

# Perancangan dan Pembuatan Aplikasi *ERD Generator* Notasi ORM dari Skrip Basis Data Oracle Berbasis J2EE

**Leo Willyanto Santoso**

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra

e-mail : [leow@petra.ac.id](mailto:leow@petra.ac.id)

## **ABSTRAK:**

*Kehadiran pemrosesan basis data diperlukan oleh berbagai institusi dan perusahaan. Basis data tidak hanya mempercepat pemerolehan informasi, tetapi juga dapat meningkatkan pelayanan kepada pelanggan. Bagi perusahaan, keuntungan seperti ini dapat meningkatkan daya saingnya terhadap perusahaan lain. Hal ini pulalah yang mendorong banyak perusahaan yang menggunakan pemrosesan manual mulai beralih memanfaatkan basis data.*

*Sejalan dengan hal di atas, proses reverse engineering terhadap suatu basis data menjadi suatu kebutuhan bagi perancang basis data untuk mengetahui struktur dari sebuah basis data. Struktur tersebut biasanya dimodelkan dalam bentuk Entity Relationships Diagram (ERD). Penggambaran struktur basis data dalam sebuah ERD dapat menggunakan berbagai notasi agar menjadi lebih mudah dimengerti. Salah satu notasi yang pendekatannya mudah dipahami adalah notasi Object Role Modeling (ORM) diagram.*

*Dengan melakukan pembalikan terhadap proses pemetaan ke skema relasi dari suatu basis data maka pembangkitan ERD dari suatu skrip pembuatan basis data Oracle dapat dilakukan. Agar lebih fleksibel, maka aplikasi yang dibangun berupa aplikasi berbasis web dengan menggunakan teknologi Servlets yang dimiliki oleh Java<sup>TM</sup> 2 SDK.*

**Kata kunci:** Basis data, Model data konseptual, ERD, Oracle, dan Servlets

## **ABSTRACT:**

*The existing of database processing is needed by many institutions and companies. Database is not only to get information faster, it is also enlarging their service to customer. For companies, this advantage can increase competency. Because of this reason, many companies using manual processing turn to database.*

*As mentioned above, database reverse engineering process has become a necessity for database developers to understand the structure of any databases. Commonly, this structure is modeled in some notations of Entity Relationships Diagram (ERD). The graphical visualization of database structure in an ERD can use many notations, so it is easy to understand. One of the approaches, which are easy to understand, is Object Role Modeling (ORM) diagram.*

*By reverse engineering the mapping process to relation schema of a database, ERD generation from a data definition language script of Oracle can be done. For more flexibility, this application is constructed web based with Servlets technology that is propertied by Java<sup>TM</sup> 2 SDK.*

**Keyword:** Database, Conceptual data model, ERD, Oracle, and Servlets

## **1. PENDAHULUAN**

Proses pembangunan sebuah basis data mempunyai beberapa tahapan. Salah satu tahapan yang terpenting adalah tahap mendesain basis data yang akan dibuat. Tujuan utama dalam mendesain sebuah basis data adalah:

1. Menggambarkan data dan hubungan antar data yang dibutuhkan oleh aplikasi utama dalam sistem informasi yang dibangun dan kelompok pemakainya.
2. Memberikan sebuah model data yang mendukung beberapa transaksi yang dibutuhkan terhadap data tersebut.

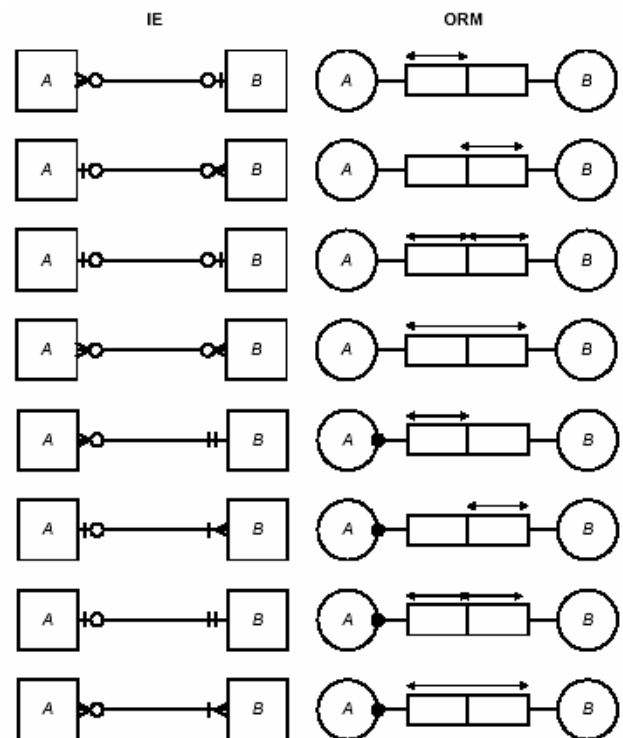
3. Menspesifikasikan sebuah desain minimal yang terstruktur untuk mencapai kebutuhan kinerja yang telah ditentukan terhadap sistem tersebut.

