

Panduan Membuat *Data Flow Diagram, Entity Relationship Diagram*, dan *Database* Menggunakan Power Designer

Disusun oleh: Oke Setiawan, S.T.



*Untuk keperluan internal dalam mendukung Tugas Besar
Sistem Informasi Manajemen – Jurusan Teknik Industri –
Universitas Katolik Parahyangan – Bandung*

Bandung - 2008

Daftar Isi

BAB I	PENDAHULUAN	3
BAB II	POWER DESIGNER.....	5
II.1	Power Designer Process Analyst	6
II.1.1	Pembuatan <i>Context Diagram</i>	10
II.1.2	Pembuatan Dekomposisi Proses (DFD Level Selanjutnya).....	13
II.2	Power Designer Data Architect	20
II.2.1	Pembuatan <i>Conceptual Data Model</i> (CDM).....	21
II.2.2	Pembuatan <i>Physical Data Model</i> (PDM)	27
II.3	<i>Generating</i> Basis Data	30
BAB III	WIN A&D	35
III.1	Memulai Proyek Baru	35
III.2	Pembuatan <i>Context Diagram</i> dan <i>Data Flow Diagram</i> 38	
III.3	Pembuatan <i>Entity Relationship Diagram</i>	40
III.4	Pembuatan Basis Data	40

BAB I PENDAHULUAN

Perancangan Sistem Informasi untuk suatu organisasi, menurut James Martin pada bukunya yang berjudul *Information Engineering*, dilakukan dengan melalui beberapa tahap. Tahap – tahap yang dilakukan antara lain: menentukan visi dan misi perusahaan, pembuatan model perusahaan secara menyeluruh, analisis masalah dan tujuan, analisis kebutuhan, perancangan (desain), konstruksi, dan implementasi.

Tulisan ini menitikberatkan pada pembahasan mengenai perancangan sistem informasi untuk tahap desain saja dengan menggunakan perangkat lunak. Perancangan dengan perangkat lunak ini dapat dilakukan setelah tahap desain melewati langkah pendefinisian proses bisnis, pendefinisian entitas untuk setiap proses bisnis, dan pembuatan matriks BSP (matriks proses – entitas).

Tahap yang dapat dilakukan dengan bantuan perangkat lunak adalah mulai dari tahap pembuatan diagram konteks (*context diagram*), diagram aliran data (*data flow diagram*), diagram relasi entitas (*entity relationship diagram*) baik model konseptual (*conceptual data model*) maupun model fisik (*physical data model*). Bantuan yang diberikan perangkat lunak

ini adalah sampai tahap meng-*generate* basis data yang akan digunakan untuk sistem informasi tersebut.

Banyak perangkat lunak yang dapat digunakan untuk membantu perancangan tersebut. Dalam tulisan ini akan dibahas dua perangkat lunak yang dapat digunakan yaitu Power Designer versi 6 dan WinA&D. Pada dasarnya penggunaan perangkat lunak ini hampir sama untuk masing – masing perangkat lunak, yang berbeda adalah tampilan program dan menu – menu yang ada di dalam program. Penjelasan cara penggunaan untuk masing – masing perangkat lunak akan dijelaskan pada bab berikutnya.

BAB II POWER DESIGNER

Power Designer adalah perangkat lunak buatan Sybase yang dibuat untuk membantu dalam perancangan sistem informasi. Namun untuk keperluan yang paling sering digunakan adalah PDPA (*Power Designer Process Analyst*) dan PDDA (*Power Designer Data Architect*). Perangkat lunak yang digunakan adalah Power Designer versi 6 meskipun sampai tulisan ini ditulis Sybase telah mengeluarkan Power Designer versi 12.5.

Power Designer Process Analyst (PDPA) digunakan untuk membantu dalam proses penggambaran *data flow diagram* mulai dari *context diagram*. Kelebihan dari perangkat lunak ini adalah dapat membantu untuk memeriksa apakah model yang dibuat sudah valid atau belum dan dapat langsung di-generate menjadi bentuk *Entity Relationship Diagram*.

Power Designer Data Architect (PDDA) digunakan untuk membantu dalam penggambaran *entity relationship diagram*. PDDA ini meng-import data dari *data flow diagram* yang telah dibuat dengan PDPA. PDDA ini akan meng-import semua *data-store* yang telah dibuat di *data flow diagram*.

Langkah – langkah penggunaan PDPA dan PDDA akan dijelaskan di masing – masing subbab di bawah ini. Sebelum memulai, pastikan dahulu bahwa kedua program tersebut telah diinstalasi di komputer. Bagi yang belum memiliki, program dapat diunduh di Laboratorium Teknologi Informasi Lt. 2.

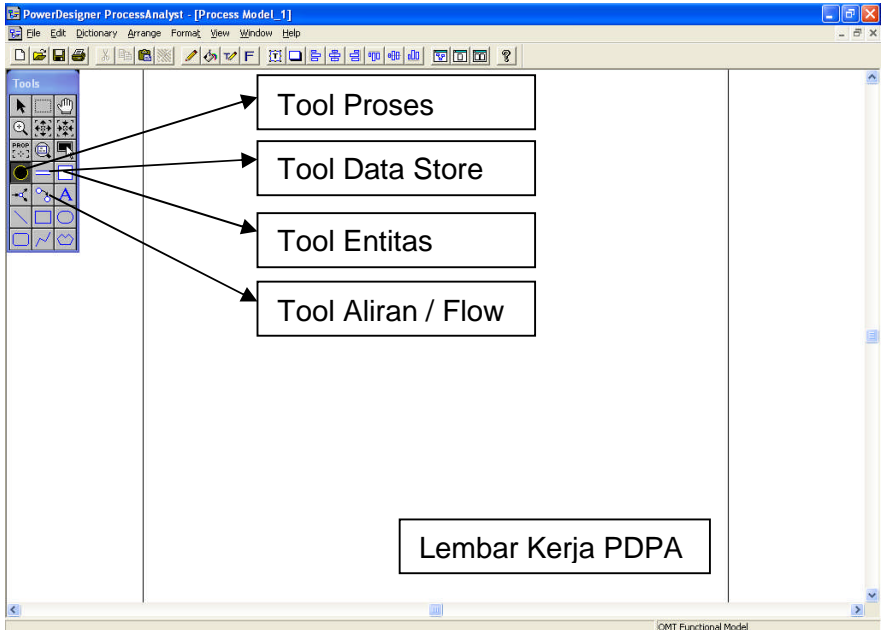
Instalasi dilakukan dengan membuka folder “Disk 1” dan menjalankan program bernama “Setup.exe” kemudian ikuti langkah selanjutnya sampai program selesai instalasi. Cara instalasi ini berlaku baik untuk instalasi PDPA maupun PDDA.

