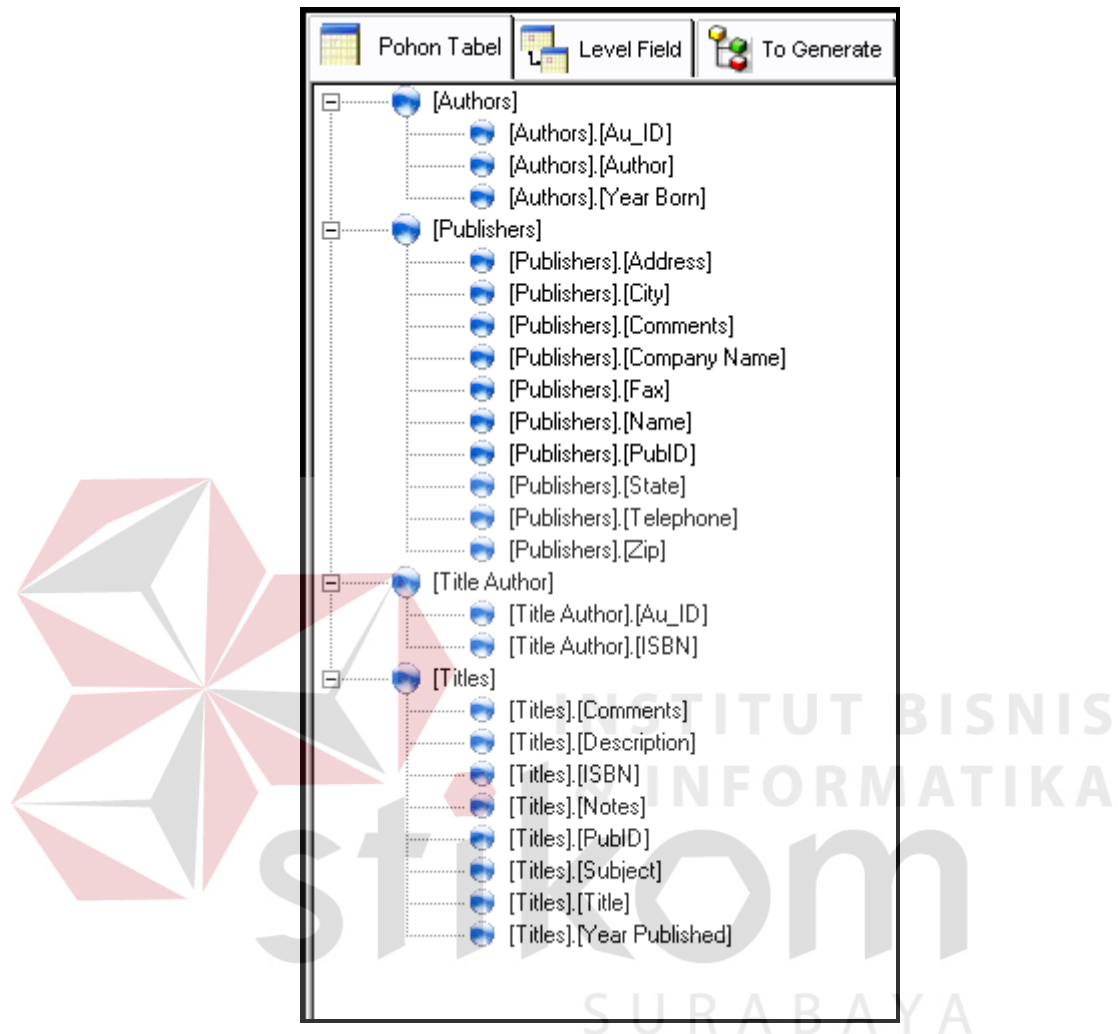


Gambar 4.24. *Test case* koneksi *database* dengan menggunakan SQL Server 7.0

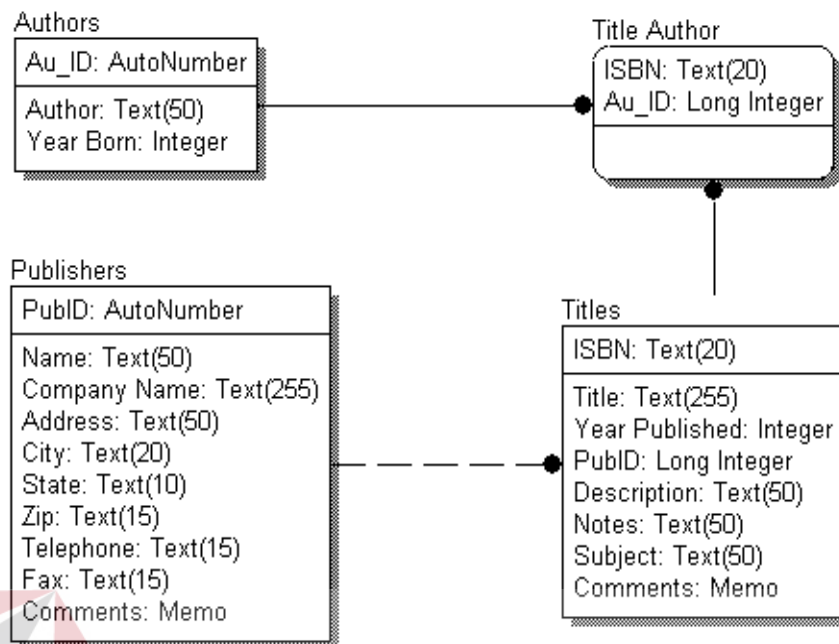
## B. Pembacaan pohon tabel

Pembacaan pohon tabel *database* ini dilakukan dengan menggunakan *Allfusion Erwin data modeler* 4.1 dan aplikasi