

APLIKASI PEMODELAN BASISDATA

TUGAS AKHIR

Oleh :

Nama	NIM
Benny Zanuwan P.G	: 3310901082
Aldino Saputra	: 3310901085

Disusun untuk memenuhi syarat kelulusan Program Diploma III



PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

BATAM

2012

LEMBAR PENGESAHAN

Batam, 29 Februari 2012

Pembimbing,

Mir'atul Khusna Mufida, S.ST

NIK. 109057

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : Benny Zanuarwan P.G

Nama : 3310901082

Adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa proyek akhir dengan judul:

Aplikasi Pemodelan Basisdata

Disusun dengan:

1. Tidak melakukan plagiat terhadap naskah karya orang lain
2. Tidak melakukan pemalsuan data
3. Tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 29 Februari 2012

Benny Zanuarwan P.G

3310901082

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : Aldino Saputra

Nama : 3310901085

Adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa proyek akhir dengan judul:

Aplikasi Pemodelan Basisdata

Disusun dengan:

1. Tidak melakukan plagiat terhadap naskah karya orang lain
2. Tidak melakukan pemalsuan data
3. Tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 29 Februari 2012

Aldino Saputra

3310901085

KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT, berkat rahmat dan hidayat-Nya, penyusun dapat menyelesaikan Tugas Akhir yang berjudul Aplikasi Pemodelan Basisdata. Aplikasi ini bertujuan untuk memudahkan dalam mengkonversi ER-Diagram menjadi Skema Relational dengan lebih cepat dan mengurangi terjadinya kesalahan.

Dalam kesempatan ini, penyusun mengucapkan terima kasih kepada :

1. Bapak Dr.Priyono Eko Sanyoto, selaku Direktur Politeknik Batam,
2. Uuf Brajawidagda, MT selaku koordinator Tugas Akhir,
3. Mir'atul Khusna Mufida, S.ST selaku pembimbing Tugas Akhir,
4. Bapak/Ibu Dosen program studi Teknik Informatika atas bimbingannya,
5. Kedua orang tua tercinta yang memberikan dukungan moral dan doa,
6. Teman-teman seperjuangan Teknik Informatika angkatan 2009 yang memberikan semangat.

Dalam penulisan ini, penyusun mengakui bahwa masih terdapat kekurangan-kekurangan dalam penyusunannya. Oleh karena itu, penyusun sangat mengharapkan bantuan dari berbagai pihak berupa kritik ataupun saran guna penyempurnaan selanjutnya. Akhir kata penyusun ucapkan terima kasih, semoga laporan ini dapat bermanfaat bagi pembaca yang ingin mengembangkan sebuah aplikasi yang serupa.

Batam, 29 Februari 2012

Penulis

ABSTRAK

APLIKASI PEMODELAN BASISDATA

Suatu organisasi membutuhkan basisdata sebagai media penyimpanan data, baik itu data produk, rekening, karyawan, maupun data yang lainnya. Dengan begitu basisdata merupakan suatu yang sangat penting bagi organisasi. Adapun pembuatan basisdata terbagi dalam 4 tahap, yaitu analisis kebutuhan, perancangan ER-Diagram, pembuatan skema relasional, dan DBMS. Pada tahap perancangan ER-Diagram kemudian mengubahnya menjadi skema relasional akan membutuhkan waktu yang lama dan memungkinkan terjadinya *human error*.

Terdapat beberapa aplikasi yang dapat membantu dalam pembuatan basisdata, seperti Power Designer dan Microsoft Visio. Aplikasi tersebut dapat digunakan untuk menggambar ER-Diagram, namun tidak bisa langsung mengonversi menjadi skema relasional. Untuk itu dibuatlah Aplikasi Pemodelan Basisdata, yaitu aplikasi yang dapat menggambar ER-Diagram sekaligus mengonversi menjadi skema relasional. Dengan demikian aplikasi ini dapat menangani 2 tahap dari 4 tahap pembuatan basisdata, yaitu perancangan ER-Diagram dan pembuatan skema relasional menjadi lebih cepat.

Setelah dilakukan implementasi terhadap aplikasi maka didapatkan aplikasi yang dapat menggambar ER-Diagram dan kemudian dapat mengkonversi gambar yang telah dibuat menjadi Skema Relasional untuk selanjutnya dapat diterjemahkan secara langsung menjadi kalimat SQL yang dijalankan pada DBMS.

Kata kunci : ER-Diagram, Skema Relasional, Konversi

ABSTRACT

DATABASE MODELING APPLICATION

Organization needs a database as data storage media. Database is very important for the organization. The Develop process of database was divided into 4 stages: requirements analysis, ER design, relational schema, and create DBMS using SQL language. When creating ER design stage and then convert the diagram to a relational schema will need a long process and this procedure may be possible cause many human error.

There are several applications to design the databases, such as Power Designer and Microsoft Visio. Both of two applications above can be used to draw the ER design, but can not directly convert into a relational schema. So Applications was made for Modeling Database, the specification are it can draw ER design than directly converting to relational schema. The application can handle half from all 4 process to design the databases.

After implementation of the application, that applications can be used to draw the ER design, and then it can be converted into a relational schema, and then it can directly translate into SQL query and running in DBMS.

Keywords : ER design, Relational Schema, Convert

DAFTAR ISI

Halaman Judul	i
Lembar Pengesahan.....	ii
Lembar Pernyataan	iii
Kata Pengantar	v
Abstrak.....	vi
Abstract.....	vii
Daftar Isi	viii
Daftar Gambar	xi
Daftar Tabel	xii
Bab I Pendahuluan	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	2
I.3 Batasan Masalah.....	2
I.4 Tujuan	2
Bab II Tinjauan Pustaka.....	3
II.1 Basisdata.....	3
II.1.1 Pengertian Basisdata	3
II.1.2 Sistem Basisdata	4
II.2 Perancangan Basisdata	5
II.3 Bahasa Java.....	10
II.3.1 Bahasa pemrograman berbasis Java	10
II.3.2 NetBeans IDE.....	10
Bab III Analisis dan Perancangan.....	12
III.1 Batasan Sistem	12
III.2 Gambaran Kerja Sistem Aplikasi Pemodelan Basisdata.....	12
III.3 Use Case Diagram Aplikasi Pemodelan Basisdata	14
III.3.1 Skenario Use Case	14
III.4 Robustness Diagram.....	15
III.4.1 Robustness Diagram Use Case Menggambar ER-Diagram	15

III.4.2	Robustness Diagram Use Case Mengkonversi ER-Diagram ke Skema Relational	16
III.4.3	Robustness Diagram Use Case Menampilkan ER-Diagram	16
III.4.4	Robustness Diagram Use Case Menyimpan Informasi Design Basisdata	16
III.5	Class Diagram	17
III.6	Interaction Sequence Diagram	17
III.6.1	Menggambar ER-Diagram	18
III.6.2	Mengkonversi ER-Diagram ke Skema Relational.....	19
III.6.3	Menampilkan ER-Diagram	20
III.6.4	Menyimpan Informasi Design Basisdata.....	21
III.7	Rancangan Kelas Rinci	22
III.7.1	Kelas guiMenggambar	22
III.7.2	Kelas kontrolerGambar	22
III.7.3	Kelas kontrolerKonversi.....	22
III.7.4	Kelas kontrolerSimpan	22
III.7.5	Kelas File	22
III.8	Algoritma.....	23
III.8.1	Algoritma Gambar	23
III.8.2	Algoritma Konversi	23
III.8.3	Algoritma Menampilkan.....	24
III.8.4	Algoritma Simpan	24
III.9	Perancangan Antarmuka.....	25
Bab IV	Implementasi dan Pengujian	27
IV.1	Implementasi Kelas	27
IV.2	Implementasi Antarmuka.....	28
IV.3	Pengujian.....	29
IV.3.1	Skenario Pengujian	29
IV.3.2	Hasil Rincian Pengujian	30
Bab V	Kesimpulan dan Saran.....	33
V.1	Kesimpulan	33