II.1 Power Designer Process Analyst

Sebelum mempelajari cara penggunaan PDPA, ada baiknya jika terlebih dahulu mengenal *interface* dari PDPA ini. *Interface* untuk program PDPA dapat dilihat pada gambar II-1. Penjelasan untuk bagian *interface* ini hanya pada bagian – bagian penting yang biasa digunakan untuk membuat *data flow diagram*.

Ada hal yang perlu diketahui pada Power Designer bahwa setiap objek memiliki *code* dan *name*. *Name* adalah nama atau label yang akan ditampilkan pada objek, sedangkan *object* adalah identitas objek itu sendiri sehingga harus unik (tidak boleh ada yang sama dalam satu proyek). Pada umumnya, *user* memberikan nama pada objek, kemudian *code*

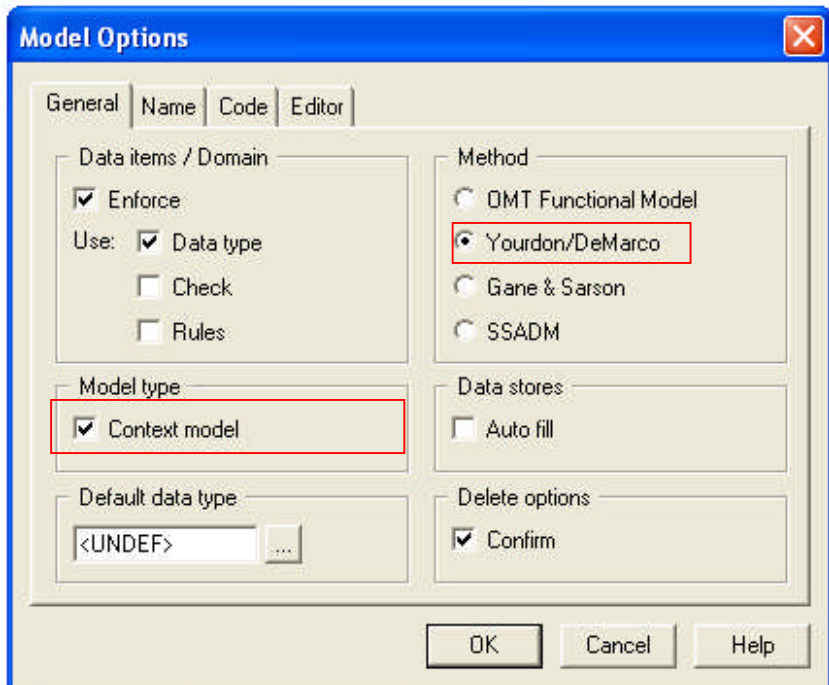
disamakan dengan nama. Caranya menekan tombol (=) pada baris *code*. Contohnya, lihat gambar II-2.



Gambar II-1 *Interface* Power Designer Process Analyst

Sebelum mulai menggambar, hal yang lebih dahulu perlu dilakukan adalah menentukan terlebih dahulu jenis model yang akan dibuat. Oleh karena itu, bukanlah "Model Options" dengan cara membuka menu **File** → **Model Options**. Jendela yang muncul dapat dilihat pada gambar II-2. Yang diubah hanyalah tipe model apakah akan membuat *context diagram* atau tidak; dan metode penggambaran yang digunakan. Yang

umum digunakan adalah Yourdon/DeMarco dan Gane&Sarson. Pelajari kembali perbedaan kedua metode ini.



Gambar II-2 *Model Options*

Setelah menentukan properti dari model tersebut, langkah selanjutnya adalah memberikan status untuk proyek yang akan dibuat, yaitu mengisi nama proyek, nama model, dan pembuat model tersebut. Caranya buka menu **Dictionary** → **Model Properties** dan jendela yang muncul dapat dilihat

pada gambar II-3. Coba perhatikan, *code* selalu tidak ada spasi dan tidak boleh ada spasi.

The image shows a 'Process Model Properties' dialog box with the following fields and values:

Field	Value
Project name:	TI Unpar
Project code:	TI_UNPAR
Name:	Sistem Informasi Perpustakaan
Code:	SISTEM_INFORMASI_PERPUSTAKAAN
Label:	
Author:	Cryo Saber Strike
Version:	
Language:	English (United States)
File:	
Created:	8/21/2008 3:32 PM
Modified:	8/21/2008 3:41 PM

Buttons at the bottom: OK, Cancel, Apply, Help.

Gambar II-3 *Process Model Properties*

Setelah properti model dan opsi model diset sesuai dengan keinginan penggambar, baru penggambaran *context diagram* dan *data flow diagram* dapat dilakukan.

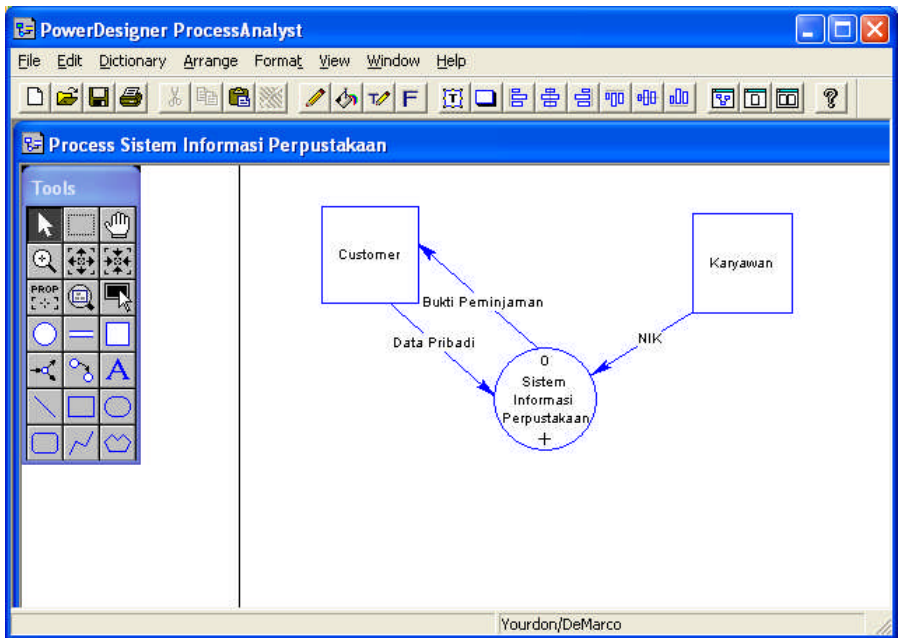
Penggambaran *context diagram* dan dekomposisi proses akan dijelaskan pada subbab tersendiri berikut ini.

II.1.1 Pembuatan *Context Diagram*

Cara menggambar *context diagram* dilakukan dengan meng-klik salah satu tool yang ada kemudian menempatkannya di lembar kerja PDPA. Langkah pertama, tempatkan dahulu satu buah proses yang merupakan *context diagram* dan semua entitas eksternal yang ada.

Langkah kedua adalah mengisikan properti untuk masing – masing entitas eksternal dan proses. Caranya adalah dengan meng-*double-click* objek yang akan diubah. Berikan nama dan *code* yang sesuai dengan keinginan namun merepresentasikan objek tersebut.

