

# PERANGKAT LUNAK untuk MENTRANSFORMASIKAN MODEL ENTITY RELATIONSHIP ke MODEL RELATIONAL

**Jajang Kusnendar**

Program Studi Pendidikan Ilmu Komputer  
Universitas Pendidikan Indonesia  
Email: jkusnendar@upi.edu

## Abstrak

Model Data merupakan kumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantik (makna) data dan batasan data. Suatu Model yang dinyatakan dalam bentuk diagram awal (preliminary design) akan lebih mudah untuk dievaluasi/dianalisis untuk kemudian dilakukan perbaikan-perbaikan untuk mendapatkan sebuah model data yang lebih permanen dan lebih mendekati kenyataan yang sesungguhnya.

Model Entity Relationship dan Model Basis Data Relasional merupakan suatu cara untuk merepresentasikan model data dalam perancangan basis data. Pada Model E-R perancangan didasarkan pada objek sedangkan pada Model Basis Data Relasional didasarkan pada Record.

Untuk itulah Perangkat Lunak Untuk Mengubah Model Entity Relationship Menjadi Model Basis Data Relasional dibuat, dengan harapan perangkat lunak ini bisa membantu dalam proses perancangan basis data.

**Kata kunci: Model Entity Relationship, Model Basis Data Relasional**

## Pendahuluan

Basis data merupakan representasi digital dari kenyataan fisik dan logik dari sebuah sistem. Mentransformasikan kenyataan dari sebuah sistem ke basis data bukanlah sebuah pekerjaan sederhana, untuk itu kita membutuhkan sebuah media antara, yaitu model data yang merupakan wujud dari perancangan basis data.

Model Data merupakan sekumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantik (makna) data serta batasan data. Ada sejumlah cara dalam merepresentasikan Model Data dalam perancangan basis data, diantaranya adalah Model Entity Relationship (Model E-R) dan Model Basis Data Relasional.

Pada Model Entity Relationship, semesta data yang ada diterjemahkan / ditransformasikan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data. Sedangkan pada Basis Data Relasional kita bisa melihat bagaimana cara / mekanisme untuk mengelola / mengorganisasi data secara fisik dalam memori sekunder, tentu saja akan berdampak pula pada bagaimana kita mengelompokkan dan membentuk keseluruhan data yang terkait dalam sistem yang sedang kita tinjau.

Perangkat Lunak Untuk Mengubah Model E-R menjadi Model Basis Data Relasional adalah suatu perangkat lunak yang diharapkan bisa membantu para perancang basis data yang ingin mengetahui langsung bagaimana bentuk Model Basis Data Relasional dari Model E-R yang dirancangnya.

## Permasalahan

- Apa saja tipe-tipe relasi yang ada didalam model E-R
- Bagaimana cara mentransformasikan model E-R ke model relational berdasarkan tipe relasi
- Bagaiman membuat perangkat lunak yang mampu mentransformasikan model E-R menjadi model relational

Permasalahan akan dibatasi pembahasannya pada hal-hal berikut :

- Model E-R yang dapat diubah ke Model Basis Data Relasional dengan perangkat lunak ini adalah Model E-R yang memiliki variasi relasi sebagai berikut: Relasi Biner, Relasi Tunggal (*Unary Relation*), Relasi Multi Entitas (*N-ary Relation*), Relasi Ganda (*Redundant Relation*), Spesialisasi dengan dua sub entitas dan Agregasi
- Gambar variasi Model E-R sudah dibuat Statis.
- Primary Key pada setiap entitas hanya satu buah.

## Metodologi Penelitian

Metodologi penelitian ini meliputi studi pustaka dan metodologi rekayasa perangkat lunak. Untuk studi pustaka diambil dari buku-buku dan referensi lain yang berhubungan dengan pokok bahasan. Adapun metodologi rekayasa perangkat lunak yang digunakan adalah *Classic life cycle*.

*Classic life cycle* adalah suatu paradigma perangkat lunak yang menuntut suatu sistem yang sistematis, mulai dari suatu level sistem kemudian terus maju ke level berikutnya. Terlepas dari segala kekurangannya model ini masih banyak yang menggunakannya dan dianggap tetap sesuai. Adapun tahapan *Classic life cycle* adalah sebagai berikut:

### 1. Analysis (Analisis)

Analisis dilakukan terhadap data yang ada serta mengumpulkan kebutuhan-kebutuhan

perangkat lunak yang akan dibangun. Merupakan tahap dimana inisialisasi pendefinisian masalah untuk penyelesaian teknis pengembangan perangkat lunak mulai dilakukan. Terminasi tahap analisis, pada saat telah didapatnya definisi permasalahan yang disetujui oleh pengguna dan pengembang.