Agar dapat mencapai tujuan tersebut maka perlu dibangun sebuah model ER yang menggambarkan model data konseptual untuk memudahkan proses desain basis data yang diinginkan. Sebuah model data konseptual adalah sekumpulan konsep yang menjelaskan struktur dari sebuah basis data dan transaksi yang berlaku terhadapnya. Tujuan utama membangun sebuah model data adalah untuk mendukung sebuah persepsi pemakai terhadap data dan untuk menyembunyikan aspek – aspek yang lebih teknis yang berhubungan dengan desain basis data yang akan dibangun. Model data konseptual ini terlepas dari sistem manajemen basis data (DBMS) dan *platform* perangkat keras yang digunakan untuk mengimplementasikan basis data tersebut.

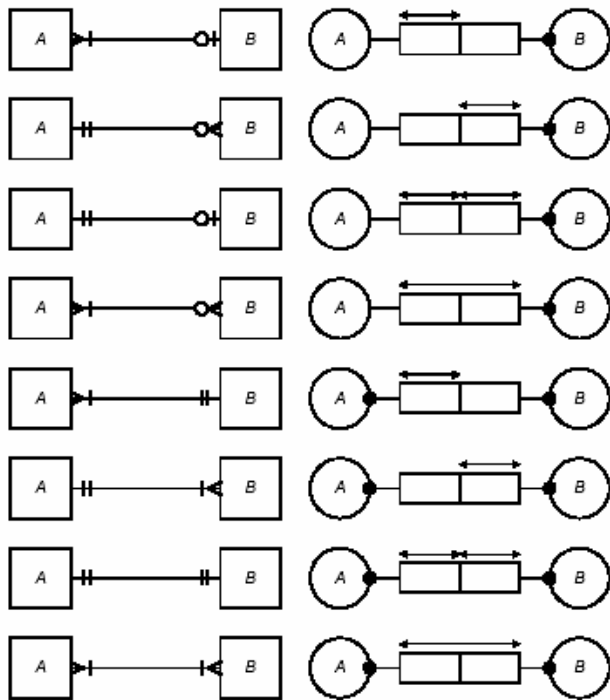
Dengan semakin berkembangnya kemajuan jaman yang ditunjang dengan pesatnya kemajuan teknologi informasi, maka ada kalanya sebuah perusahaan berkeinginan untuk memperbaiki proses bisnis yang dijalankannya agar sesuai dengan permintaan pasar. Ketika hal itu terjadi maka perlu kiranya untuk meninjau kembali sistem yang sudah ada termasuk juga perubahan yang perlu dilakukan terhadap struktur basis data yang digunakan saat ini. Untuk mempermudah hal tersebut maka diperlukan suatu alat untuk memperoleh kembali model data konseptual dalam bentuk *Entity Relationships Diagram* (ERD) notasi *Object Role Modeling* (ORM) dari basis data yang dimiliki saat ini. Salah satu cara untuk memperoleh ERD tersebut yaitu dengan melakukan *reverse engineering* terhadap skrip yang digunakan untuk membangun basis data dari sistem yang sudah ada pada perusahaan tersebut.

## 2. NOTASI ERD

ERD digambarkan dengan menggunakan notasi–notasi tertentu. Beberapa notasi yang biasa digunakan untuk membuat ERD adalah notasi Chen, Martin, Korth, dan Bachman. Pada penelitian ini digunakan notasi *ORM*. Pemilihan notasi ORM karena pada notasi ini lebih menyederhanakan proses desain dengan menggunakan *natural language*. Dengan mengekspresikan sebuah model dalam *natural concept*, seperti *object* dan *role*, maka diperoleh pendekatan konseptual dalam pemodelan. Untuk lebih memahami simbol-simbol yang digunakan pada notasi ORM, dibuat perbandingan antara notasi ORM dengan notasi *Information Engineering* (IE) yang dapat dilihat pada gambar 1 dan gambar 2.