program. *Database* yang digunakan adalah *database* Bibilio pada Access dan *database* Northwind pada SQL Server.

1. Pembacaan pohon tabel dengan menggunakan *database* Access 2000

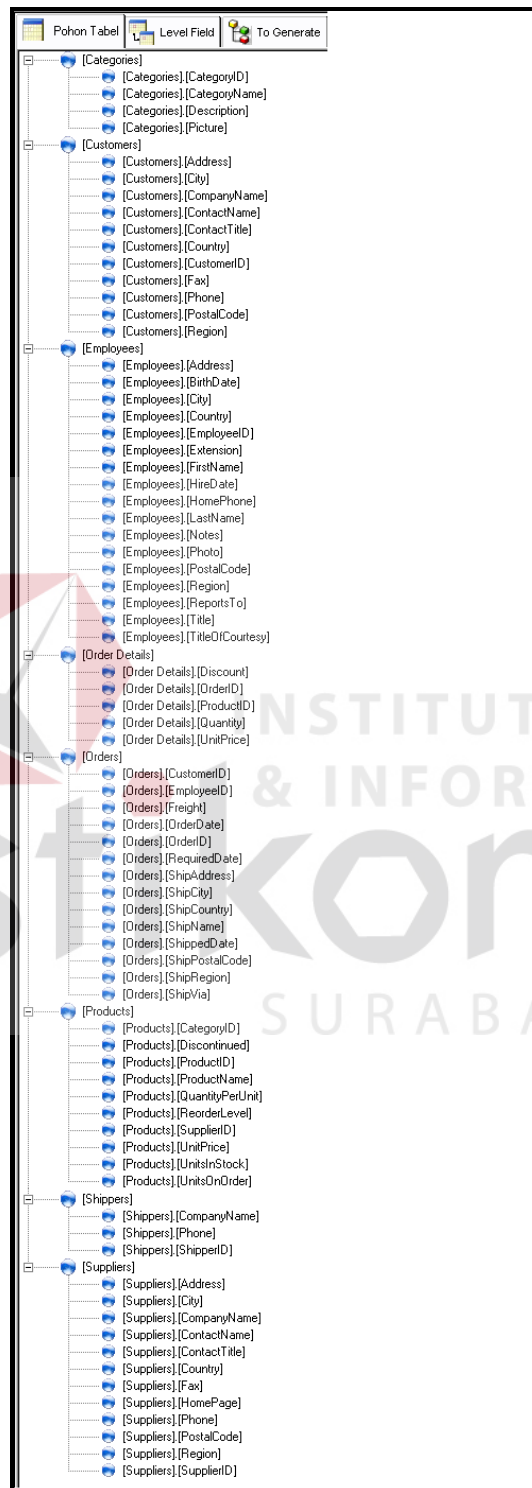


Gambar 4.25. *Test case* pembacaan pohon tabel *database* Bibilio pada Access 2000