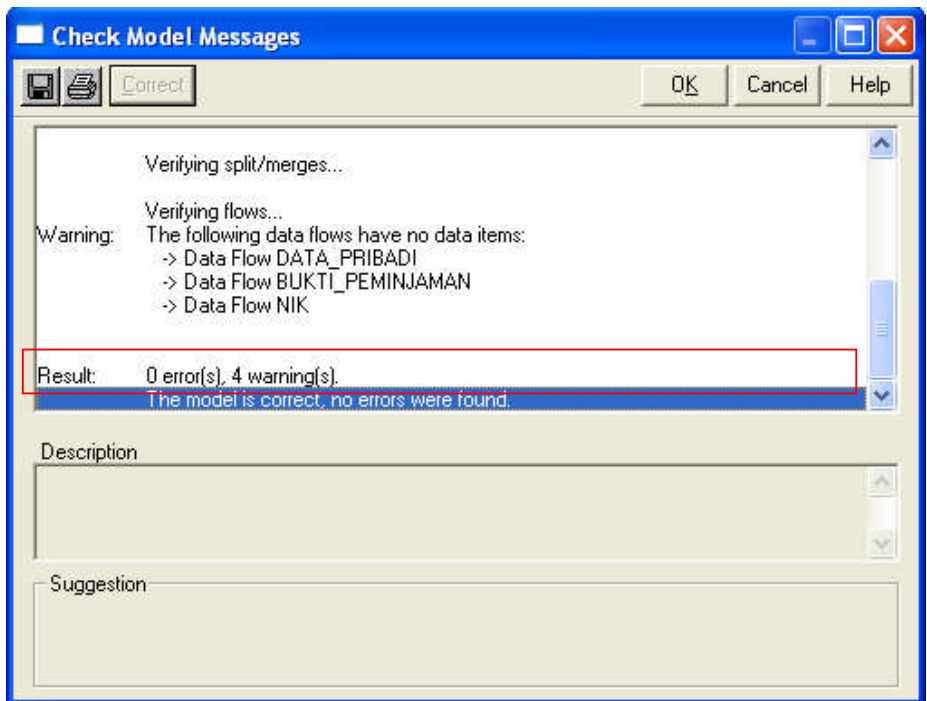
Langkah ketiga adalah menghubungkan entitas eksternal dan proses dengan menggunakan aliran (*flow tool*). **Ingat**, untuk menghubungkannya, tariklah garis dari tengah objek ke tengah objek yang lainnya. Jangan menarik garis dari luar objek karena program tidak akan mengenali sumber atau tujuan aliran data tersebut. Contoh penggambaran *context diagram* dapat dilihat pada gambar II-4.



Gambar II-4 Contoh *Context Diagram*

Langkah keempat adalah mengevaluasi model tersebut apakah penggambaran model tersebut telah sesuai dengan ketentuan yang berlaku dalam penggambaran objek. Caranya adalah dengan membuka menu **Dictionary** → **Check Model** atau menekan tombol **F4**. Akan muncul sebuah jendela yang menunjukkan berapa banyak jumlah *error* yang ditemukan dan berapa banyak *warning* yang diberikan pada bagian RESULT di paling akhir (Gambar II-5). Keterangan *error* dan *warning* dapat

dilihat pada bagian atasnya mengapa *error* dan apa yang harus diperhatikan, namun perhatikan **hanya** bagian *error* saja.



Gambar II-5 *Check Model Messages*

Langkah – langkah yang telah diuraikan adalah langkah – langkah untuk membuat *context diagram*. Pada *context diagram* hanya dapat dibuat satu buah proses saja, sedangkan pada *breakdown* proses dapat dibuat lebih dari satu buah proses.

II.1.2 Pembuatan Dekomposisi Proses (DFD Level Selanjutnya)

Setelah membuat *context diagram*, langkah selanjutnya yang harus dilakukan adalah memecah proses tersebut menjadi proses yang lebih detil. Langkah ini yang disebut dengan dekomposisi proses.

Dekomposisi proses pada Power Designer dilakukan dengan memilih proses yang akan didekomposisi, kemudian tekan klik kanan (*right click*) proses tersebut kemudian pilih ***Decompose***. Setelah pilihan tersebut dipilih, akan muncul jendela model proses yang baru seperti pada *context diagram*.

Pada saat sebuah proses didekomposisi, entitas – entitas dan data yang masuk ke proses dan keluar dari proses akan secara otomatis ditampilkan pada jendela yang baru tersebut. Jika sebuah entitas memberikan atau menerima data lebih dari satu buah, maka pada saat dekomposisi akan muncul entitas tersebut beberapa kali namun berbeda dengan entitas yang hanya ada satu kali. Perbedaannya adalah adanya gambar segitiga kecil di sudut kanan bawah entitas tersebut.

Hal yang perlu diperhatikan adalah jangan menghapus entitas maupun aliran data yang secara otomatis dibuat oleh Power Designer karena semua objek itu saling berhubungan

dari awal sampai akhir. Untuk menjaga konsistensi pembuatan DFD ini, ikutilah beberapa langkah yang dijelaskan berikut ini.

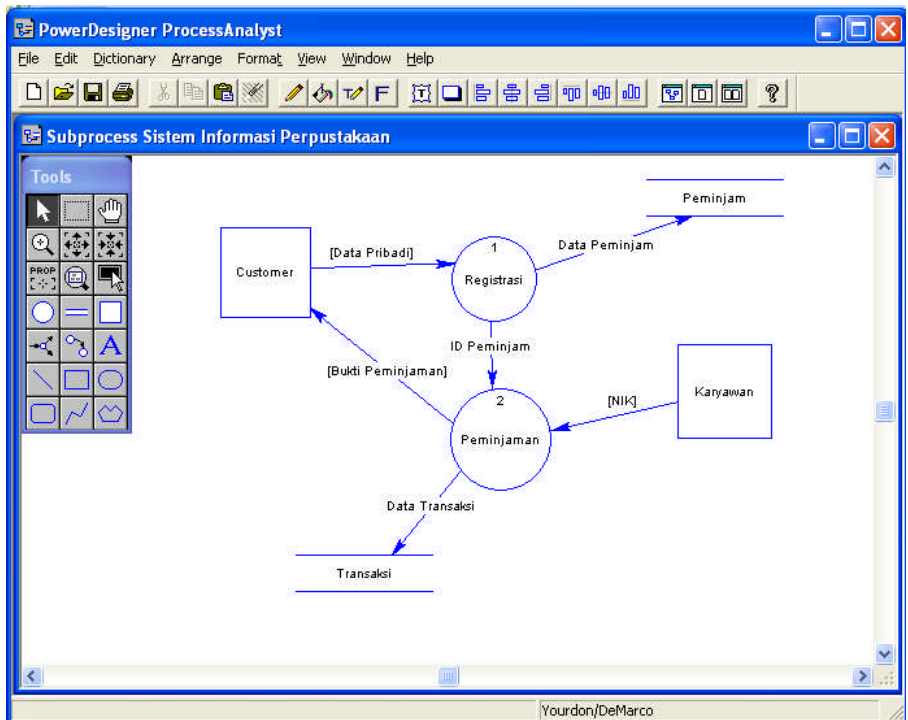
Langkah pertama pembuatan dekomposisi proses ini adalah menambahkan proses – proses yang diperlukan di subproses ini. Langkah ini sama seperti saat membuat proses pada *context diagram*. Setelah ditempatkan, semua properti dari objek diubah sesuai dengan nama proses yang telah direncanakan beserta *code* dari masing – masing proses tersebut.