### 2. Design (Desain)

Pada tahap desain dilakukan perubahan kebutuhan-kebutuhan menjadi bentuk karakteristik yang dimengerti perangkat lunak sebelum dimulai penulisan program. Adapun proses yang dilakukan pada tahap ini adalah: mendekomposisi modul sistem yang akan dikembangkan, penetapan rancangan masukan dan keluaran yang diperlukan, penetapan struktur data yang dipilih, penetapan prosedur kerja internal dan penetapan formula pengolahan data

### 3. Coding (Penulisan Program)

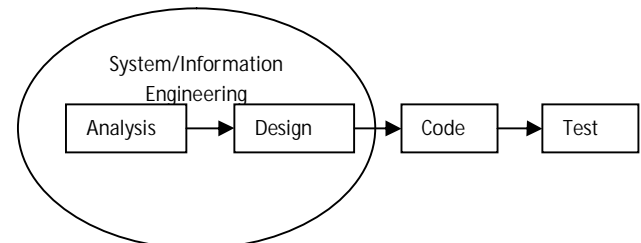
Penulisan program dilakukan setelah tahap desain selesai, yaitu dengan mengubah desain menjadi bentuk program yang dimengerti komputer. Pada tahap ini dilakukan konversi dari hasil rancangan (spesifikasi program) menjadi *source code*.

### 4. Testing (Pengujian)

Setelah program dapat berjalan, selanjutnya dilakukan pengujian dengan memfokuskan pada logika internal dari perangkat lunak, fungsi eksternal dan mencari segala kemungkinan masalah. Selanjutnya memeriksa apakah perangkat lunak sudah sesuai dengan yang diharapkan atau belum. Pengujian merupakan proses untuk eksekusi

program yang telah selesai dibuat untuk memeriksa apakah terdapat kesalahan atau tidak.

Menurut Roger S. Pressman (2001) dalam bukunya yang berjudul *Software Engineering*, gambar dari *waterfall model* dapat dilihat pada gambar berikut:



Gambar 1. Model Sekuensial linear/*Waterfall Model*

## Model Data

Model Data adalah kumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantik (makna) data dan batasan data [FAT99]. Oleh karena itu yang ingin ditekankan adalah makna dari data dan keterhubungannya dengan data yang lainnya.

Ada sejumlah cara dalam merepresentasikan Model Data dalam perancangan basis data, yang secara umum dapat dibagi menjadi 2 kelompok, yaitu :

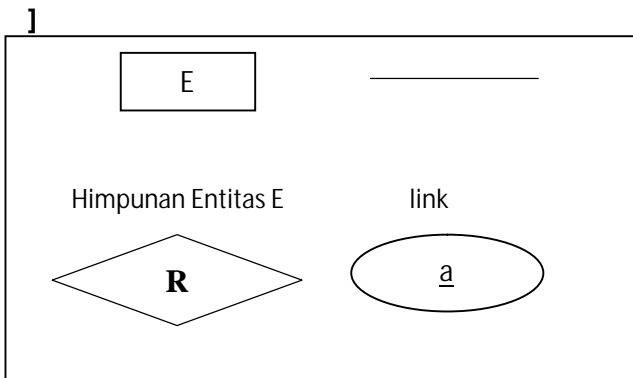
1. Model Logik Data Berdasarkan Objek (Objek-Based Logical Models) yang terdiri dari :
  - a) Model Keterhubungan Entitas (Entity-Relationship Model)
  - b) Model Berorientasi Objek (Object Oriented Model)
  - c) Model Data Semantik (Semantik Data Model)
  - d) Model Data Fungsional (Functional Data Model)
2. Model Logik Data Berdasarkan Record (Record-Based Logical Models) yang terdiri dari :
  - a) Model Relasional (Relational Model)
  - b) Model Hirarki (Hierarchical Model)
  - c) Model Jaringan (Network Model)

## Diagram Entity-Relationship (Diagram E-R)

Sebagaimana yang disebutkan diatas bahwa Model E-R yang merepresentasikan seluruh fakta dari "dunia nyata" yang kita tinjau, dapat digambarkan dengan lebih

sistematis dengan menggunakan Diagram Entity-Relationship (Diagram E-R).

Notasi-notasi simbolik didalam Diagram Entity-Relationship dapat dilihat pada Gambar 2 dibawah ini :



**Gambar 2 Notasi Simbolik Diagram E-R**

Type relasi yang dimiliki diagram E-R terdiri dari :

1. Relasi **satu ke satu** (*one-to-one*)
2. Relasi satu ke banyak (*one to many*)
3. Relasi banyak ke banyak (*many to many*)
4. Relasi Tunggal (*unary relation*)
5. Relasi Multi Entitas (*N-ary Relation*)
6. Relasi Ganda (*Redundant Relation*)
7. Agregasi

### Model Relasional

Model Relasional sering pula disebut Model Basis Data Relasional. Model data ini diperkenalkan pertama kali oleh E.F. Codd. Untuk selanjutnya penyusun akan menggunakan istilah Model Basis Data Relasional.