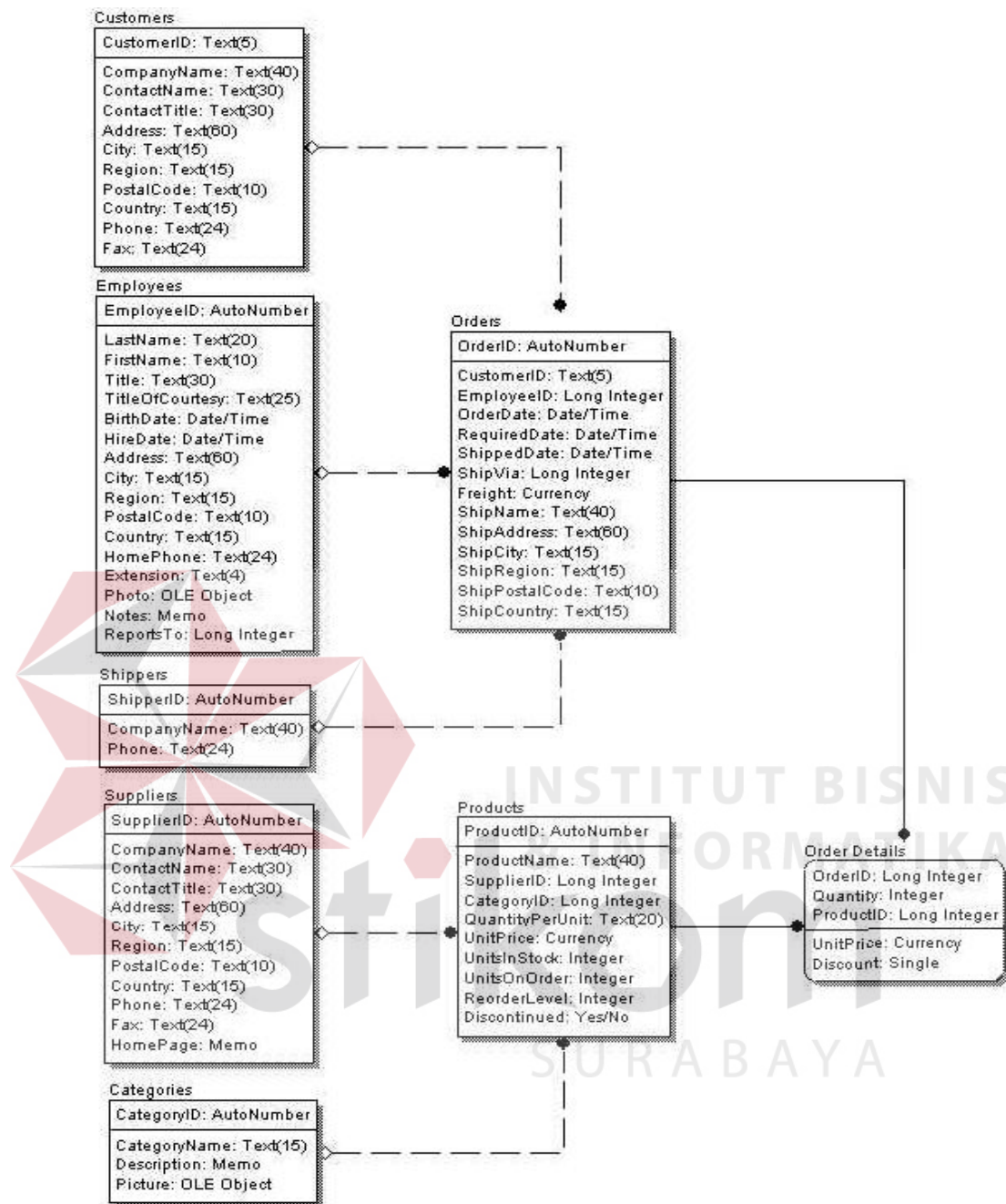


Gambar 4.26. *Schema database Bibilio pada Allfusion Erwin data modeler 4.1*

## 2. Pembacaan pohon tabel dengan menggunakan *database* SQL Server



Gambar 4.27. Test case pembacaan pohon tabel *database* Northwind pada *SQL Server 7*



Gambar 4.28. *Schema database Northwind pada Allfusion Erwin data modeler 4.1*

### C. Generasi data

*Test case* generasi data ini dilakukan dengan menggunakan *database* Bibilio pada Access dan *database* Northwind pada SQL Server. Dimana *output*

dari generasi data dapat dilihat pada gambar berikut.