Langkah kedua adalah menghubungkan entitas dan aliran data yang telah ditampilkan oleh program secara otomatis. Ingat baik – baik bahwa menghubungkan garis tersebut harus ke tengah objek untuk memastikan tidak ada aliran yang terputus. Setelah semua entitas dan aliran hasil dekomposisi dari proses sebelumnya dihubungkan, baru lanjutkan ke langkah berikutnya.

Langkah ketiga adalah membuat aliran antarproses yang belum berhubungan antara proses yang satu dengan yang lain. Kemudian isikan pula nama data yang mengalir dan *code* dari aliran tersebut. Jika ada entitas yang lebih dari satu kali muncul ingin dimunculkan hanya satu saja, atau dengan kata lain, ingin menampilkan entitasnya satu kali dengan aliran data yang banyak, hapuslah salah satu entitas tersebut, **bukan**

menghapus aliran datanya. Untuk lebih jelasnya, lihat gambar II-6 untuk entitas *customer*.

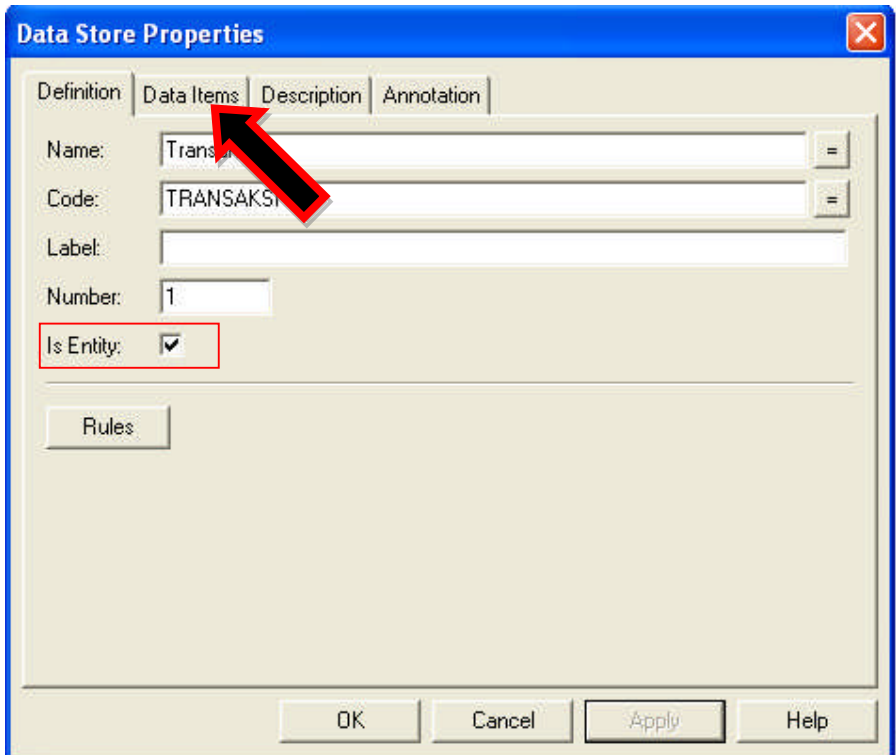
Langkah keempat adalah menambahkan *data store* jika memang diperlukan. Pada saat mengubah properti dari *data store* ini ada sedikit perbedaan dengan objek lain.



Gambar II-6 Dekomposisi Proses

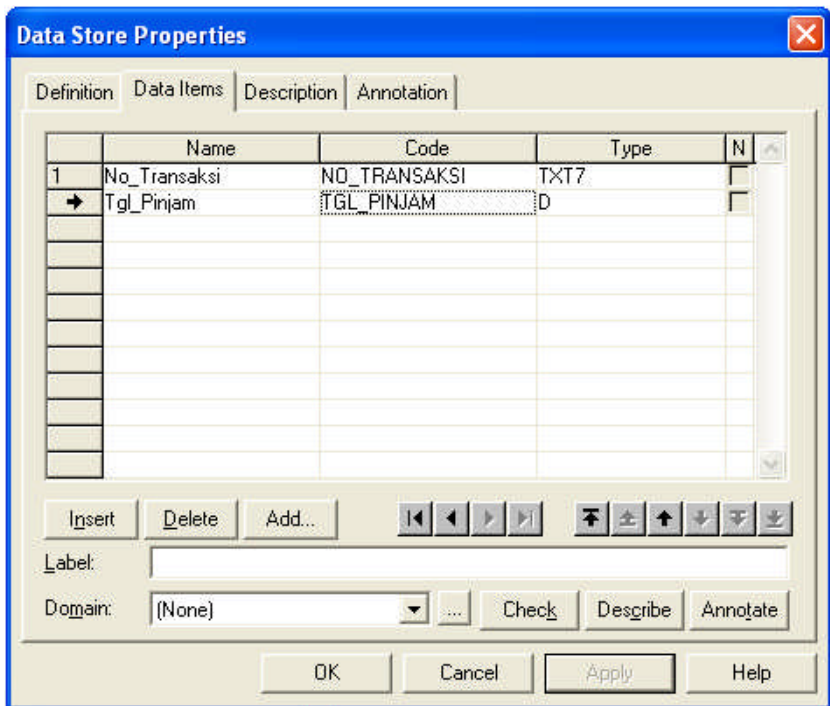
Cara memberikan properti untuk *data store* dilakukan dengan tiga tahap. Tahap pertama seperti cara memberikan

properti pada objek lain, yaitu memberikan nama dan *code* untuk *data store* tersebut. Tahap yang kedua adalah memberikan tanda *check* (✓) pada bagian “**Is Entity**” jika *data store* tersebut merupakan data yang akan disimpan ke dalam basis data (*database*). Lihat contoh pada gambar II-7 untuk penjelasan yang lebih baik.



Gambar II-7 Properti *Data Store*

Tahap ketiga adalah memilih tab “**Data Items**” untuk menentukan *field* apa saja yang diperlukan untuk entitas tersebut. Penentuan *field* apa saja pun sudah harus ditentukan pula tipe dari *field* tersebut, apakah bertipe *text* atau *string* dengan lebar tertentu, bertipe *integer*, *date*, *time*, dsb.