V.2	Saran.....	33
	Daftar Pustaka	34

DAFTAR GAMBAR

Gambar 2. 1 Notasi simbol pada Entity Relationship.....	8
Gambar 2. 2 Contoh Skema Relasional	9
Gambar 3. 1 Deskripsi Sistem Aplikasi Pemodelan Basisdata.....	12
Gambar 3. 2 Use Case Pemodelan Basisdata.....	14
Gambar 3. 3 Robustness Diagram Use Case Menggambar ER-Diagram.....	15
Gambar 3. 4 Robustness Diagram Use Case Mengkonversi	16
Gambar 3. 5 Robustness Diagram Use Case Menampilkan ER-Diagram.....	16
Gambar 3. 6 Robustness Diagram Use Case Menyimpan Informasi Design Basisdata	16
Gambar 3. 7 Class Diagram	17
Gambar 3. 8 Sequence Diagram Menggambar ER-Diagram.....	18
Gambar 3. 9 Sequence Diagram Mengkonversi ER-Diagram ke Skema Relational	19
Gambar 3. 10 Sequence Diagram Menampilkan ER-Diagram.....	20
Gambar 3. 11 Sequence Diagram Menyimpan Informasi Design Basisdata.....	21
Gambar 3. 12 Gui Menggambar	25
Gambar 4. 1 Skenario Pengujian	29

DAFTAR TABEL

Tabel III. 1 Spesifikasi Batasan Sistem pada PC	12
Tabel III. 2 Deskripsi Gui Menggambar	26
Tabel IV. 1 Daftar Implementasi Kelas	27
Tabel IV. 2 Daftar implementasi kelas antarmuka	28
Tabel IV. 3 Hasil Rincian Pengujian	30

Bab I Pendahuluan

I.1 Latar Belakang

Pada saat sekarang ini, kesuksesan suatu organisasi bergantung pada kemampuannya menangkap data secara akurat dan tepat waktu, dalam hal pengoperasian, pengaturan data secara efektif, maupun penggunaan data untuk keperluan analisis.

Suatu organisasi tentunya akan membutuhkan basisdata sebagai media penyimpanan data, baik itu data produk, rekening, karyawan, maupun data yang lainnya. Kemampuan untuk mengatur atau mengolah sejumlah data, dan kecepatan untuk mencari informasi yang relevan, adalah aset yang sangat penting bagi suatu organisasi. Oleh karena itu basisdata memiliki peran yang sangat penting bagi suatu perusahaan. Orang yang bertanggung jawab pada pembuatan basisdata tersebut disebut sebagai database designer.

Sebuah sistem basisdata pada dasarnya merupakan komputerisasi sistem penyimpanan data. Basisdata itu sendiri dapat dianggap sebagai tempat penyimpanan atau wadah untuk koleksi komputerisasi data arsip.

Jika dilihat lebih lanjut ada beberapa langkah dalam merancang basisdata. Pertama, menganalisis hal-hal apa saja yang dibutuhkan basisdata (requirement analysis). Setelah menganalisis kebutuhan kemudian akan dibangun Entity Relationship Diagram (ERD). ERD adalah representasi grafis dari logika basisdata dengan menyertakan deskripsi detail 3 elemen, yakni seluruh entitas (entity), hubungan atau relasi (relationship) dan atribut (attribute). Selanjutnya ERD dikonversi menjadi skema relational. Skema relational merupakan kumpulan tabel-tabel yang saling berrelasi satu dengan lainnya.

Pada suatu ERD yang memiliki elemen yang kompleks akan membutuhkan waktu yang lama untuk dapat mengetahui berapa skema relational yang terbentuk. Skema relational yang didapat juga kurang akurat karena masih bergantung pada

kemampuan dari seorang basisdata designer dan tidak menutup kemungkinan terjadinya human error.

Untuk itu Tugas Akhir ini dilakukan untuk mengembangkan teknologi yang dapat mengkonversi ERD menjadi skema relasional secara cepat dan akurat. Sehingga akan sangat membantu bagi seorang basisdata designer dalam merancang sebuah basisdata.

I.2 Rumusan Masalah

Berdasarkan latar belakang diatas didapatkan rumusan masalah sebagai berikut:

1. Basisdata designer akan membutuhkan waktu yang lama untuk mengetahui skema relasional jika ERD yang memiliki banyak elemen yang kompleks.
2. Skema relasional yang didapat kurang akurat, masih memungkinkan terjadinya human error.

I.3 Batasan Masalah

Adapun batasan masalah yang ada pada Tugas Akhir ini adalah hanya bisa menangani konversi data ER Diagram menjadi Skema Relasional.

I.4 Tujuan

Tujuan dilakukannya Tugas Akhir ini adalah memudahkan basisdata designer dalam mengkonversi ERD menjadi skema relasional secara cepat dan akurat, sehingga mengurangi terjadinya human error.

Bab II Tinjauan Pustaka

II.1 Basisdata

II.1.1 Pengertian Basisdata

Ada beberapa definisi mengenai basis data menurut ahli atau ilmuwan, yaitu :

- George Tsu-der Chou : Basis data merupakan kumpulan informasi bermanfaat yang diorganisasikan ke dalam aturan yang khusus. Informasi ini adalah data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang (Abdul, 1999).
- Fabbri dan Schwab : Basis data yaitu sistem berkas terpadu yang dirancang terutama untuk meminimalkan duplikasi data.
- Ramez Elmasri : Basis data merupakan penyajian suatu aspek dari dunia nyata (real world). Basis data juga merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang terkumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut basis data. Dan Basis data perlu dirancang, dibangun dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa user dan beberapa aplikasi yang sesuai dengan kepentingan user.

Dari beberapa pengertian para ahli tersebut dapat di simpulkan bahwa pengertian Basis data sendiri adalah data yang mempunyai berbagai sumber data dalam pengumpulan data, bervariasi derajat interaksi kejadian dari dunia nyata, dirancang dan dibangun agar dapat digunakan oleh beberapa user untuk berbagai kepentingan.

II.1.2 Sistem Basisdata

Gabungan antara basis data dan perangkat lunak SMDB (Sistem Manajemen BasisData) termasuk di dalamnya program aplikasi yang dibuat dan bekerja dalam satu sistem disebut dengan Sistem Basis Data.

Data Base Management System (DBMS) / Sistem Manajemen Basis Data (SMB). Kelebihan dari DBMS antara lain adalah :

- **Kepraktisan** DBMS menyediakan media penyimpan permanen yang berukuran kecil namun banyak menyimpan data jika dibandingkan dengan menggunakan kertas.
- **Kecepatan** Komputer dapat mencari dan menampilkan informasi yang dibutuhkan dengan cepat.
- **Mengurangi Kejemuhan** Pekerjaan yang berulang-ulang dapat menimbulkan kebosanan bagi manusia, sedangkan mesin tidak merasakannya.
- *Update to date* Informasi yang tersedia selalu berubah dan akurat setiap.

Keuntungan-keuntungan dalam penggunaan DBMS antara lain adalah:

- Pemusatan kontrol data : Dengan satu DBMS di bawah kontrol satu orang atau kelompok dapat menjamin terpeliharanya standar kualitas data dan keamanan batas penggunaannya serta dapat menetralkan konflik yang terjadi dalam persyaratan data dan integritas data dapat terjaga.
- Pemakaian data bersama (*Shared Data*) : Informasi yang ada dalam basis data dapat digunakan lebih efektif dengan pemakaian beberapa user dengan kontrol data yang terjaga.
- Data yang bebas (*independent*) : Program aplikasi terpisah dengan data yang disimpan dalam komputer.
- Kemudahan dalam pembuatan program aplikasi baru.

- Pemakaian secara langsung : DBMS menyediakan antarmuka yang memudahkan pengguna dalam mengolah data.
- Data yang berlebihan dapat dikontrol : Data yang dimasukkan dapat terjadi kerangkapan (*redundant*), untuk itu DBMS berfungsi untuk menurunkan tingkat redundancy dan pengelolaan proses pembaruan data.
- Pandangan user (*user view*) : Ada kemungkinan basis data yang diakses adalah sama, maka DBMS mampu mengatur antarmuka yang berbeda dan disesuaikan dengan pemahaman tiap user terhadap basis data menurut kebutuhan.

Kelemahan-kelemahan DBMS antara lain :

- **Biaya** : kebutuhan untuk mendapatkan perangkat lunak dan perangkat keras yang tepat cukup mahal, termasuk biaya pemeliharaan dan sumber daya manusia yang mengelola basis data tersebut.
- **Sangat kompleks** : sistem basis data lebih kompleks dibandingkan dengan proses berkas, sehingga dapat mudah terjadinya kesalahan dan semakin sulit dalam pemeliharaan data.
- **Resiko data yang terpusat** : Data yang terpusat dalam satu lokasi dapat beresiko kehilangan data selama proses aplikasi.