#### 1. Generasi data menggunakan database Access

	Au ID	Author	Year Born
▶	31460	Sofyan Maulana	21528
	10413	Fauzi Effendy	7215
	15076	Ahmad Fauzan	19143
	30544	Joni	31811
	15631	Sofyan Maulana	4941
	3095	Ahmad Fauzan	10497
	1350	Miftahul Choiriyah	27361

Gambar 4.29. Generasi data menggunakan *database* Bibilio pada Access 2000

#### 2. Generasi data menggunakan database SQL Server

	ShipperID	CompanyName	Phone
▶	1	Speedy Express	(503) 555-9831
	2	United Package	(503) 555-3199
	3	Federal Shipping	(503) 555-9931

Gambar 4.30. Generasi data menggunakan *database* Northwind pada SQL Server 7.0

### 4.3.2. Jenis Data dan Instrumen Pengumpul Data

Jenis data yang diuji coba adalah *database* yang berekstensi MDB pada *Microsoft Access*, dan *database* yang berekstensi LDB pada *SQL Server*. Sedangkan instrumen pengumpul data menggunakan Erwin 4.1.

### 4.3.3. Analisis Hasil Uji Coba

Analisa dilakukan secara kualitatif, dengan melakukan analisa terhadap pemenuhan kebutuhan sistem *test* data generator dari subyek coba yang telah diidentifikasi pada Data Flow Diagram (DFD) dan Entity Relational Diagram (ERD) dalam tahap akusisi dan desain. Keberhasilan uji coba validasi terhadap pemenuhan kebutuhan sistem *test* data generator di setiap subyek coba merupakan



hasil yang diharapkan pada penelitian Tugas Akhir ini.

Tabel 4.1. Rincian Evaluasi Output *test case* pada sistem.

No.	Tujuan	Input	Output yang diharapkan	Output Aplikasi	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.	Pohon tabel Bibilio yang ada pada <i>form</i> lembar kerja pohon tabel sesuai dengan <i>database</i> Bibilio.mdb pada Access (gambar 4.23).	<i>Test case</i> koneksi <i>database</i> Access.
2.	Memastikan system dapat melakukan koneksi ke sistem <i>database</i> SQL Server.	Data koneksi <i>database</i> SQL Server (Northwind).	Pohon table Northwind.ldb yang ada pada <i>form</i> lembar kerja pohon tabel.	Pohon tabel Bibilio yang ada pada <i>form</i> lembar kerja pohon tabel sesuai dengan <i>database</i> Northwind.ldb pada SQL Server (gambar 4.24).	<i>Test case</i> koneksi <i>database</i> SQL Server.

Tabel 4.1. Rincian Evaluasi Output *test case* pada sistem (lanjutan).

No.	Tujuan	Input	Output yang diharapkan	Output Aplikasi	Keterangan
3.	Memastikan bahwa <i>star</i> skema yang dibuat sesuai dengan <i>database</i> target (tujuan).	Tabel yang ada pada <i>database</i> Access (Bibilio).	Pohon tabel Bibilio.mdb yang ada pada <i>form</i> lembar kerja pohon tabel.	Pohon tabel Bibilio yang ada pada <i>form</i> lembar kerja pohon tabel sesuai dengan output Erwin 4.1 (gambar 4.26).	<i>Test case</i> pembacaan pohon tabel <i>database</i> Access.
4.	Memastikan bahwa pohon tabel yang dibuat sesuai dengan <i>database</i> target (tujuan).	<i>Database</i> pohon tabel SQL Server yang digunakan adalah Northwind.	Pohon tabel Northwind.ldb yang ada pada <i>form</i> lembar kerja pohon tabel.	Pohon tabel Bibilio yang ada pada <i>form</i> lembar kerja pohon tabel sesuai dengan output Erwin 4.1 (gambar 4.28).	<i>Test case</i> pembacaan pohon tabel <i>database</i> SQL Server.

Tabel 4.1. Rincian Evaluasi Output *test case* pada sistem (lanjutan).

No.	Tujuan	Input	Output yang diharapkan	Output Aplikasi	Keterangan
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> .	Data generasi yang dihasilkan sesuai dengan output pada <i>database</i> Access (gambar 4.29).	<i>Test case</i> generasi data pada <i>database</i> Access.
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> .	Data generasi yang dihasilkan sesuai dengan output pada <i>database</i> SQL Server (gambar 4.30).	<i>Test case</i> generate data pada <i>database</i> SQL Server.

Berdasarkan hasil uji coba (*test case*) di atas, sistem generasi data ini dapat dikatakan berhasil men-generate data karena data yang dihasilkan sesuai dengan aturan yang telah dibuat sebelumnya.