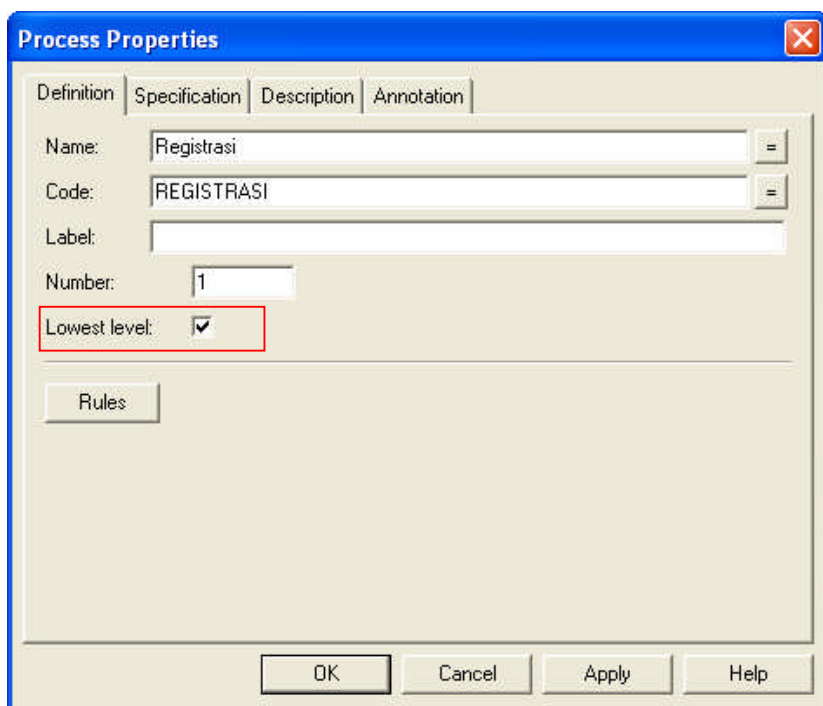


Gambar II-8 Properti *Data Store* – *Data Items*

Contoh yang dibuat pada gambar II-8 adalah untuk *data store* Transaksi dengan keperluan *field* No_Transaksi dengan

tipe `String[7]` dan *field* `Tgl_Pinjam` bertipe *Date*. Untuk keperluan pembuatan ERD selanjutnya, dibuat pula contoh untuk *data store* `Peminjam` dengan keperluan *field* `ID_Konsumen` bertipe `String[7]`, `Nama` bertipe `String[30]`, `Alamat` bertipe `String[50]`, `No_HP` bertipe `String[13]`, dan `No_Telp` bertipe `String[13]`.

Setelah selesai proses untuk dekomposisi satu proses, lakukan dekomposisi untuk proses yang lainnya. Proses yang terdapat pada dekomposisi proses pun dapat didekomposisi lagi menjadi lebih detil. Jangan lupa, apabila sebuah proses sudah tidak memiliki dekomposisi lagi, beri tanda *check* (✓) pada bagian “**Lowest Level**” seperti ditunjukkan pada gambar II-9.



Gambar II-9 Properti Proses

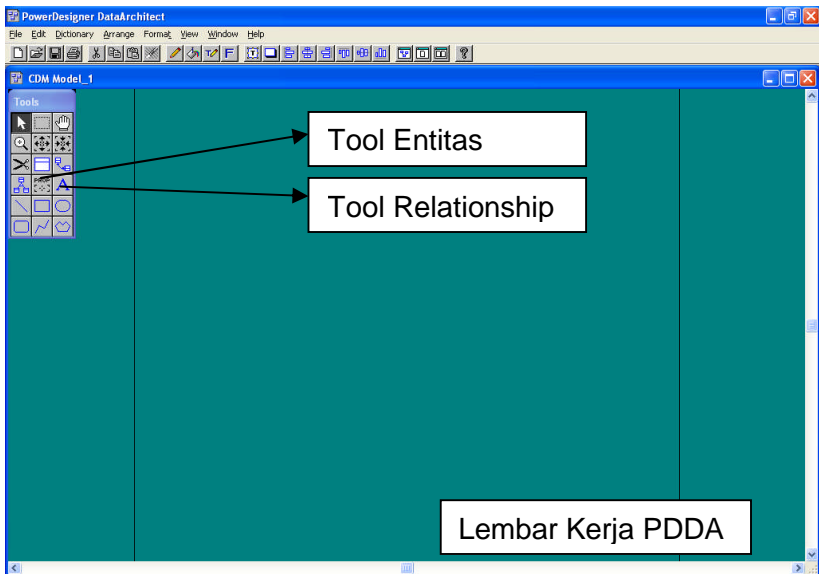
Setelah selesai pada sebuah subproses, lakukan hal yang sama untuk membuat subproses atau dekomposisi proses yang lainnya. Patuhi tahapan – tahapan yang telah diberikan untuk menjaga konsistensi pembuatannya, namun tidak menutup kemungkinan untuk mengubah apa yang telah dibuat. Jangan pula lupa lakukan pengecekan terhadap model tersebut dengan menekan tombol **F4** atau memilih menu **Dictionary** → **Check Model**.

Apabila seluruh proses telah dibuat dan didekomposisi, berarti pembuatan DFD telah selesai dilakukan. Langkah selanjutnya yang perlu dilakukan adalah dengan membuat *Entity Relationship Diagram* (ERD). Pastikan pula *file* DFD telah disimpan dengan baik di media penyimpanan tetap (*hard disk*).

II.2 Power Designer Data Architect

Power Designer Data Architect adalah program dari Power Designer yang digunakan untuk membantu dalam menggambar *Entity Relationship Diagram* (ERD). Kelebihan dari PDDA ini adalah dapat meng-*import data items* dari DFD yang dibuat dengan Power Designer Process Analyst.

Pada saat membuka program ini, yang pertama kali ditampilkan adalah jendela untuk membuat CDM (*Conceptual Data Model*) yaitu model konsep dari pembuatan basis data. Tampilan program ini dapat dilihat pada gambar II-10.