Model Basis Data Relasional memiliki kelebihan dibanding model data yang lainnya. Model Data ini memiliki kemudahan dalam penerapan dan kemampuannya dalam mengakomodasi berbagai kebutuhan pengelolaan basis data yang ada di dunia nyata (*real world*), hal ini merupakan salah satu alasan mengapa model data ini lebih populer diterapkan.

Pada Model Basis Data Relasional, basis data akan 'disebar' (dipilah-pilah) ke dalam berbagai tabel 2 dimensi. Setiap tabel selalu terdiri atas lajur mendatar yang disebut dengan Baris Data (Row/Record) dan lajur vertical yang biasa disebut dengan Kolom (Column/Field). Di setiap pertemuan Basis Data dan Kolom itulah, item-item data (satuan data terkecil) ditempatkan. Dalam kehidupan kita sehari-hari, Tabel merupakan bentuk natural (alamiah) dalam menyatakan fakta/data yang sering kita gunakan. Itulah

sebabnya, model ini lebih mudah kita terapkan ketimbang model basis data yang lain.

### Analisis

#### Pemetaan dari Model E-R ke Model Basis Data Relasional

Untuk mengubah Model E-R ke Model Basis Data Relasional, maka Model E-R tersebut harus sudah lengkap terlebih dahulu. Setelah lengkap baru kita melakukan proses perubahan.

Untuk Tipe Asosiasi satu ke satu dan satu ke banyak, dapat dilakukan dengan cara penambahan/peleburan atribut-atribut kesalah satu Relasi. Sedangkan untuk Tipe Asosiasi banyak ke banyak dilakukakan dengan cara menambahkan Relasi baru sebagai penghubung.

Penggabungan atau peleburan atribut key diantara dua entitas, akan sangat tergantung dari tipe asosiasinya. Berikut adalah aturan untuk menentukan peleburan atribut :

1. Jika Tipe Asosiasi adalah satu ke satu, maka atribut key (Primary Key) dapat digabungkan ke entitas pertama atau entitas kedua.
2. Jika Tipe Asosiasi adalah satu ke banyak maka atribut key (Primary Key) digabungkan ke entitas yang menunjuk ke banyak.
3. Jika Tipe Asosiasi adalah banyak ke banyak, maka atribut key (Primary Key) dari entitas kesatu dan entitas kedua digabungkan ke Relasi baru sebagai penghubung.

Proses Mengubah Model E-R Menjadi Model Basis Data Relasional sesuai dengan variasi relasi yang dipilih

Secara umum setiap Entitas dan Relationship dari Model E-R akan diimplementasikan menjadi nama Relasi pada Model Basis Data Relasional, sedangkan atribut-atribut pada Model E-R akan diimplementasikan menjadi kolom pada Model Basis data Relasional.

Berikut akan dijelaskan lebih rinci proses perubahan Model E-R untuk setiap variasi relasi menjadi Model Basis Data Relasional :

1. Relasi Biner
  - a) Tipe Asosiasi satu ke satu Relationship yang menghubungkan dua buah Entitas dengan Tipe Asosiasi satu ke satu akan diimplementasikan dalam bentuk penambahan/penyertaan atribut-atribut

Entitas ke Relasi yang mewakili salah satu dari kedua Entitas. Contoh jika entitas DOSEN berelasi dengan JURUSAN (1 ke 1)

Maka untuk Tipe asosiasi satu ke satu Akan selalu ada dua pilihan peleburan (dalam bentuk penyertaan atribut kesalah satu Relasi). Untuk menentukan pilihan yang tepat kita perlu melihat Tipe Asosiasi Minimum-nya. Pada contoh di atas misalkan setiap dosen hanya boleh mengepalai satu jurusan, tapi tidak semua dosen dapat mengepalai satu jurusan. Dengan demikian berarti tipe asosiasi minimumnya adalah 0 (nol). Sedang untuk jurusan hanya boleh dikepalai seorang dosen, tetapi tidak boleh ada jurusan yang tidak ada kepalanya. Dengan demikian, tipe asosiasi minimumnya adalah 1 (satu). Jika demikian, maka yang paling tepat dileburkan ke Relasi yang memiliki tipe asosiasi yang minimumnya paling besar yaitu ke Relasi Jurusan. Jika tipe asosiasi minimumnya sama, maka lebih baik dileburkan ke relasi yang jumlah row-nya lebih sedikit atau ukuran Relasinya diperkirakan lebih kecil.