Gambar 1. Notasi IE dan Notasi ORM



Gambar 2. Notasi IE dan Notasi ORM (lanjutan)

### 3. Structured Query Language

*Structured Query Language (SQL)* adalah bahasa basis data secara luas yang mempunyai pernyataan untuk definisi data, *query* dan *update*. Sedangkan pernyataan yang digunakan dalam SQL untuk membentuk suatu definisi data dalam basis data disebut *Data Definition Language (DDL)*.

#### 3.1 Perintah CREATE TABLE

Perintah **CREATE TABLE** digunakan untuk menspesifikasikan relasi baru dengan memberi nama dan menspesifikasikan atribut dan pembatas.

Masing-masing atribut dispesifikasikan dengan memberikan nama atribut, tipe data dan beberapa kemungkinan pembatas.

#### 3.2 Perintah ALTER TABLE

Definisi dari tabel dapat diubah dengan menggunakan perintah **ALTER TABLE**.

Perintah tersebut memungkinkan untuk menambah atau menghapus kolom yang ada, mengubah ukuran dan menambahkan atau menghapus tabel dengan paksa, dan juga menambahkan pembatas pada suatu tabel.

### 3.3 Skrip Basis data Oracle

Skrip basis data Oracle adalah sekumpulan perintah-perintah SQL sederhana yang digunakan untuk melakukan berbagai operasi yang biasa dilakukan terhadap sebuah basis data dalam Oracle seperti peningkatan kinerja SQL, peningkatan kinerja basis data, melakukan *back up*, membuat spesifikasi relasi-relasi dalam basis data dan lain sebagainya.

Karena perangkat ini berhubungan dengan relasi-relasi dalam basis data, maka skrip basis data Oracle yang digunakan ialah skrip yang bertujuan untuk membangun relasi dalam basis data Oracle. Adapun perintah-perintah yang biasanya terdapat dalam skrip tersebut adalah perintah untuk membuat skema lengkap dengan spesifikasi tabel, atribut dan batasan-batasannya. Skrip inilah yang nantinya diubah menjadi ERD yang dapat menggambarkan model konseptual dari sebuah basis data Oracle.

### 4. Java 2 Platform, Enterprise Edition (J2EE)

Java 2 Platform, Enterprise Edition membuat standar pengembangan aplikasi multitier berkelas enterprise berbasis komponen. Java merupakan bahasa yang sangat baik dalam menulis komponen pada sisi *server*. *Server* dapat di-*install* dengan sistem operasi yang berbeda-beda, sistem operasi yang dipakai oleh *server* yang satu belum tentu sama dengan sistem operasi yang dipakai pada *server* lainnya. Oleh karena itu dibutuhkan komponen atau aplikasi yang dapat berjalan pada beberapa *server*. Pengembang cukup membuat aplikasi untuk satu *server* tapi dapat dijalankan pada semua *server*.

J2EE merupakan kumpulan servis *middleware* handal yang meringankan tugas pengembang aplikasi pada sisi *server*. Teknologi yang terkandung didalamnya antara lain adalah: RMI, J2SE, EJB, JNDI, JMS, JDBC, JTA dan JTS, Java Mail, Connector, Servlet dan JSP, Java idl, serta XML.

#### 4.1 Java Server Pages (JSP)

JSP adalah salah satu solusi pengembangan aplikasi berbasis *web* berupa scripting yang diletakkan di *server*. Seperti bahasa scripting lainnya, ASP, PHP, Perl dan sebagainya. JSP memungkinkan pembangunan *web* yang dinamis seperti menampilkan data dari suatu basis data. JSP mempunyai beberapa keunggulan diantaranya adalah dapat mengakses semua *library*, yaitu fungsi yang ditulis dalam bahasa Java. Seperti solusi scripting yang lain, JSP juga menyediakan fasilitas penanganan *session*, *cookies*, dan lain sebagainya.