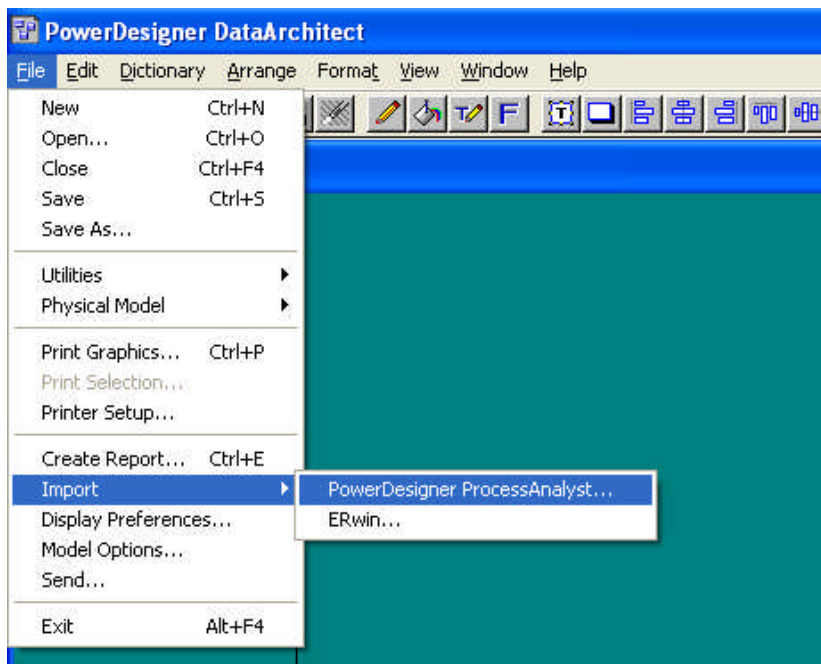


Gambar II-10 Tampilan Program Power Designer Data Architect

II.2.1 Pembuatan *Conceptual Data Model* (CDM)

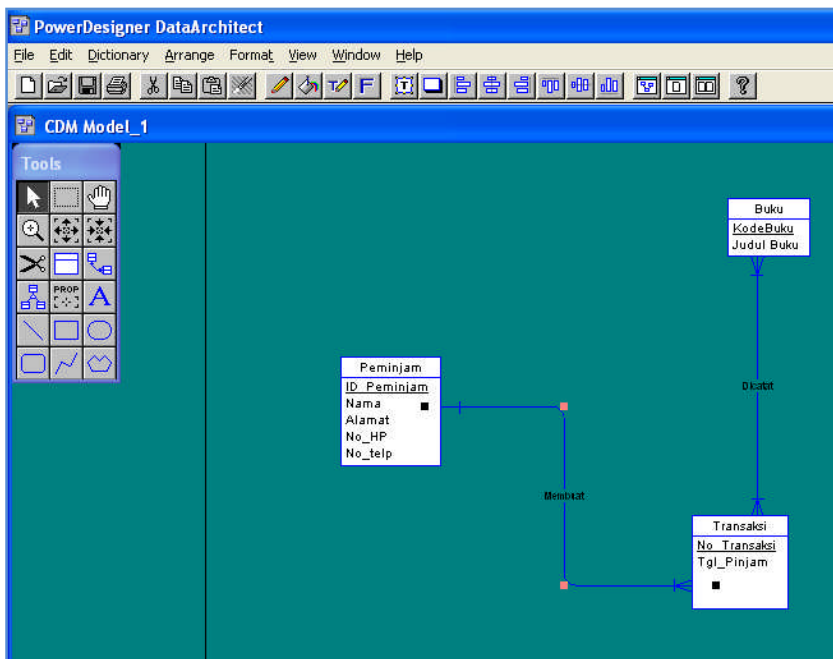
Langkah paling awal untuk membuat ERD dengan perancangan berorientasi proses adalah meng-*import data items* yang telah dibuat pada *data store* yang tersimpan pada *file* yang dibuat dengan Power Designer Process Analyst. Caranya adalah dengan membuka menu **File** → **Import** → **PowerDesigner ProcessAnalyst...** kemudian bukalah *file* yang telah dibuat. Kemudian akan muncul jendela baru yang memberi pilihan apa saja yang akan diambil dari *file* tersebut.

Pilihlah semua dengan memberi tanda *check* (✓) untuk semua bagian.



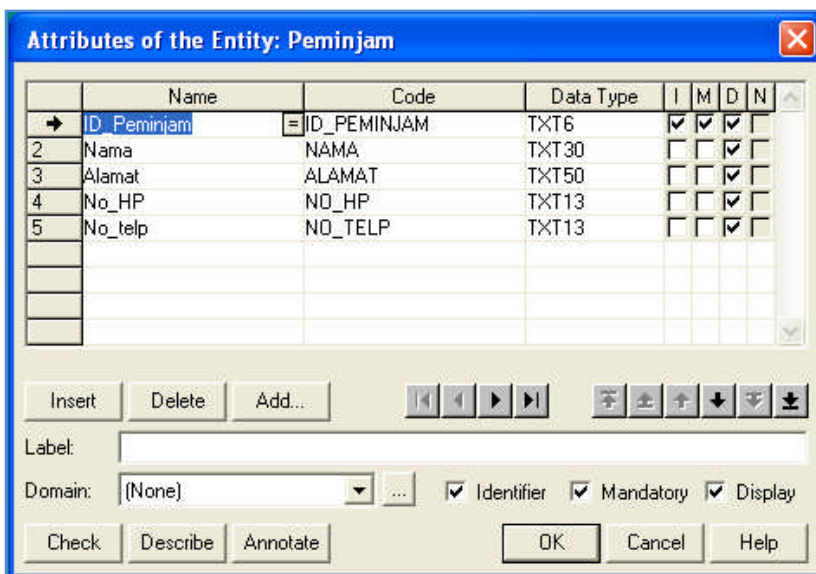
Gambar II-11 Meng-import file PDPA

Setelah berhasil di-import maka akan muncul entitas – entitas yang akan dibuat struktur basis datanya. Entitas yang muncul ini akan memunculkan juga semua atribut yang dimiliki entitas tersebut. Jangan lupa untuk mengubah **Model Properties** seperti saat membuat DFD.



Gambar II-12 Tampilan Setelah Data Di-import

Langkah pertama pembuatan CDM ini adalah membereskan atribut – atribut yang telah ada yaitu menentukan *primary key* (kunci utama) dari setiap *field*. Caranya adalah klik-kanan entitas yang akan diubah kemudian pilih **Attributes**, maka akan muncul jendela untuk mengubah – ubah atribut. Berikan tanda *check* (✓) pada bagian I (*Identifier*) pada atribut yang akan menjadi *primary key*. Lakukan ini untuk semua atribut.



Gambar II-13 Jendela Atribut

Ada hal yang perlu diperhatikan dalam menentukan tipe dari atribut. Pada konsep basis data dikenal beberapa tipe String, yaitu *Character*, *Variable Character*, dan *Text/Memo*. *Text/Memo* adalah variabel karakter yang panjangnya tidak terbatas dan bisa terdapat perpindahan baris (ENTER) di dalamnya. *Character* (Char) dan *Variable Character* (VarChar) perbedaannya adalah penggunaan media penyimpanannya. Tipe Char akan memanfaatkan alokasi penyimpanan selebar deklarasi yang ditentukan sedangkan VarChar memakan

penyimpanan selebar tulisan yang diinput meskipun memiliki lebar maksimum seperti yang telah dideklarasikan.