Maka Model Basis Data Relasional yang dihasilkan dari contoh diatas adalah menjadi :

$$\begin{matrix} \text{Dosen} & = & (*) \\ \text{KodeDos, AlamatDos} & & \\ \text{Jurusan} & & = \\ (*)\text{KodeJur, NamaJur, KodeDos} & & \end{matrix}$$

Penentuan Tipe asosiasi Minimum akan sangat tergantung pada fakta di "dunia nyata" yang sedang ditinjau, karena untuk kasus yang sama bisa saja berbeda untuk tipe asosiasi minimumnya. Oleh karena itu maka proses pengubahan ke bentuk Model Basis Data Relasional-nya pada perangkat lunak ini, akan sangat bergantung pada tipe asosiasi minimumnya. Jadi penentuan Model Basis Data Relasional yang tepat tinggal disesuaikan dengan fakta yang sedang ditinjau. Pada perangkat lunak ini tipe asosiasi minimum terbesar diasumsikan dimiliki oleh entitas kedua, untuk kasus yang tipe asosiasinya satu ke satu.

Maka Output yang berupa Model Basis Data Relasional Untuk contoh di atas adalah sebagai berikut :

$$\begin{matrix} \text{Dosen} = (*\text{KodeDos, AlamatDos}) \\ \text{Jurusan} & & = \\ (*)\text{KodeJur, NamaJur, KodeDos} & & \end{matrix}$$

b) Tipe Asosiasi satu ke banyak

Untuk Relationship dengan Tipe Asosiasi satu ke banyak, akan diimplementasikan dalam bentuk pemberian/pencantuman atribut key dari Entitas yang berderajat satu ke Entitas yang berderajat Banyak. Atribut key dari Entitas yang berderajat satu ini, menjadi atribut tambahan bagi Entitas yang berderajat banyak. Contoh jika entitas DOSEN berelasi dengan KULIAH (1 Ke n). Maka Output yang berupa Model Basis Data Relasional-nya adalah sebagai berikut :

$$\begin{matrix} \text{Dosen} & = & \\ (*\text{KodeDos, namaDos}) & & \\ \text{Kuliah} & & = \\ (*\text{KodeKul, sem, namakul, sks, KodeDos}) & & \end{matrix}$$

c) Tipe Asosiasi banyak ke banyak

Pada Tipe Asosiasi banyak ke banyak, akan diwujudkan dalam bentuk Relasi khusus yang memiliki atribut yang berasal dari key-key dari kedua Entitas yang dihubungkannya. Contoh jika entitas MAHASISWA berelasi dengan KULIAH (many to many) dengan relationship MENGAJAR, maka Output yang berupa Model Basis Data Relasional yang dihasilkan adalah sebagai berikut :

$$\begin{matrix} \text{Mahasiswa} & = & \\ (*\text{nim, namamhs, alamatmhs}) & & \\ \text{Kuliah} = (*\text{kodekul, sem, namakul, sks}) & & \\ \text{Mempelajari} & & = \\ (\text{nim, kodekul, indeksnilai}) & & \end{matrix}$$

2. Relasi Tunggal (Unary Relation)

Implementasi Relasi Tunggal dari/ke Entitas yang sama akan tergantung pada Tipe Asosiasinya. Untuk Relasi Tunggal dengan Tipe Asosiasi Satu Ke Banyak dapat diimplementasikan melalui penggunaan atribut key dua kali tapi untuk fungsi yang berbeda. Jika kita memiliki Entitas A dengan dua atribut x dan y dengan x sebagai key, maka Relasi Tunggal terhadap Entitas tersebut diwujudkan dengan menambahkan kembali atribut x tersebut ke A. Karena nama atribut harus unik, maka atribut x yang kedua harus diganti namanya. Contoh jika entitas DOSEN berelasi dengan entitas DOSEN sendiri, maka Model Basis Data Relasional yang dihasilkan adalah sebagai berikut :

$$\begin{matrix} \text{Dosen} & = & \\ (*\text{KodeDos, NamaDos, KodeDosPendamping}) & & \end{matrix}$$

Untuk membedakan, atribut KodeDos pada Relationship Pendamping, digabung dengan nama Relationshipnya itu sendiri. Sehingga key tetap KodeDos dan

KodeDos dari Relationship menjadi KodeDosPendamping.

Untuk Tipe Asosiasi Banyak Ke Banyak akan diimplementasikan dengan pembentukan Relasi baru.