#### 4.2 Servlets

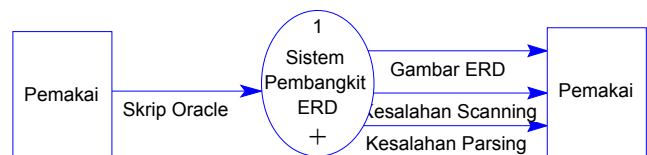
*Servlet* adalah suatu *interface* atau aplikasi yang menangani *request* dari *client web* dan memberikan respon ke *client*. *Servlets* identik dengan CGI yaitu berupa *file binary* yang dijalankan oleh *web server* ketika ada *request* terhadap *Servlets* tersebut. *Servlets* ditulis dalam bahasa Java dan seperti ASP mempunyai kemampuan untuk mengakses semua *library* Java.

### 5. IMPLEMENTASI SISTEM

Aplikasi pembangkit ERD merupakan aplikasi yang bertugas untuk membangkitkan ERD dari sebuah skrip pembuatan basis data Oracle. Aplikasi ini mempunyai beberapa proses yang terbagi dalam tiga tahapan penting yaitu tahap *scanning* skrip basis data, tahap parsing terhadap hasil *scanning* yang telah dilakukan dan yang terakhir yaitu tahap pembangkitan ERD yang ditampilkan pada halaman *web*.

#### 5.1 Desain Sistem Pembangkitan ERD

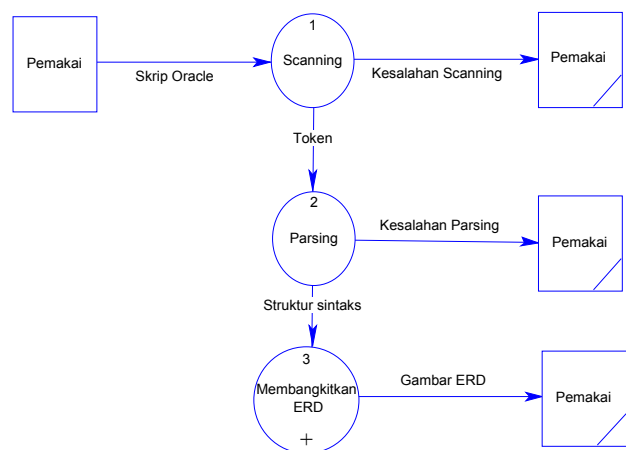
Sistem pembangkit ERD merupakan sebuah sistem yang menerima *input* dari pemakai dan menampilkan hasil pembangkitan (*output*) kepada pemakai. Diagram alir tingkat 0 sistem pembangkit ERD dapat dilihat pada gambar 3.



Gambar 3. Diagram Aliran Data Tingkat 0

Sistem menerima sebuah *input* dari pengguna yaitu skrip pembuatan basis data dan menghasilkan tiga pilihan *output* yaitu gambar ERD jika pada proses *scanning* dan *parsing* tidak terjadi kesalahan, kesalahan *scanning* jika pada proses *scanning* terjadi kesalahan dan kesalahan *parsing* jika pada proses *parsing* terjadi kesalahan.

Sistem pembangkit ERD memiliki beberapa proses yang digambarkan pada diagram aliran data tingkat 1 pada gambar 4.



Gambar 4. Diagram Aliran Data Tingkat 1

Proses-proses yang terdapat dalam sistem ini adalah:

1. *Scanning*

Proses ini berfungsi sebagai *Scanner* yang bertugas untuk membaca karakter-karakter yang terdapat pada skrip sebagai *input*-nya dan mengelompokkannya dalam *token-token*. *Output* proses ini berupa *token-token* yang dikirimkan ke proses *parsing*. Jika terjadi kesalahan dalam proses ini maka *scanning* akan mengirimkan pesan kesalahan sebagai *output*-nya.

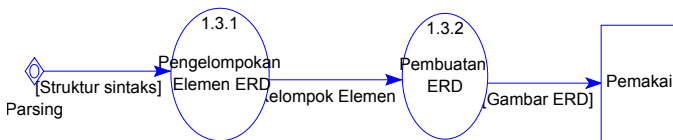
## 2. *Parsing*

Proses ini berfungsi sebagai *Parser* yang bertugas untuk mengelompokkan *token-token* hasil *scanning* menjadi struktur sintaks. *Input* proses ini berupa *token-token* yang dihasilkan proses *scanning*. *Output* dari proses ini adalah struktur sintaks yang dikirim ke proses selanjutnya.