Perbedaan VarChar dan Char dapat dicontohnya seperti demikian. Char[12] dan VarChar[12] berarti memiliki lebar maksimum 12 karakter. Jadi, pada *field* yang bertipe Char[12] selalu menghabiskan penyimpanan 12 karakter meskipun ada karakter yang bernilai *null*. Misalnya disimpan sebuah kata Bandung pada kedua *field* tersebut. Untuk *field* bertipe Char[12] akan memakan *storage* sebesar 12 karakter meskipun kata Bandung hanya terdiri dari 7 karakter, sedangkan *field* bertipe VarChar[12] hanya memakan *storage* untuk 7 karakter saja.

Pertanyaan selanjutnya yang muncul adalah kapan Char dan VarChar digunakan? Biasanya VarChar digunakan untuk *field* yang panjang isinya tidak tentu misalnya nama orang, alamat, dan sejenisnya. Sedangkan Char digunakan untuk *field* yang panjang isinya selalu tetap dan tidak boleh kurang atau lebih, misalnya kode produk, kode konsumen, dan sejenisnya.

Langkah kedua adalah menentukan relasi antara satu entitas dengan entitas lain. Langkah ini dilakukan dengan menggambar *relationship* yang dihubungkan antarentitas. Jangan lupa pula, seperti pada halnya menggambar aliran proses, menarik garis harus dimulai dari tengah objek ke tengah objek. Kemudian tentukan properti dari relasi tersebut

dengan mengisi nama relasi, *code* relasi, kardinalitas (*one to many*, *many to one*, *many to many*, dsb.), dan *mandatory* atau tidaknya relasi tersebut. Perhatikan gambar II-14 untuk keterangan yang lebih jelas.

Relationship Properties

Definition | Description | Annotation

Peminjam — Transaksi

Peminjam Transaksi

Name: Membuat = Code: MEMBUAT =

Label:

Cardinality

☐ One to One ☒ One to Many ☐ Many to One ☐ Many to Many

"Peminjam" to "Transaksi"

☒ Mandatory ☐ Dependent ☐ Dominant Min: 1 Max: n

Role:

"Transaksi" to "Peminjam"

☒ Mandatory ☐ Dependent ☐ Dominant Min: 1 Max: 1

Role:

Rules ☒ Generate

OK Cancel Apply Help

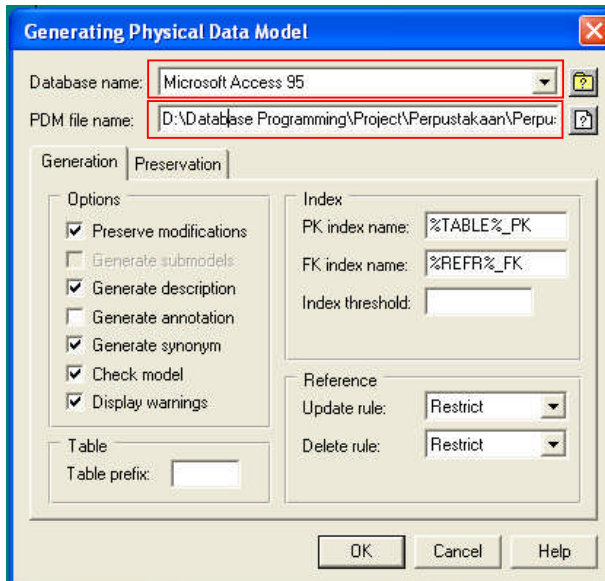
Gambar II-14 Properti *Relationship*

Langkah ketiga adalah mengecek model tersebut apakah sudah valid atau tidak secara teknik penggambaran. Caranya sama seperti pada saat mengecek model pada DFD yaitu membuka menu **Dictionary** → **Check Model** atau menekan tombol **F4**. Kemudian lihatlah *report* dari hasil pengecekan model tersebut.

II.2.2 Pembuatan *Physical Data Model* (PDM)

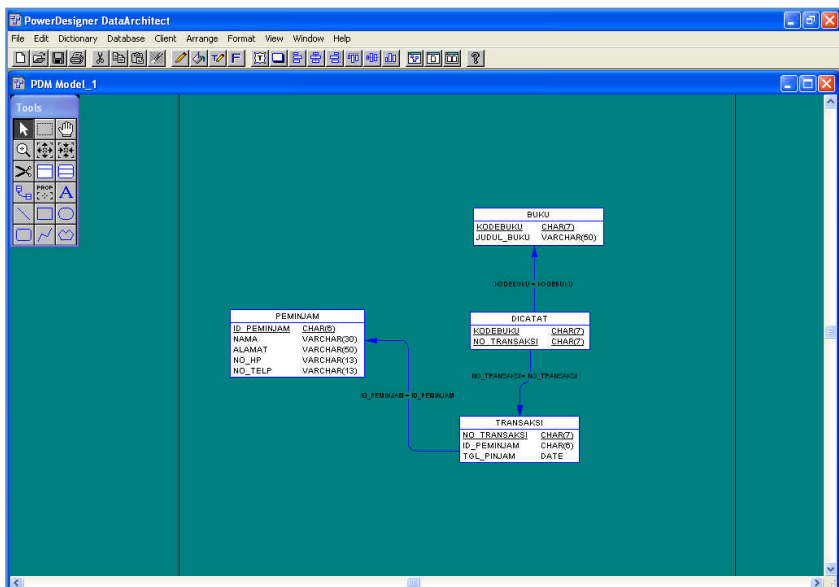
Apabila semua entitas telah dihubungkan dan telah diuji dengan pengecekan model tersebut, langkah selanjutnya adalah membuat *Physical Data Model*. Cara membuat *physical data model* (PDM) adalah dengan membuka menu **Dictionary** → **Generate Physical Model**.

Physical Model merupakan bentuk akhir ERD yang merupakan penggambaran akan basis data yang akan digunakan beserta hubungan antara tabel yang satu dengan yang lain. Oleh karena itu, entitas – entitas yang ada pada PDM inilah yang pada akhirnya menjadi dasar pembuatan struktur basis data.



Gambar II-15 Jendela *Generating Physical Data Model*

Hal yang penting pada saat membuat PDM adalah menentukan basis data apa yang akan digunakan untuk menyimpan data – data organisasi tersebut, apakah menggunakan SQL, Microsoft Access, Clipper, dBase, dll. Selain itu juga tentukan nama *file* yang akan digunakan untuk menyimpan model fisik tersebut. Setelah selesai *click* tombol OK dan PDM akan langsung ditampilkan.



Gambar II-16 *Physical Data Model*

Setelah PDM ditampilkan, jangan lupa untuk mengubah nama model dan kode model pada **Model Properties** supaya apa yang digambarkan menjadi jelas. Pada PDM juga, entitas yang berelasi *many-to-many* akan dibentuk sebuah entitas turunan hasil relasi tersebut. Tambahkan lah atribut – atribut yang diperlukan pada entitas turunan tersebut jika diperlukan tanpa mengubah atribut aslinya.

Setelah selesai mengubah atau menambah atribut – atribut tersebut, pembuatan ERD telah selesai sampai tahap akhirnya. Tahapan perancangan selanjutnya adalah dengan

membuat basis data (*database*) fisik yang akan diimplementasikan pada organisasi.