Model Basis Data Relasional yang dihasilkan adalah sebagai berikut :

Kuliah = (\*KodeKul, NamaKul)

KuliahPrasyarat =

(KodeKul, KodeKulKuliahPrasyarat)

3. Relasi Multi Entitas

Secara umum, Relasi Multi Entitas yang menghubungkan lebih dari dua Entitas akan diimplementasikan sebagai sebuah Relasi khusus di tambah dengan Relasi yang berasal dari Entitas yang ikut terlibat. Namun jika N buah entitas kita dapat memastikan bahwa Tipe Asosiasi parsial diantara (N-1) buah entitas dengan suatu entitas (misal X) adalah satu ke banyak, maka relasi tadi tidak perlu diwujudkan sebagai sebuah Relasi khusus dan atribut-atributnya cukup dilekatkan pada himpunan entitas X. Contoh Entitas yang terlibat DOSEN, RUANG dan KULIAH, untuk mendapatkan Model Basis Data Relasionalnya, maka akan dilakukan proses sebagai berikut :

(a) Jika Tipe Asosiasi Entitas I Ke Entitas II adalah Satu Ke Banyak dan Tipe Asosiasi Entitas III Ke Entitas II Adalah Satu Ke Banyak, Maka Atribut-atribut dilekatkan ke Entitas II

(b) Jika Tipe Asosiasi Entitas II Ke Entitas I adalah Satu Ke Banyak dan Tipe Asosiasi Entitas III Ke Entitas I Adalah Satu Ke Banyak, Maka Atribut-atribut dilekatkan ke Entitas I

(c) Jika Tipe Asosiasi Entitas I Ke Entitas III adalah Satu Ke Banyak dan Tipe Asosiasi Entitas II Ke Entitas III Adalah Satu Ke Banyak, Maka Atribut-atribut dilekatkan ke Entitas III

(d) Jika ketiga syarat di atas tidak dipenuhi maka akan dibuat Relasi khusus.

Keempat proses diatas akan ditambah dengan Relasi dari entitas yang lainnya yaitu yang ikut terlibat. Untuk Contoh diatas maka Model Basis Data Relasionalnya adalah sebagai berikut :

Dosen = (\*KodeDos, alamatDos)

Ruang =

(\*koderuang, namaruang, kapasitas)

Kuliah =

(\*kodekul, sem, namakul, sks, KodeDos, kode ruang)

4. Relasi Ganda

Untuk Relasi Ganda, dalam proses pengubahannya sama, yaitu tergantung Tipe Asosiasinya. Model Basis Data Relasionalnya adalah sebagai berikut :

Dosen =

(\*KodeDos, NamaDos)

Kuliah =

(\*kodekul, sem, KodeDos)

Menguasai =

(KodeDos, kodekul)

5. Agregasi

Agregasi sebenarnya dapat dipandang sebagaimana Relasi pada umumnya. Karena Relasi ini dibentuk dari Relasi lain yang secara kronologis lebih dulu terbentuk, maka pengimplementasiannya juga harus dilakukan setelah Relasi prasyarat tersebut terimplementasikan. Untuk selanjutnya kita tinggal meninjau Tipe Asosiasinya. Model Basis Data Relasional untuk contoh diatas adalah sebagai berikut :

Mahasiswa = (\*Nim, Nama)

Kuliah = (\*KodeKul, NamaKul)

Praktikum =

(\*kodepra, namapra, Jml\_jam)

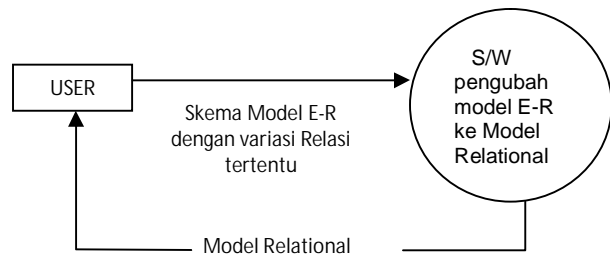
Mempelajari = (Nim, KodeKul)

Mengikuti =

(Nim. KodeKul, KodePra, nilai)

Perancangan

Untuk membatasi sistem dan menunjukan interaksi sistem dengan komponen di luar sistem, maka perlu dibuat Diagram Konteks yang merupakan gambaran sistem secara keseluruhan. Berikut adalah gambar Diagram Konteks dari perangkat lunak yang akan dibangun :



Gambar 3 Diagram Konteks

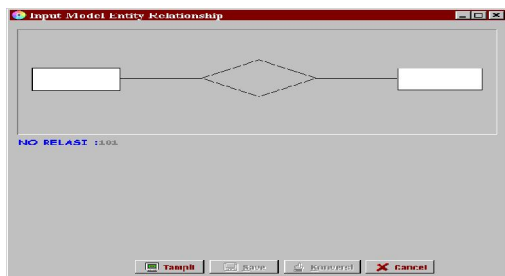
Dari rancangan diatas terlihat bahwa skema model E-R yang dikirim oleh pengguna akan langsung di konversi menjadi model relasional yang sesuai dengan tipe relasi dari model E-R nya.