## 3. *Membangkitkan ERD*

Proses ini membangkitkan ERD berdasarkan struktur sintaks yang diterima sebagai *input* yang diberikan oleh proses *parsing*. *Output* dari proses ini adalah gambar ERD yang berhasil dibentuk dan dikirimkan ke pemakai.

Agar dapat membangkitkan suatu ERD maka dalam proses membangkitkan ERD terdapat dua proses lagi seperti terlihat pada gambar 5.



**Gambar 5. Diagram Aliran Data Tingkat 2 (Proses Scanning)**

Proses pengelompokkan elemen ERD bertugas untuk mengelompokkan elemen-elemen ERD dari struktur sintaks yang diterima dan menghasilkan kelompok-kelompok elemen sebagai *output*-nya. Kelompok elemen tersebut selanjutnya digunakan oleh proses pembuatan ERD untuk membangkitkan gambar ERD-nya. *Output* dari proses pembuatan ERD berupa gambar yang dikirimkan ke pemakai.

## 5.2 Pembuatan Spesifikasi *Scanner* dan *Parser*

*Scanning* merupakan proses awal yang dilakukan oleh aplikasi ini. Proses ini bertugas untuk membaca karakter-karakter yang terdapat pada skrip yang dibangkitkan dan mengelompokkannya ke dalam kelompok-kelompok *token*. *Token-token* tersebut diproses lebih lanjut oleh *Parser* dalam proses *parsing*. Proses *scanning* dilakukan oleh *Scanner*.

*Parsing* merupakan proses kedua yang dilakukan setelah hasil *scanning* terhadap skrip didapatkan. Proses ini bertugas untuk mengelompokkan *token-token* yang dihasilkan oleh proses *scanning* bersama-sama ke dalam struktur sintaks. Proses *parsing* ini dilakukan oleh sebuah *Parser*.

## 5.3 Pengelompokan Elemen Pembentuk ERD

Untuk membangkitkan ERD dari perintah-perintah SQL yang dispesifikasikan dalam skrip basis data, perlu adanya pengelompokkan elemen-elemen dasar yang membangun ERD berdasarkan spesifikasi yang terdapat pada skrip. Ada lima elemen dasar yang membangun ERD dari sebuah basis data. Kelima elemen tersebut adalah:

1. Entitas
2. Atribut
3. Atribut yang bertindak sebagai *primary key*
4. Relasi yang mempunyai kardinalitas *one-to-one*(1:1) dan *one-to-many*(1:N)
5. Relasi yang mempunyai kardinalitas *many-to-many* (M:N)

*Class Interpreter* berguna untuk mengelompokkan elemen-elemen dasar berdasarkan hasil *scanning* dan *parsing* yang dilakukan oleh *class lexer* dan *class parser*. Adapun tipe data yang digunakan untuk menyimpan elemen-elemen tersebut adalah tipe data *Vector*. *Vector* ini menyimpan informasi yang dipergunakan untuk setiap elemen dasar.

*Class Interpreter* membaca hasil *scanning* dan *parsing* yang telah dilakukan oleh *Scanner* dan *Parser*, lalu menentukan jenis-jenis elemen dasar yang ada dari perintah SQL dan

mengelompokkannya ke dalam kelompok-kelompok tertentu. Ketika *class* ini menjumpai perintah pembuatan suatu tabel, maka hal pertama yang dilakukan ialah memasukkan tabel tersebut ke dalam kelompok entitas terlebih dahulu, kemudian menyimpan kolom-kolom pada tabel tersebut dan dikelompokkan ke dalam kelompok elemen atribut. Jika ditemukan perintah yang mendefinisikan bahwa suatu kolom tertentu adalah *primary key* untuk suatu tabel maka nama kolom tersebut akan dikelompokkan ke dalam elemen atribut yang bertindak sebagai *primary key*. Namun jika ditemukan perintah yang mendefinisikan suatu kolom adalah *foreign key* untuk suatu tabel, maka *class* mengumpulkan informasi untuk membangun suatu relasi yang berkaitan dengan kolom tersebut.