II.3 **Generating Basis Data**

Secara umum arsitektur basis data dapat dibagi ke dalam dua bagian besar, yaitu arsitektur basis data lokal dan arsitektur basis data *client-server*. Perbedaannya adalah pada letak penyimpanan datanya, apakah setiap komputer menyimpan data masing – masing atau setiap komputer menyimpan data ke sebuah komputer pusat.

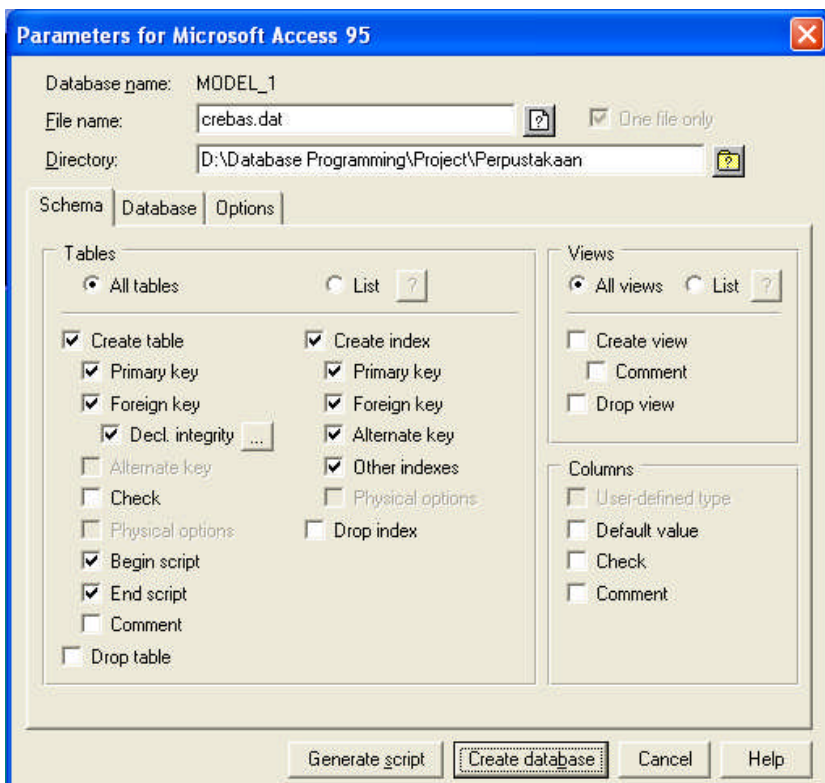
Basis data yang dapat dibuat secara otomatis menggunakan Power Designer Data Architect tidak menentukan apakah arsitekturnya akan lokal atau *client-server*. PDDA hanya membuat struktur basis data berupa tabel – tabel yang telah berisi *field – field* tempat penyimpanan data beserta tipe – tipe datanya.

Sebagai pengetahuan dasar saja pada tulisan ini, pembuatan basis data hanya dijelaskan dua macam saja. Pertama, penyimpanan basis data ke dalam sebuah *file* berformat **.mdb** yang merupakan asosiasi dari Microsoft Access secara langsung. Kedua, penyimpanan basis data menggunakan koneksi ODBC yang biasa digunakan dengan MySQL atau SQL Server.

Pada tulisan ini yang akan dibahas adalah penyimpanan cara yang pertama yaitu pembuatan struktur basis data yang disimpan pada sebuah *file* Microsoft Access. Nantinya *file* ini yang digunakan untuk menyimpan data – data yang diproduksi oleh semua bagian perusahaan.

Langkah pertama adalah buka menu **Database** → **Change Target Database**. Kemudian pilihlah *database* yang akan digunakan. Pada saat Power Designer versi ini dibuat, versi Microsoft Access yang ada baru versi Microsoft Access 95. Oleh karena itu pilih Microsoft Access 95 dan tekan tombol OK.

Langkah kedua adalah membuka menu **Database** → **Generate Database** dan akan muncul jendela baru seperti pada gambar II-17. Tentukanlah *directory* dan *filename* sesuai dengan keinginan kemudian klik tombol **Create Database**.

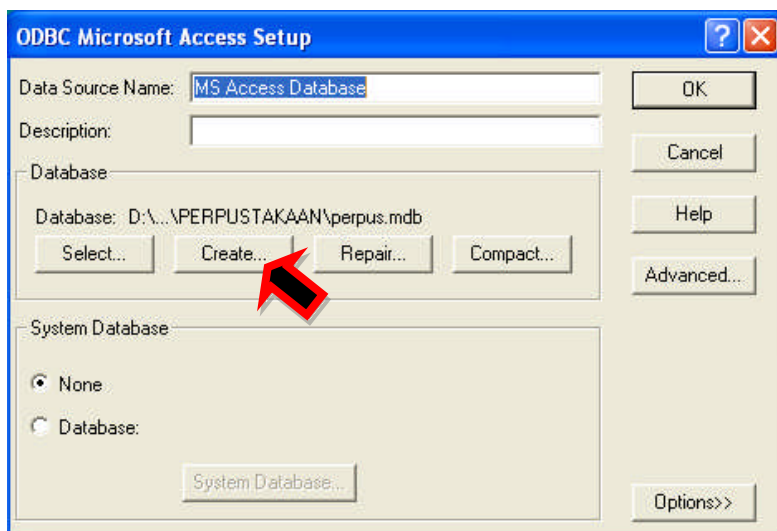


Gambar II-17 *Parameters for Microsoft Access 95*

Setelah tombol ditekan akan muncul jendela baru yaitu **“Connect to an ODBC Data Source”** seperti yang dapat dilihat pada gambar II-18. Kemudian pilih *Datasource Name* **“MS Access Database”** dan tekan tombol **Setup** untuk menentukan *file* mana yang akan digunakan.



Gambar II-18 Pemilihan *Data Source*



Gambar II-19 *Setup file* Microsoft Access

Setelah keluar jendela seperti pada gambar II-19, kebutuhan sekarang adalah membuat *file* baru tempat menyimpan data – data tersebut. Pertama, pilih tombol **Create** kemudian tentukan nama *file* yang akan digunakan dan *folder* tempat menyimpan *file* tersebut. Apabila telah berhasil, nama *file* tersebut beserta *folder* akan tertulis pada bagian *database* seperti pada gambar II-18, kemudian tekan tombol OK.

Setelah kembali pada jendela seperti pada gambar II-17, tekan tombol **Connect**. Setelah ditekan, basis data telah dibuat menjadi sebuah *file* Microsoft Access dan dapat dibuka dengan menggunakan Microsoft Access 95 ke atas. Jangan lupa untuk melakukan **Disconnect** pada database yang sudah dibuat dengan membuka menu **Database** → **D**isconnect. Setelah selesai, simpan semua *file* baik CDM maupun PDM dan berakhirlah pembuatan basis data dengan bantuan komputer.