Diagram konteks tersebut diatas memiliki beberapa sub-proses penting, yaitu : sub-proses pemilihan skema relasi, sub-proses melingkupi entitas dan sub-proses penentuan

tipe relasi. Jika ketiga sub-proses tersebut lengkap maka selanjutnya akan dilakukan transformasi ke model relational yang sesuai dengan aturan seperti yang dijelaskan sebelumnya.

**Layar Masukan dan Keluaran**

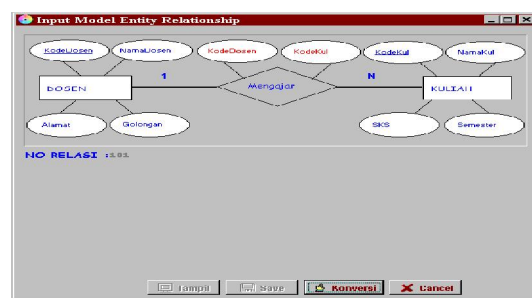
Layar masukan dan keluaran akan dirancang sedemikian rupa sehingga diharapkan mudah digunakan pemakai. Skema Model vE-R yang akan ditampilkan pada layar masukan tergantung dari variasi relasi yang dipilih. Contoh Layar masukan untuk Relasi Biner dapat dilihat pada gambar 4.



Gambar 4. Contoh Layar Masukan

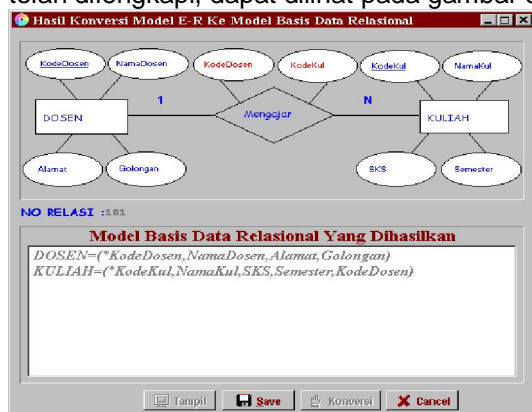
Untuk Mendapatkan Model Basis Data Relasional, pemakai tinggal menekan tombol "Konversi" selanjutnya akan ditampilkan hasilnya berupa Model Basis Data Relasional. Contoh dapat dilihat pada gambar 6.

Dari contoh di atas, pemakai bisa menyimpan hasil konversi tersebut dengan cara menekan tombol "Save" atau membatalkannya dengan menekan tombol "Cancel". Pada No Relasi, Karakter pertama menunjukkan variasi relasi dan dua karakter terakhir menunjukkan jumlah Model Basis Data Relasional yang telah dihasilkan untuk variasi relasi tersebut.



Gambar 6. Contoh Layar Keluaran

Dari contoh layar masukan diatas, pemakai tinggal memilih yaitu, menampilkan hasil konversi yang pernah dibuat, atau merancang Model E-R yang baru. Untuk Menampilkan Model E-R yang pernah dibuat, pemakai tinggal menekan tombol "Tampil". Sedangkan untuk merancang baru, pemakai tinggal melengkapi entitas, atribut, key serta relationship. Contoh Layar Masukan yang telah dilengkapi, dapat dilihat pada gambar 5.



Gambar 5 Contoh Layar Masukan yang telah diisi

Pada Layar masukan di atas terdapat No Relasi sebanyak tiga angka. Angka pertama menunjukkan kode variasi relasi, sedangkan dua angka berikutnya menunjukkan No keberapa dari Model E-R yang di konversi tersebut.

**Implementasi**

1. Penyiapan Antarmuka

Implementasi perangkat lunak ini dilakukan dengan membuat *file project* relasional.dpr. Suatu *file project* merupakan gabungan dari berbagai modul program yang disebut sebagai unit program. Setiap unit program pada delphi memiliki ekstensi \*.pas, yang berfungsi untuk melaksanakan pengendalian terhadap proses yang dilakukan. Setiap unit program disertai dengan *file* yang berisi data tentang layar antarmuka yan disebut *Form* yang disimpan pada *file* dengan ekstensi \*.dfm. Pada implementasi akan dibentuk *file* unit dan beberapa *file form* dimana setiap *form* membentuk antarmuka pemakai. Untuk lebih jelasnya dapat dilihat pada tabel 1. sebagai berikut :