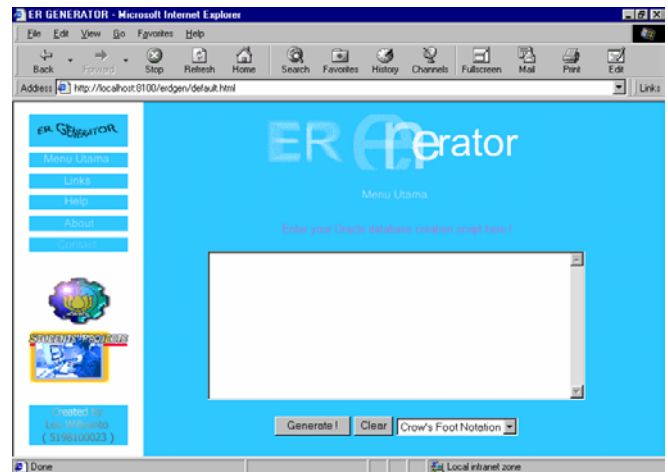
#### 5.4 Pembangkitan ERD

Untuk memperoleh suatu ERD, perlu dilakukan suatu analisis terhadap data elemen dasar yang telah diperoleh dari *class Interpreter*. Elemen-elemen dasar untuk entitas, atribut yang bertindak sebagai *primary key* dan relasi M:N dapat dengan mudah dibangkitkan dengan mengambil data dan informasi yang tersimpan dalam masing-masing *Vector* penyimpanan kelompok elemen tersebut.

Relasi 1:N yang akan dibangkitkan terdiri dari dua jenis yaitu relasi 1:N dengan batasan partisipasi parsial-total (kardinalitas 1 mempunyai partisipasi parsial dan kardinalitas N mempunyai partisipasi total) atau semua partisipasinya parsial. Penentuan batasan partisipasi diperoleh dari informasi mengenai sifat dari suatu kolom pada suatu tabel. Sifat yang dimaksud disini ialah sifat dimana suatu kolom harus mengandung suatu nilai atau tidak, yang ditentukan dengan memberikan keterangan “*null*” atau “*not null*” pada kolom yang bersangkutan. Jika kolom yang menjadi *foreign key* adalah “*not null*” maka batasan partisipasinya adalah parsial-total. Namun jika “*null*” maka batasan partisipasinya adalah semua parsial.

## 6. UJI COBA

Halaman utama dari aplikasi pembangkit ERD dapat dilihat pada gambar 6.

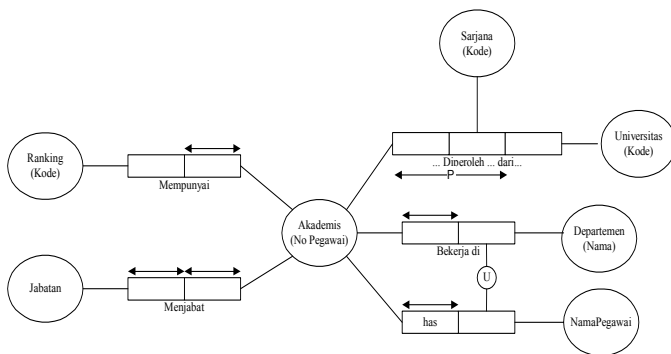


Gambar 6. Halaman Utama Pembangkit ERD

Untuk membangkitkan sebuah ERD dari sebuah skrip pembuatan basis data Oracle, skrip tersebut harus dituliskan dalam halaman *web* seperti pada gambar 6.

ERD dari skrip yang telah ditulis pada halaman pertama akan muncul setelah tombol “Bangkitkan !” ditekan. Dengan menekan tombol tersebut, maka skrip yang dimasukkan akan diolah dan hasil pengolahan akan ditampilkan pada halaman berikutnya. Jika skrip yang ditulis benar maka halaman berikutnya akan menampilkan gambar ERD dari skrip tersebut, namun jika terdapat kesalahan pada skrip yang diberikan maka yang ditampilkan adalah pesan kesalahan yang menginformasikan letak kesalahan yang terdapat pada skrip tersebut.

Contoh hasil pembangkitan ERD dari basis data Universitas dengan menggunakan notasi ORM dapat dilihat pada gambar 7.