II.2 Perancangan Basisdata

Secara umum perancangan basisdata terbagi atas 4 tahap, yaitu :

1. Analisa Kebutuhan (*Requirement Analysis*)

Sebelum melakukan desain basis data, harus mengetahui dan menganalisa keinginan pemakai terhadap suatu basis data sedetail mungkin. Proses ini disebut analisa kebutuhan. Untuk menentukan ,kebutuhan, pertama kali harus diidentifikasi bagian lain dari sistem informasi yang berhubungan dengan sistem basis data. Termasuk di

dalamnya pemakai dan aplikasi baru dan yang sudah ada, kemudian dianalisa.

Analisa kebutuhan dibawa ke user akhir atau pelanggan sistem basis data oleh tim ahli analis kebutuhan. Kebutuhan awal lebih informal, tidak lengkap, tidak konsisten dan sebagian tidak benar. Perlu pekerjaan yang lebih banyak untuk mentransformasi kebutuhan awal ke aplikasi yang lebih spesifik yang dapat digunakan oleh pengembangan sebagai langkah awal untuk menulis implementasi dan uji coba.

2. Model E-R (*Conceptual Design*)

Model E-R berisi detail deskripsi dari tipe-tipe entity, relasi dan batasan (*constraint*). Hasil rancangan berupa ER Diagram. ER model biasa digunakan untuk mengembangkan inisial dari desain basis data. ER model menyediakan suatu konsep yang bermanfaat yang dapat mengubah deskripsi informal dari apa yang diinginkan oleh user menjadi hal yang lebih detail, presisi, dan deskripsi detail tersebut dapat diimplementasikan ke dalam DBMS.

ER Diagram ialah suatu alat utama pemodelan data dan membantu menggambarkan data ke dalam entitas dan hubungan antar entitas. ERD berupa notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relatif kompleks. Dengan ERD kita dapat menguji model dengan mengabaikan proses yang harus dilakukan.

Elemen-elemen ERD :

1. Entitas (*entity*)

Entitas (*entity*) adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau di mana terdapat data. Entitas juga dapat diartikan sebagai ‘objek’ didunia nyata yang bisa dibedakan dengan ‘objek’ yang lain.

2. Relasi (*Relationship*)

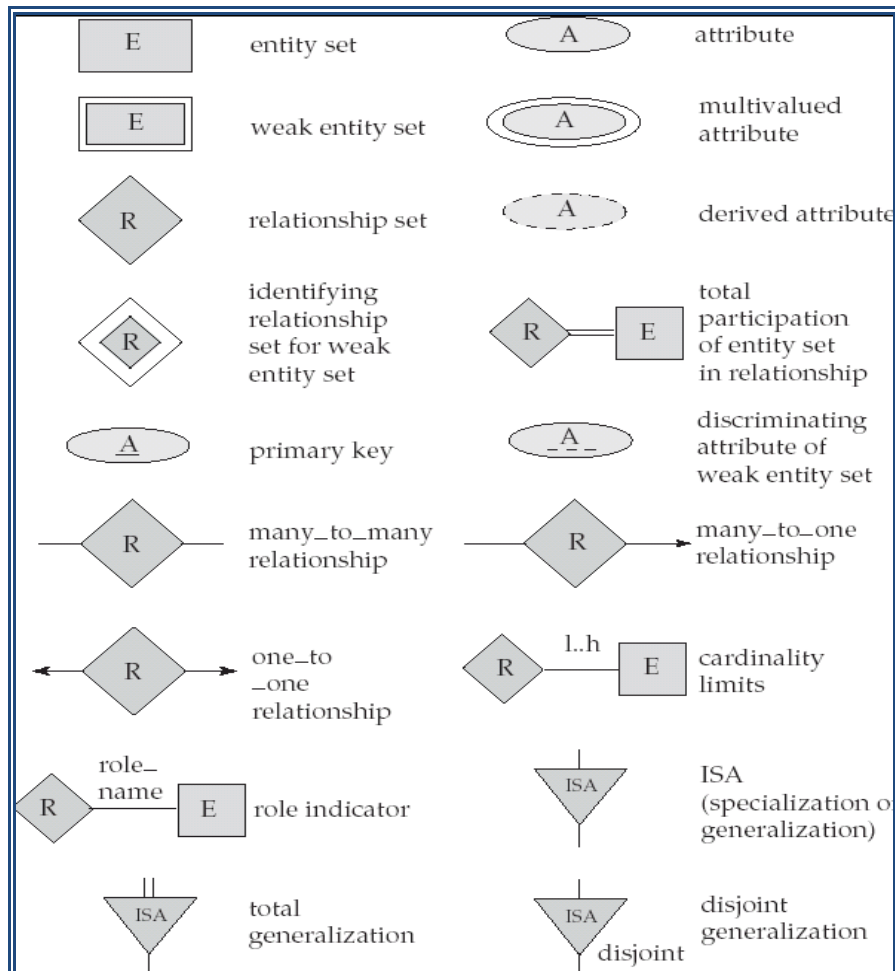
Relasi dapat diartikan sebagai hubungan yang terjadi diantara satu entitas dengan entitas yang lainnya.

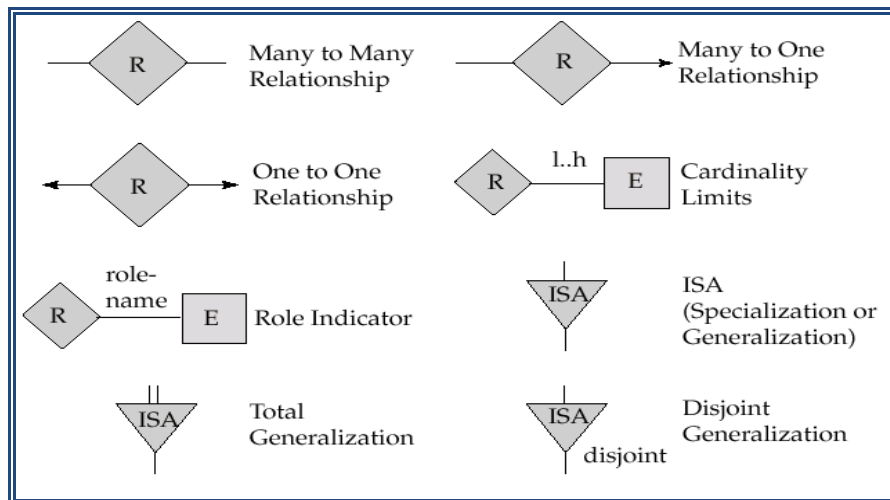
3. Atribut

Atribut merupakan gambaran karakteristik dari sebuah entitas atau himpunan entitas. Contoh : atribut untuk entitas mahasiswa adalah nim, nama, alamat, ipk, program studi, hobi, dan sebagainya.

Ada beberapa notasi yang digunakan untuk membuat ER Diagram. Misalnya Notasi Chen, Martin, El Masri, dan Korth Akan tetapi pada umumnya sama, perbedaanya ada pada pemilihan simbol yang digunakan. Notasi yang umum digunakan yaitu El Masri, karena mudah dibaca dan dimengerti.

Notasi simbol pada Entity Relationship :





Gambar 2. 1 Notasi simbol pada Entity Relationship

3. Skema Relasional (*Logical Design*)

Setelah ER dibuat, perlu dilakukan *mapping* ke skema relasi agar basisdata siap di implementasikan. Skema relasional adalah relasi yang memiliki nama, didefinisikan oleh himpunan pasangan atribut dan domainnya.

Aturan pemetaan :

1. Untuk entity E (*strong*), buat skema relasi, misal R, yang atributnya terdiri dari semua atribut atomik dan semua komponen dari atribut kompositnya E. Primary key dari R adalah atribut kunci dari E.
2. Untuk entity W (*weak*) yang tergantung pada entity E, buat skema relasi, misal R, yang atributnya terdiri dari semua atribut atomik dan semua komponen dari atribut komposit W. Tambahkan atribut kunci dari E ke R sebagai *foreign key*. Primary key dari R adalah gabungan dari *foreign key* dan atribut kuncinya W.
3. Untuk relasi biner 1:1 yang menghubungkan entity S dan T, pilih salah satu skema relasi, misal skema relasinya S, dan tambahkan atribut kuncinya T sebagai *foreign key* di skema relasinya S. Jika relasi yang menghubungkan S dan T memiliki atribut, tambahkan atribut tersebut ke S.