Tabel 1 Nama Unit dan Nama Form yang digunakan

File Unit	File Form	Deskripsi
UnitUtamaRelasi	FormUtama	Untuk menampilkan Form Utama
UnitModelRelasi	FormKonversi	Untuk menampilkan relasi biner
UnitModelAG	FormKonversiAG	Untuk menampilkan

		kan agregasi
UnitModelRG	FormKonversiRG	Untuk menampilkan relasi ganda
UnitModelRME	FormKonversiRME	Untuk menampilkan relasi multi entitas
UnitModelIRT	FormKonversiIRT	Untuk menampilkan relasi tunggal
UnitModelSpesial	FormKonversiSpesial	Untuk menampilkan spesialisasi
UnitPembuat	FormPembuat	Untuk menampilkan tentang pembuat
UnitPilih	FormPilih	Untuk memilih variasi relasi

```

LEn11.Font.Style:=[FsUnderline];

LEn11.Caption:=Copy(EdEn11.Text,2,Length(EdEn11.Text)-1);

EdR1.Text:=Copy(EdEn11.Text,2,Length(EdEn11.Text)-1);
End Else
Begin
LEn11.Font.Style:=[];
End;
If Copy(EdEn21.Text,1,1) = '*' Then
Begin

LEn21.Font.Style:=[FsUnderline];

LEn21.Caption:=Copy(EdEn21.Text,2,Length(EdEn21.Text)-1);

EdR2.Text:=Copy(EdEn21.Text,2,Length(EdEn21.Text)-1);
End Else
Begin
LEn21.Font.Style:=[];
End;
    
```

b) Konversi Model E-R ke Model Basis Data Relasional, secara umum akan sangat tergantung pada tipe asosiasi dari variasi relasinya. Berikut adalah contoh Algoritma Konversi Model E-R ke Model Basis Data Relasional :

Algoritma 2 Konversi Model E-R ke Model Basis Data Relasional

```

{ Algoritma Konversi
IS : Model E-R lengkap dan siap dikonversi
FS : Model Basis Data Relasional hasil konversi }

Temp:='';
For i:=1 to 4 do
Begin
Case i of
1 : atribut[i]:=EdEn11.Text;
2 : atribut[i]:=EdEn12.Text;
3 : atribut[i]:=EdEn13.Text;
4 : atribut[i]:=EdEn14.Text;
End;
If atribut[i] <> '' then
Begin
Temp:=temp+atribut[i];
Temp:=temp + ',';
End;
End;
Temp1:='';
For i:=1 to 4 do
Begin
Case i of
1 : atribut[i]:=EdEn21.Text;
2 : atribut[i]:=EdEn22.Text;
3 : atribut[i]:=EdEn23.Text;
4 : atribut[i]:=EdEn24.Text;
End;
If atribut[i] <> '' then
Begin
Temp1:=temp1+atribut[i];
Temp1:=temp1 + ',';
End;
End;
Temp2:='';
For i:=1 to 4 do
Begin
    
```

2. Implementasi Program

Implementasi program dilakukan sesuai dengan hasil analisis yang telah dilakukan, dimana pada perangkat lunak yang akan dibangun ini terdapat bagian utama yang merupakan bagian terpenting dari perangkat lunak ini, yaitu penentuan foreign key dan perubahan Model E-R ke Model Basis Data Relasional berdasarkan tipe asosiasi dan variasi relasinya. Pada implementasi ini tidak akan ditulis semua, melainkan hanya sebagian contoh saja. Implementasi lengkap dapat dilihat pada listing program.

a) Penentuan *foreign key* dilakukan dengan melakukan pemeriksaan terhadap setiap atribut pada masing-masing entitas, jika ada tanda bintang maka tampilkan pada atribut relasinya. Berikut contoh penentuan *foreign key* :

Algoritma 1. Penentuan Foreign Key

```

{ Algoritma Penentuan Foreign Key
IS : Atribut setiap entitas sudah lengkap
FS : Atribut key pada entitas, dijadikan foreign key pada Relationship }
If Copy(EdEn11.Text,1,1) = '*' Then
Begin
    
```