**Gambar 7. Hasil Pembangkitan ERD Notasi ORM**

## 7. KESIMPULAN DAN SARAN

Beberapa kesimpulan yang dapat ditarik dari penelitian ini adalah sebagai berikut:

1. ERD hasil bangkitan aplikasi pembangkit ERD belum dapat menggambarkan semua simbol notasi dikarenakan adanya kekurangan informasi dari skrip basis data akibat adanya proses pemetaan ke skema relasi.
2. Aplikasi Pembangkit ERD mempermudah pembuatan dokumentasi model konseptual dari suatu basis data. Hal ini sangat membantu ketika *user* membutuhkan model konseptual dari suatu basis data yang dimilikinya.
3. Dengan memanfaatkan teknologi *web* dan Java serta penggunaan tool SableCC, proses perancangan dan pembuatan aplikasi ini menjadi lebih mudah. Aplikasi ini dapat dengan mudah digunakan karena berbasis *web*, sehingga pengguna hanya memerlukan *browser internet*, tanpa harus tergantung pada sistem operasi tertentu.

Beberapa saran untuk pengembangan lebih lanjut dari penelitian ini yaitu:

1. Aplikasi Pembangkit ERD dapat dikembangkan lebih lanjut dengan menambahkan beberapa notasi yang lain misalnya saja notasi Chen, Korth, Martin, dan IDEF1X.
2. Aplikasi Pembangkit ERD dapat dikembangkan menjadi pembangkit ERD

lanjut (*Enhanced-ERD*) dengan menambahkan kemampuan untuk pembangkitan elemen tambahan antara lain untuk *sub class* dan *generalisasi*.

3. Penambahan tata bahasa dan token-token baru sangat bermanfaat untuk melengkapi tata bahasa dan token yang dapat dikenali oleh aplikasi ini.
4. Aplikasi ini dapat dilengkapi dengan fasilitas untuk melakukan *Round Trip Reverse Engineering* terhadap skema basis data Oracle.

## Daftar Pustaka

1. Aho, Alfred V., dan Jeffrey D. Ullman, *Principles Of Compiler Design*. Addison – Wesley Publishing Company, Reading, Massachusetts, 1977.
2. Aho, Alfred V., Ravi Sethi, dan Jeffrey D. Ullman, *Compilers Principles, Technique, and Tolls*. Addison – Wesley Publishing Company, 1986.
3. Allaire. *Developing Application With Jrun*, 2000.
4. Berk, Elliot, JLex: A lexycal analyzer generator for Java™, 1997, Halaman World-Wide Web URL: <http://www.cs.princeton.edu/appel/modern/java/JLex/>
5. Bloch, Cynthia dan Stephanie Bodoff. *The Java Tutorial Trail: Sevlets*, 1999. Halaman World-Wide Web URL: <http://java.sun.com/docs/books/tutorial/servlets/index>.
6. Elmasri, Ramez dan Shamkant B. Navathe, *Fundamentals of Database Systems*. 2<sup>nd</sup> ed., The Benjamin/Cummings Publishing Company Inc., 1994.



7. Gagnon, Etienne, *SableCC An Object Oriented Compilers Framework*, Thesis on Master of Science School of Computer Science, McGill University, Montreal, 1998.
8. Halpin, Terry A., *Conceptual Schema and Relational Database Design*, 2<sup>nd</sup> Edition, Prentice Hall, 1995.
9. Halpin, Terry A. 2001b, *Microsoft's new database modelling tool: Part 1*. Journal of Conceptual Modelling, August 2001 issue.
10. Halpin, Terry A. 2001c, *Microsoft's new database modelling tool: Part 2*. Journal of Conceptual Modelling, August 2001 issue.
11. Halpin, Terry A. 2001d, *Microsoft's new database modelling tool: Part 3*. Journal of Conceptual Modelling, August 2001 issue.
12. Lorentz, Diana, *Oracle8<sup>TM</sup> Server SQL Reference Release 8.0*. Oracle Corporation, 500 Oracle Parkway, Redwood city, 1997
13. McGill, Sable *SableCC version 2.16.1* 2001Halaman World-Wide Web URL: [Http://www.sablecc.org](http://www.sablecc.org)
14. Roman, Ed. "Mastering Enterprise Java Bean", *Ph.D. dissertation*, University of Cambridge 1995
15. Wu, Thomas C., *An Introduction to Object – Oriented Programming With Java*, WCB/McGraw-Hill, 1999