4. Untuk relasi biner 1:n yang menghubungkan entity S dan T, dimana S berada pada sisi n, tambahkan atribut kuncinya T sebagai *foreign key* di skema relasinya S. Jika relasi yang menghubungkan S dan T memiliki atribut, tambahkan atribut tersebut ke S.
5. Untuk relasi biner n:m yang menghubungkan S dan T, buat skema relasi baru, misal R. Tambahkan atribut kuncinya S dan T sebagai *foreign key* di R. Jika relasi yang menghubungkan S dan T memiliki atribut, tambahkan atribut tersebut ke R. *Primary key*nya R adalah gabungan kedua *foreign key* tersebut.
6. Jika entity E memiliki *multivalued* atribut A, maka buat skema relasi baru, misal R, yang atributnya terdiri dari atributnya A ditambah atribut kuncinya E sebagai *foreign key*. Semua atributnya R berfungsi sebagai primary key.

Contoh Skema Relasional :

```

PEGAWAI      (NOPEG, NAPEG, ALM1, KDPOS, TGLLAH, UMUR,
              SUPERVISOR-ID, NOBAG)

BAGIAN       (NOBAG, NABAG, LOKASI, MANAGER)

PROYEK      (NOPRO, NAPRO, NOBAG)

LOKPR       (NOPRO, LOKAPR)

PEKERJAAN   (NOPEG, NOPRO, JAM)

TANGGUNGAN  (NOPEG, NAMA, JNKELT, HUBUNGAN)

```

Gambar 2. 2 Contoh Skema Relasional

4. DBMS (*Physical Design*)

Merupakan tahap akhir, berisi definisi struktural internal storage, dll. Physical design merupakan suatu proses yang menghasilkan deskripsi implementasi basisdata pada penyimpanan sekunder. Menggambarkan struktur penyimpanan dan metode akses yang digunakan untuk mencapai akses yang efisien terhadap data. Dapat dikatakan juga desain fisik merupakan cara pembuatan menuju sistem DBMS tertentu.

II.3 Bahasa Java

II.3.1 Bahasa pemrograman berbasis Java

Java telah mengakomodasi hampir seluruh fitur penting bahasa – bahasa pemrograman yang ada semenjak perkembangan komputasi modern manusia :

- Dari SIMULA, bahasa pada tahun 65-an, bahasa yang paling mempengaruhi Java sekaligus C++. Dari bahasa ini diadopsi bentuk-bentuk dasar dari pemrograman berorientasi objek.
- Dari LISP – bahasa tahun 55-an. Diadopsi fasilitas *garbage collection*, serta kemampuan untuk meniru *generic list processing*, meski fasilitas ini jarang yang memanfaatkannya.
- Dari Algol – bahasa pada tahun 60-an, diambil struktur kendali yang dimilikinya.
- Dari C++, diadopsi sintaks, sebagian semantiks dan *exception handling*.
- Dari bahasa Ada, diambil *strongly type*, dan *exception handling*.
- Dari Objective C, diambil fasilitas *interface*.
- Dari bahasa SmallTalk, diambil pendekatan *single-root class hiérarchie*, dimana objek adalah satu kesatuan hirarki pewarisan.
- Dari bahasa Eiffel, fasilitas assertion yang mulai diterapkan di sebagian JDK 1.4

II.3.2 NetBeans IDE

NetBeans IDE merupakan lingkungan pengembangan, sebuah *tool, editor* untuk menulis bahasa pemrograman, mengkompilasi, mencari kesalahan dan membuat sebuah program. Netbeans IDE sendiri dikembangkan menggunakan bahasa Java. NetBeans menyediakan banyak fitur aplikatif

yang akan membuat pengembangan sebuah aplikasi menjadi lebih mudah, bahkan sangat lebih mudah. Hampir seluruh pelaksanaan *editing source code* dan pengaturan file konfigurasi dilakukan secara otomatis.

NetBeans merupakan sebuah proyek software *OpenSource*. Proyek Netbeans mulai diprakarsai oleh Perusahaan Sun Microsystems sejak bulan Juni 2000 dan terus berkembang hingga saat ini.

NetBeans IDE merupakan Integrated Development Environment (IDE) *open source* untuk *software developer*. IDE ini berjalan pada berbagai macam platform termasuk Windows, Linux, Solaris, dan MacOS. Selain itu, NetBeans IDE juga merupakan *software* yang mudah diinstal dan powerful. NetBeans IDE menyediakan kemampuan untuk *developer* dengan semua *tool* yang dibutuhkan untuk membuat aplikasi desktop, aplikasi enterprise, aplikasi web dan juga aplikasi *mobile* yang profesional serta lintas platform. Pengguna dapat menyediakan platform Java, C/C++, PHP, JavaScript ataupun Groovy untuk mengembangkan aplikasi yang mereka inginkan.

Adapun kelebihan Netbeans IDE antara lain :

- Mendukung berbagai bahasa pemrograman seperti Java, C/C++, PHP, JavaScript ataupun Groovy
- Berjalan pada multiplatform sistem operasi termasuk Windows, Linux, Mac OS, Solaris
- Berfungsi untuk pengembangan aplikasi mobile menggunakan bahasa Java
- Mendukung untuk pengembangan aplikasi web menggunakan PHP
- Mendukung permodelan perangkat lunak dengan UML (Unified Modeling Language)
- Terdapat banyak modul untuk mengembangkan lebih lanjut
- Merupakan produk free (baca: bebas) tanpa ada batasan penggunaannya
- Merupakan produk open source (baca: kode sumber terbuka)

Bab III Analisis dan Perancangan

Pada bab analisis dan perancangan ini akan menjelaskan tentang Batasan Sistem, Gambaran Kerja Sistem, *Use Case Diagram*, Analisis Kelas, *Interaction Sequence Diagram*, *Class Diagram*, Rancangan Kelas Rinci dan Algoritma.

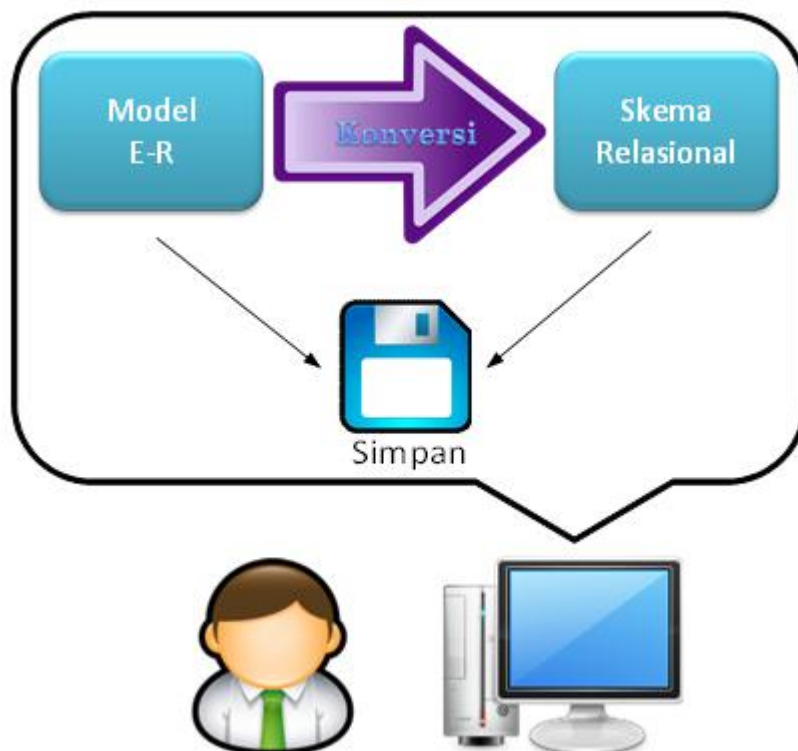
III.1 Batasan Sistem

Dalam pembuatan tugas akhir ini memiliki batasan sistem. Adapun spesifikasi pada sistem ini, yaitu :