<pre> Case I of   1 : atribut[i]:=EdR1.Text;   2 : atribut[i]:=EdR2.Text;   3 : atribut[i]:=EdR3.Text;   4 : atribut[i]:=EdR4.Text; End; If atribut[i] &lt;&gt; ''then Begin   Temp2:=temp2+atribut[i];   Temp2:=temp2 + ','; End; End; {ALTERNATIF PELEBURAN 1 KE 1} if (EdTipe1.Text='1')and(EdTipe2.Text='1') then Begin   MemoMR.Lines[0]:=EdEntitas1.text + '=' + '(' + copy(temp,1,length(temp)-1)   ');   MemoMR.Lines[1]:=EdEntitas2.text + '=' + '(' + copy(temp1,1,length(temp1)-1) + ',' + EdR1.text + ')'; End; {END ALGORITMA KONVERSI 1 KE 1}  {ALTERNATIF PELEBURAN 1 KE N} if (EdTipe1.Text='1')and(EdTipe2.Text='N') then Begin   MemoMR.Lines[0]:=EdEntitas1.text + '=' + '(' + copy(temp,1,length(temp)-1)   ');   MemoMR.Lines[1]:=EdEntitas2.text + '=' + '(' + copy(temp1,1,length(temp1)-1) + ',' + EdR1.text + ')'; End; {END ALGORITMA KONVERSI N KE 1}  {ALTERNATIF PELEBURAN N KE N} if (EdTipe1.Text='N')and(EdTipe2.Text='M') then Begin   MemoMR.Lines[0]:=EdEntitas1.text + '=' + '(' + copy(temp,1,length(temp)-1)   ');   MemoMR.Lines[1]:=EdEntitas2.text + '=' + '(' + copy(temp1,1,length(temp1)-1) + ',' + EdR11.text + ')';   MemoMR.Lines[2]:=EdRelasi.text + '=' + '(' + copy(temp2,1,length(temp2)-1) + ')'; End; {END ALGORITMA KONVERSI N KE M} </pre>	Baru	Perancangan Baru kemudian pilih variasi yang diinginkan
	Membuka Hasil Rancangan	Pilih menu File, kemudian pilih variasi relasi yang diinginkan untuk ditampilkan, lalu Klik 'Tampil' dan masukan no relasi yang ingin ditampilkan lalu pilih OK
	Mengubah Model E-R ke Model Basis Data Relasional	Setelah Model E-R lengkap diisi, klik 'Konversi'
	Menyimpan	Setelah hasil konversi muncul tekan 'Save'
	Menghapus	Pilih variasi Relasi, lalu tampilkan berdasar no relasi, setelah muncul klik 'Delete' dan muncul kotak dialog lalu pilih OK
	Tentang Pembuat	Pilih Menu Tentang lalu klik 'Pembuat'
	Tentang Program	Pilih Menu Tentang lalu klik 'Program'
Keluar	Pilih Menu File lalu klik keluar	

**Kesimpulan**

- Adapun kesimpulan yang dihasilkan adalah sebagai berikut :
1. Proses mengubah Model E-R ke Model Basis Data Relasional secara umum sangat tergantung kepada tipe asosiasi dari variasi relasinya.
  2. Hasil keluaran adalah berupa Model Basis Data Relasional sesuai dengan Model E-R yang dimasukkan sebagai input.
  3. Perangkat lunak ini mampu mengubah Model E-R menjadi Model Basis Data Relasional berdasar variasi relasi dan tipe asosiasinya.
  4. Perangkat lunak ini dapat membantu dalam proses perancangan Basis Data.
  5. Perangkat lunak ini belum bisa digunakan untuk kasus yang kompleks.

**Saran**

- Berdasarkan pada pengalaman dalam pengembangan Perangkat lunak ini ada beberapa saran sebagai berikut :
1. Perangkat lunak ini belum bisa menangani kasus yang kompleks, karena belum ada fasilitas edit.
  2. Perangkat lunak ini belum sampai ke tahap pembuatan tabel dari Model Basis Data Relasional yang dihasilkan,

**3. Implementasi Penggunaan**

Perangkat lunak ini dibangun dengan tujuan membantu pemakai dalam merancang basis data, yang diharapkan bisa dimanfaatkan dan membantu dalam proses mengubah dari Model E-R ke Model Basis Data Relasional. Pemakai sebaiknya sudah memiliki kemampuan menggunakan Sistem Operasi Windows dan minimal mengetahui tentang perancangan basis data. Untuk lebih jelas mengenai penggunaan perangkat lunak ini dapat dilihat pada tabel 2 sebagai berikut :

**Tabel 2.** Kegiatan dan Aksi berdasarkan Menu Program

Kegiatan	Aksi
Membuat Rancangan	Pilih menu File, lalu pilih



penyusun menyarankan agar kelak bisa dikembangkan sampai pada tahap pembuatan tabel (Create Table).

3. Primary key dari setiap entitas hanya bisa satu, disarankan bisa dikembangkan untuk menangani kasus yang primary key-nya lebih dari satu.

## Reference

1. Batini Ceri Navathe, **Conceptual Database Design An Entity-Relationship Approach**, The Benjamin/Cummings Publishing Company, Inc, 1992
  2. Djoko Pramono, **Mudah Menguasai Delphi 2.0**, jilid satu, PT Elex Media Komputindo, Jakarta 1998
  3. Djoko Pramono, **Mudah Menguasai Delphi 2.0**, jilid dua, PT Elex Media Komputindo, Jakarta 1998
  4. Fathansyah, **Basis Data**, Informatika Bandung, 1999
  5. Harianto Kristanto, **Konsep dan Perancangan Database**, Andi Yogyakarta, 1999
  6. Jogiyanto, H.M., **Analisis & Disain Sistem Informasi Pendekatan Terstruktur dan Praktek Aplikasi Bisnis**, Andi Yogyakarta, 1999
  7. Niklaus Wirth, **Algoritma + Struktur Data = Program**, Andi Yogyakarta, 1997
- Presman, R.S., **Software Engineering A Practitioner's Approach**, McGraw-Hill International Editions, 1992