Tabel III. 1 Spesifikasi Batasan Sistem pada PC

Deskripsi	:	Spesifikasi
Sistem Operasi	:	Windows 7
IDE	:	NetBeans
RAM	:	512 MB

III.2 Gambaran Kerja Sistem Aplikasi Pemodelan Basisdata



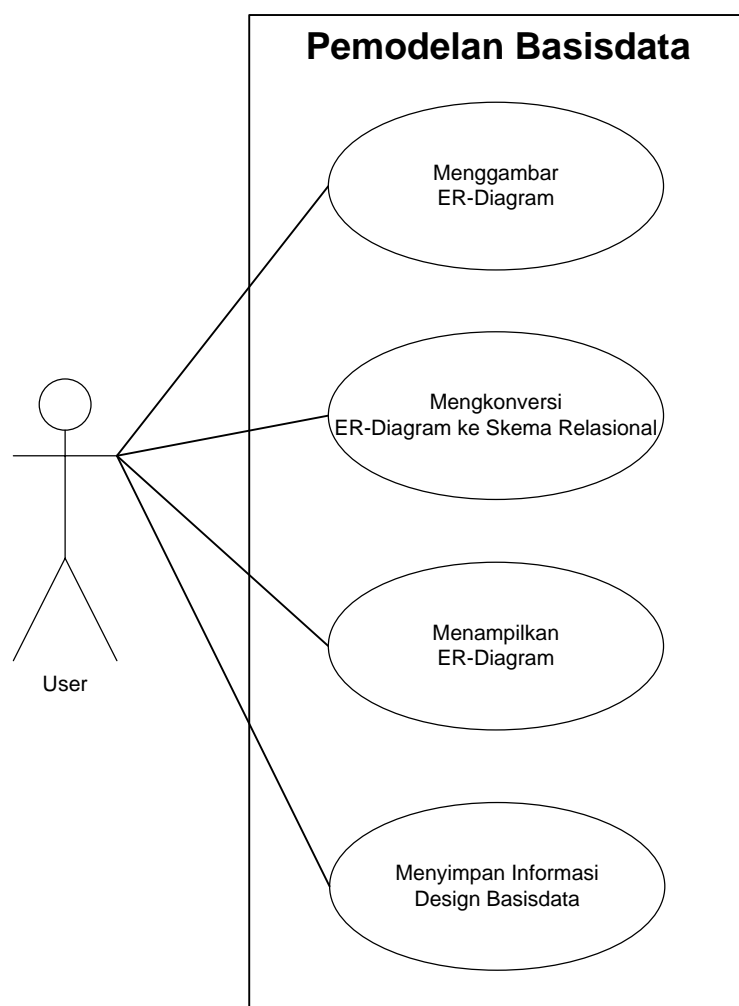
Gambar 3. 1 Deskripsi Sistem Aplikasi Pemodelan Basisdata

Deskripsi Sistem Aplikasi Pemodelan Basisdata pada Gambar 3.1 menjelaskan tentang proses kerja aplikasi Pemodelan Basisdata. Berikut merupakan spesifikasi sistem aplikasi pemodelan basisdata :

1. *User* dapat menggambar ER-Diagram
2. Sistem mengkonversi ER-Diagram menjadi Skema Relasional
3. Sistem menampilkan ER-Diagram dan Skema Relasional
4. Sistem dapat menyimpan ER-Diagram
5. Sistem dapat menyimpan Skema Relasional

III.3 Use Case Diagram Aplikasi Pemodelan Basisdata

Use Case Diagram aplikasi pemodelan basisdata ini digunakan untuk menggambarkan hubungan sejumlah *external actor* dengan *use case* yang terdapat dalam sistem aplikasi pemodelan basisdata. *Use Case Diagram* ini hanya menggambarkan keadaan lingkungan sistem yang dapat dilihat dari luar oleh aktor.



Gambar 3. 2 Use Case Pemodelan Basisdata

III.3.1 Skenario Use Case

Dari gambar 3.2 dapat dijelaskan bahwa *user* dapat menggambar ER-Diagram dan kemudian bisa mengkonversinya menjadi Skema Relasional dan *user* bisa menyimpan informasi design basisdata dan juga menampilkannya kembali.

III.3.1.1 Use Case Menggambar ER-Diagram

- Kondisi awal : Tersedia tools untuk menggambar ED-Diagram
Skenario : *User* menggunakan tool-tool yang tersedia untuk menggambar ER-Diagram
Kondisi akhir : ER-Diagram ditampilkan sesuai yang digambarkan

III.3.1.2 Use Case Mengkonversi ER-Diagram ke Skema Relasional

- Kondisi awal : Tersedia tombol konversi dan ER-Diagram sudah ada
Skenario : *User* mengklik tombol konversi kemudian system akan mengkonversi ER-Diagram ke Skema Relasional
Kondisi akhir : Didapat Skema Relasional sesuai dengan ER-Diagram yang telah digambar

III.3.1.3 Use Case Menampilkan ER-Diagram

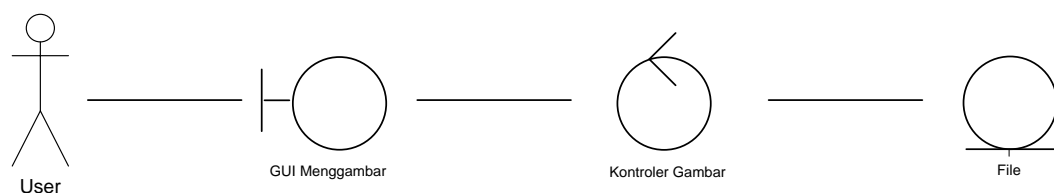
- Kondisi awal : Tersedia file informasi design basisdata
Skenario : *User* mencari file yang ingin ditampilkan kemudian system akan menampilkan ER-Diagram
Kondisi akhir : ER-Diagram ditampilkan

III.3.1.4 Use Case Menyimpan Informasi Design Basisdata

- Kondisi awal : ER-Diagram telah selesai digambar
Skenario : Setelah menggambar ER-Diagram *user* mengklik tombol simpan maka *system* akan menyimpan informasi design basisdata
Kondisi akhir : Informasi *design* basisdata tersimpan di media penyimpanan

III.4 Robustness Diagram

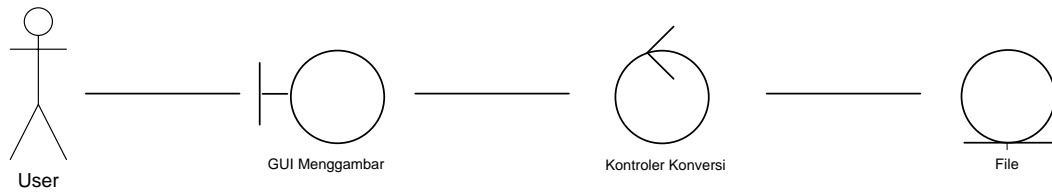
III.4.1 Robustness Diagram Use Case Menggambar ER-Diagram



Gambar 3. 3 Robustness Diagram Use Case Menggambar ER-Diagram

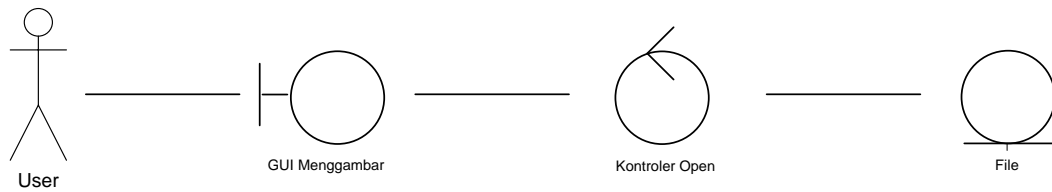
III.4.2 Robustness Diagram Use Case Mengkonversi ER-Diagram ke Skema

Relasional



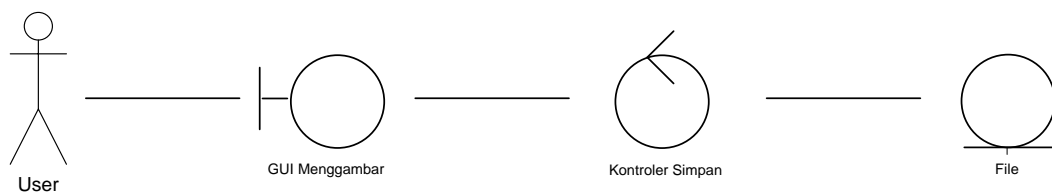
Gambar 3. 4 Robustness Diagram Use Case Mengkonversi ER-Diagram ke Skema Relasional

III.4.3 Robustness Diagram Use Case Menampilkan ER-Diagram



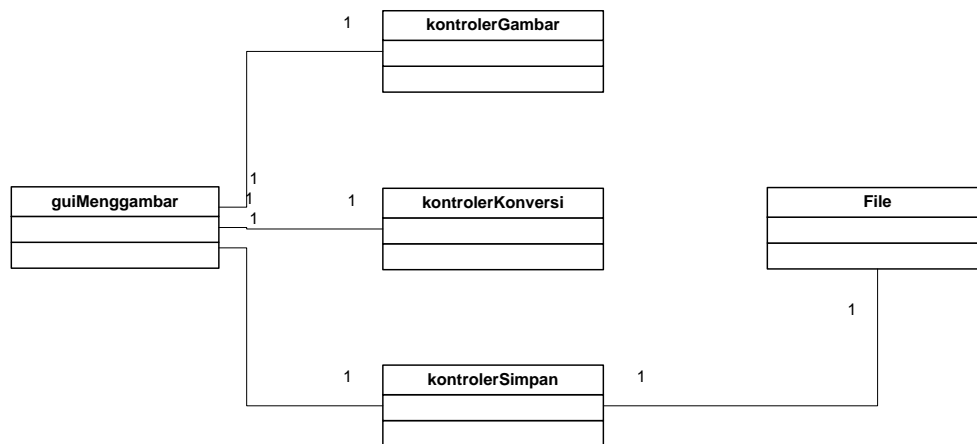
Gambar 3. 5 Robustness Diagram Use Case Menampilkan ER-Diagram

III.4.4 Robustness Diagram Use Case Menyimpan Informasi Design Basisdata



Gambar 3. 6 Robustness Diagram Use Case Menyimpan Informasi Design Basisdata

III.5 Class Diagram



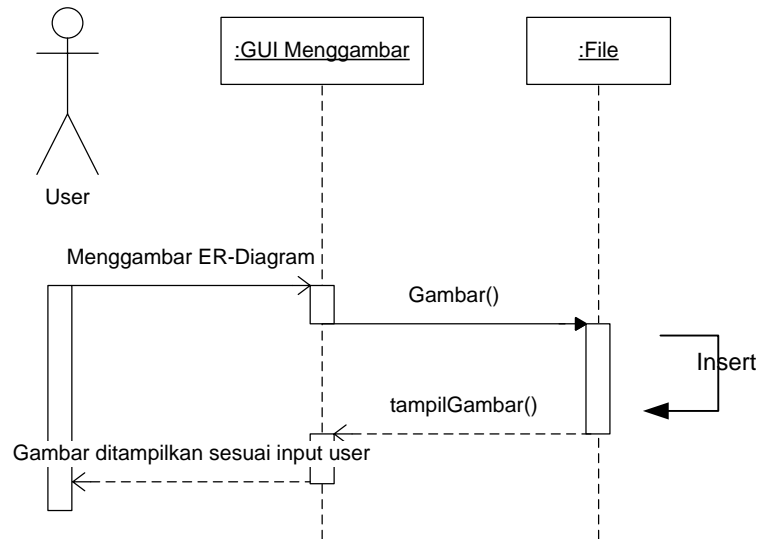
Gambar 3. 7 Class Diagram

Pada gambar 3.7 menjelaskan hubungan antar kelas dalam suatu diagram kelas. Diagram kelas diatas menandai kardinalitas dimana kelas 1 kontrolerKonversi bisa menangani 1 guiMenggambar, 1 kontrolerGambar bisa menangani 1 guiMenggambar, 1 kontrolerSimpan bisa menangani 1 guiMenggambar dan 1 entity bisa menangani 1 entitySimpan.

III.6 Interaction Sequence Diagram

Diagram ini menggambarkan urutan proses yang akan terjadi dalam sistem ini. Diagram ini juga menggambarkan method yang dijalankan oleh masing-masing kelas setiap proses yang terjadi pada sistem.

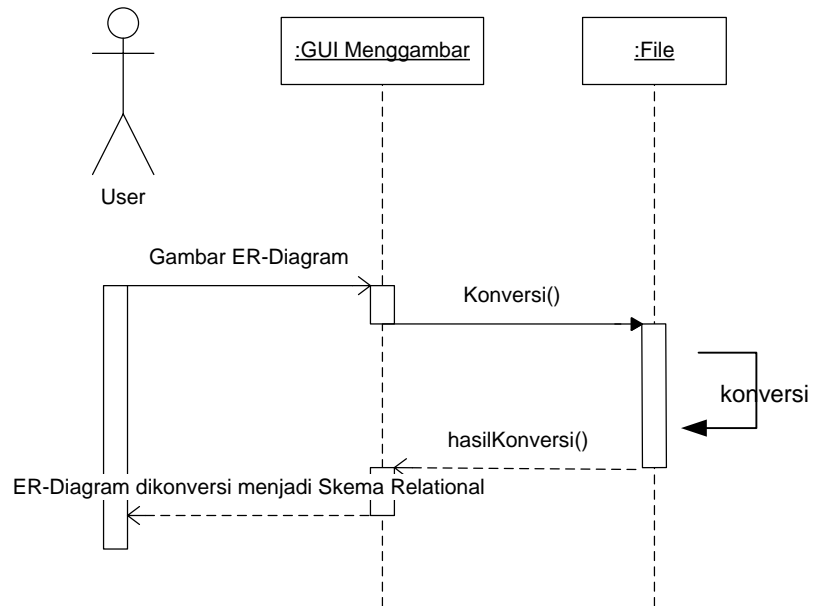
III.6.1 Menggambar ER-Diagram



Gambar 3. 8 Interaction Sequence Diagram Menggambar ER-Diagram

Pada gambar 3.8 *user* akan menggunakan *tools* menggambar yang tersedia pada layar, kemudian *system* akan menampilkan gambar sesuai dengan *tool* yang digunakan.

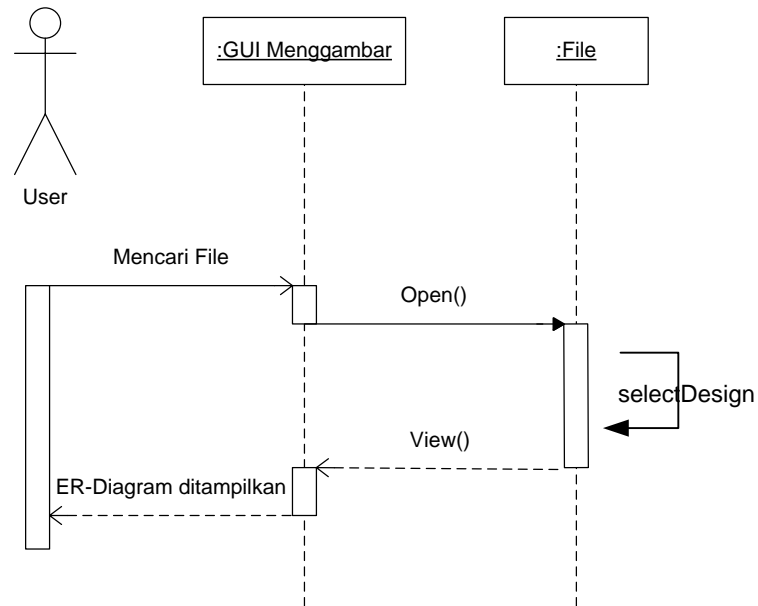
III.6.2 Mengkonversi ER-Diagram ke Skema Relasional



Gambar 3.9 Interaction Sequence Diagram Mengkonversi ER-Diagram ke Skema Relasional

Pada gambar 3.9 *user* menggambar ER-Diagram kemudian mengklik konversi, kemudian *system* akan mengkonversi ER-Diagram ke Skema Relasional dan akan ditampilkan kembali ke layar

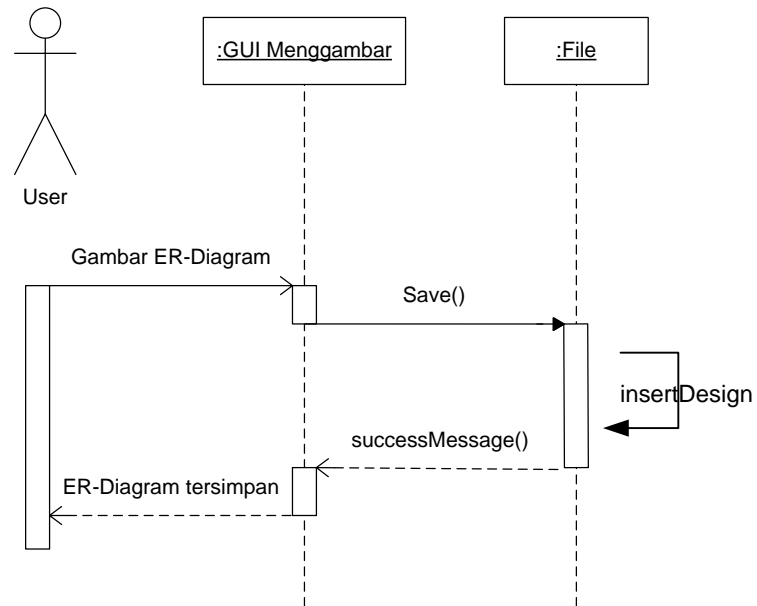
III.6.3 Menampilkan ER-Diagram



Gambar 3. 10 Interaction Sequence Diagram Menampilkan ER-Diagram

Pada gambar 3.10 *user* memilih *file* yang ingin ditampilkan, kemudian *system* akan menampilkan *file* yang dipilih.

III.6.4 Menyimpan Informasi *Design* Basisdata



Gambar 3. 31 Interaction Sequence Diagram Menyimpan Informasi *Design* Basisdata

Pada gambar 3.11 ER-Diagram telah digambar, kemudian *user* mengklik tombol *save* dan *system* akan melakukan penyimpanan *file* pada media penyimpanan yang kemudian ER-Diagram tersimpan.

III.7 Rancangan Kelas Rinci

III.7.1 Kelas guiMenggambar

guiMenggambar
-Convert:button
+btnKonversi()

III.7.2 Kelas kontrolerGambar

kontrolerGambar
-gambar:String
+inputGambar() +outputGambar()

III.7.3 Kelas kontrolerKonversi

kontrolerKonversi
-konversi:String
+inputKonversi() +outputKonversi()

III.7.4 Kelas kontrolerSimpan

kontrolerSimpan
-simpan:String -open:String
+save() +successMessage() +search() +view()

III.7.5 Kelas File

File
-design:String

+setDesign()

+getDesign()

III.8 Algoritma

III.8.1 Algoritma Gambar

Nama Operasi : Gambar

Algoritma :

```
{ menggambar ER-Diagram }
Initial state : worksheet dalam keadaan kosong
Final state : ER-Diagram telah dibuat pada worksheet

Algoritma:
If (tekan buttonEntitas)
    then
        Jalankan query = "insert into entitas values
('nama_entitas','worksheet)";
        Tampil gambar entitas;
else if(tekan buttonAtribut)
    then
        jalankan query = "insert into atribut values
('nama_atribut','entitas_reference','keterangan)";
        tampil gambar atribut;
else if(tekan buttonRelasi)
    then
        jalankan query = "insert into relasi values
('entitas_awal','entitas_akhir)";
        tampil gambar relasi;
```

III.8.2 Algoritma Konversi

Nama Operasi : Konversi

Algoritma :

```
{ mengkonversi ER-Diagram ke Skema Relational }
Initial state : terdapat gambar ER-Diagram pada worksheet
Final state : ER-Diagram dikonversi menjadi Skema Relational

Algoritma:
```

```

If (tekan buttonKonversi)
    Then
    If (one to one)
    then
        Jalankan query = "select nama_entitas, nama_atribut from
atribut where enetitas_references in (select nama_entitas from
entitas where worksheet_reference='')"
    Else if (one to many || many to one)
        Jalankan query = "select nama_relasi from relasi where
keterangan='one to many'";
        If (keterangan == one to many)
        Get Nama_Relasi;
        else if(keterangan == many to one)
        get Nama_Realsi;
    else(many to many);
    jalankan query = "select * from relasi";
    jalankan query = "select * from entitas";
    get nama_entitas;
    get nama_relasi;
    Tampil skema relasional;

```

III.8.3 Algoritma Menampilkan

Nama Operasi : Menampilkan

Algoritma :

```

{ menampilkan gambar ER-Diagram }
Initial state : worksheet dalam keadaan kosong
Final state : ER-Diagram ditampilkan pada worksheet

Algoritma:
If (tekan buttonOpen)
    then
        Tampil ER-diagram pada worksheet;

```

III.8.4 Algoritma Simpan

Nama Operasi : Simpan

Algoritma :

```

{ menyimpan gambar ER-Diagram }
Initial state : gambar ER-Diagram belum tersimpan

```

Final state : ER-Diagram telah disimpan

Algoritma:

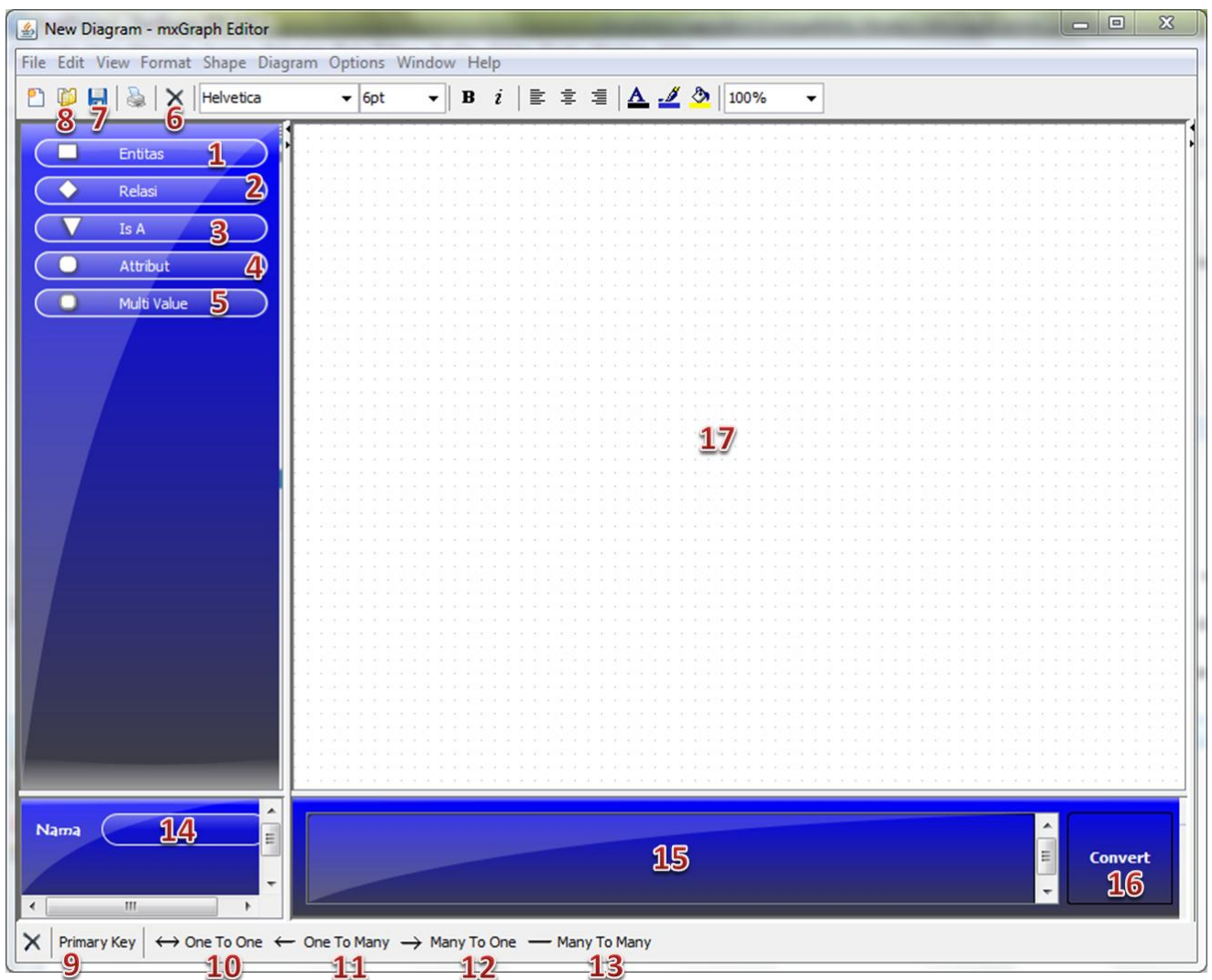
If (tekan buttonSave)

then

Tampil gambar entitas;

III.9 Perancangan Antarmuka

Aplikasi pengolah basisdata ini hanya memiliki 1 gui, yaitu gui menggambar.



Gambar 3. 12 Gui Menggambar

Tabel III. 2 Deskripsi Gui Menggambar

No	Id_Objek	Jenis	Nama	Keterangan
1	buttonEntitas	Button Glass	Entitas	Menambahkan gambar Entitas
2	buttonRelasi	Button Glass	Relasi	Menambahkan gambar Relasi
3	buttonIsa	Button Glass	IS A	Menambahkan spesialisasi IS A
4	buttonAtribut	Button Glass	Atribut	Menambahkan gambar Atribut
5	buttonMultivalue	Button Glass	Multi Value	Menambahkan gambar Atribut Multi Value
6	buttonHapus	Toolbar	Hapus	Menghapus objek
7	buttonSave	Toolbar	Save	Menyimpan gambar
8	buttonOpen	Toolbar	Open	Menampilkan gambar
9	buttonPK	JButton	PK	Menambahkan primary key
10	buttonOnetoone	Button Glass	One To One	Menambahkan kardinalitas One To One
11	buttonOnetomany	Button Glass	One To Many	Menambahkan kardinalitas One To Many
12	buttonManytoOne	Button Glass	One To Many	Menambahkan kardinalitas Many To One
13	buttonManytomany	Button Glass	Many To Many	Menambahkan kardinalitas Many To Many
14	textObject	Text Field	Object	Menampilkan keterangan object
15	textSkemaRelational	Text Field	Skema Relational	Menampilkan Skema Relational
16	buttonConvert	Button Glass	Convert	Mengkonversi gambar yang telah dibuat menjadi Skema Relational
17	worksheet		Worksheet	Worksheet untuk menggambar ER-Diagram

Bab IV Implementasi dan Pengujian

Untuk mengetahui keberhasilan dari program yang telah dirancang, maka perlu dilakukan pengujian terhadap aplikasi ini. Dalam bab ini akan dibahas mengenai proses pengujian yang dilakukan untuk mengetahui keakuratan, keefektifitas, efisiensi dan lain-lain dari aplikasi ini.

IV.1 Implementasi Kelas

Implementasi kelas dalam Aplikasi Pemodelan Basisdata dijelaskan dalam *table* 4.1 berikut:

Tabel IV. 1 Daftar Implementasi Kelas

No	Nama Kelas	Nama file fisik	Nama file Executable
1	Atribut	Atribut.java	Atribut.class
2	Entitas	Entitas.java	Entitas.class
3	Relasi	Relasi.java	Relasi.class
4	ISA	ISA.java	ISA.class
5	ImpleAtribut	ImpleAtribut.java	ImpleAtribut.class
6	ImpleEntitas	ImpleEntitas.java	ImpleEntitas.class
7	ImpleRelasi	ImpleRelasi.java	ImpleRelasi.class
8	ImpleISA	ImpleISA.java	ImpleISA.class
9	InterAtribut	InterAtribut.java	InterAtribut.class
10	InterEntitas	InterEntitas.java	InterEntitas.class
11	InterRelasi	InterRelasi.java	InterRelasi.class
12	InterISA	InterISA.java	InterISA.class
13	DatabaseUtilities	DatabaseUtilities.java	DatabaseUtilities.class
14	BasicEditor	BasicEditor.java	BasicEditor.class
15	DefaultFileFilter	DefaultFileFilter.java	DefaultFileFilter.class
16	EditorActions	EditorActions.java	EditorActions.class
17	EditorMenuBar	EditorMenuBar.java	EditorMenuBar.class
18	EditorToolBar	EditorToolBar.java	EditorToolBar.class
19	ERSchema	ERSchema.java	ERSchema.class

Setelah melakukan implementasi kelas, ternyata dibutuhkan kelas sebanyak 19 kelas. Sebagai antarmuka dibutuhkan kelas ERSchema, EditorToolBar, EditorMenuBar yaitu berfungsi untuk *form* antarmuka, *toolbar* serta menubar pada antarmuka aplikasi.

Pada kelas Atribut, Entitas, Relasi dan ISA berfungsi untuk menampung *variable* kelas Atribut, Entitas, Relasi dan ISA, sedangkan kelas ImpleAtribut, ImpleEntitas, ImpleRelasi, ImpleISA berfungsi untuk menjalankan query dari variable kelas Atribut, Entitas, Relasi, ISA. Pada kelas InterAtribut, InterEntitas, InterRelasi, InterISA berfungsi untuk memanggil kelas ImpleAtribut, ImpleEntitas, ImpleRelasi, ImpleISA. Kelas DatabaseUtilities berfungsi sebagai kelas koneksi ke database.

Kelas BasisEditor berfungsi untuk melakukan konversi ER-Diagram menjadi Skema Relational, kelas EditorActions berfungsi untuk menyimpan informasi *design* basisdata dan juga berfungsi untuk menampilkan file yang sudah disimpan, kelas DefaultFileFilter berfungsi sebagai kelas untuk penyimpanan data.

IV.2 Implementasi Antarmuka

Implementasi antarmuka dalam Aplikasi Pemodelan Basisdata dijelaskan dalam tabel 4.2 berikut:

Tabel IV. 2 Daftar implementasi kelas antarmuka

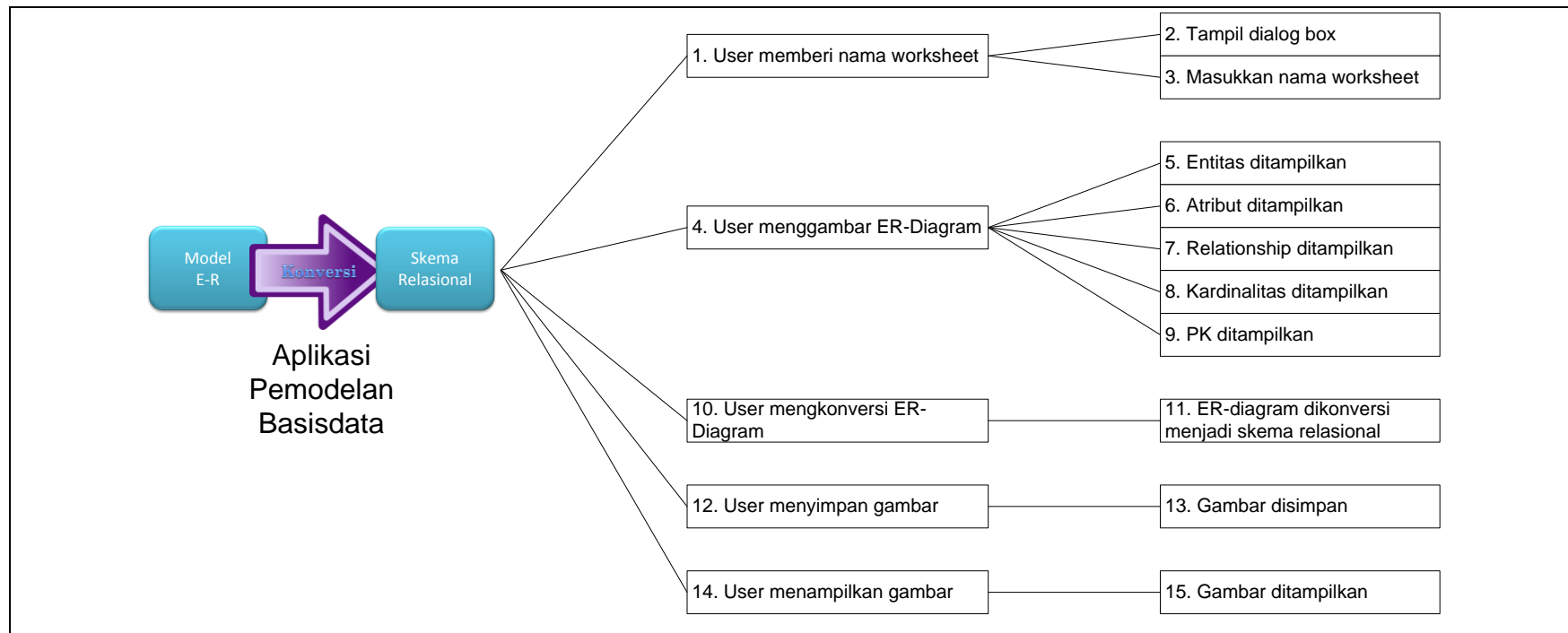
No	Nama Kelas	Nama file fisik	Nama file Executable
1	Menggambar	ERSchema.java	ERSchema.class

Berdasarkan perancangan dan implementasi yang dilakukan, aplikasi hanya membutuhkan satu antarmuka, karena semua kegiatan dalam menggambar ER-Diagram dan konversi menjadi skema relational dapat dilakukan dalam satu antarmuka saja.

IV.3 Pengujian

IV.3.1 Skenario Pengujian

Skenario pengujian dilakukan untuk menguji aplikasi yang telah dibuat sesuai dengan yang telah diskenariokan. Skenario pengujian implementasi aplikasi Pemodelan Basisdata dapat dilihat pada Gambar 4.1



Gambar 4. 1 Skenario Pengujian

IV.3.2 Hasil Rincian Pengujian

Tabel IV. 3 Hasil Rincian Pengujian

No	Kelas	Fungsi	Use Case	Skenario	Data Uji	Target	Verifikasi
1	guiMenggambar	Insert		User memberi nama worksheet	Beri worksheet dengan nama Coba	Tampil worksheet dengan nama Coba	✓
2	KontrolerGambar	Insert	Menggambar ER-Diagram	User menggambar ER-Diagram	Menggambar entitas Dosen	Tampil entitas Dosen	✓
					Menggambar entitas Mahasiswa	Tampil entitas Mahasiswa	✓
					Menggambar atribut id_Dosen dan nama_Dosen untuk entitas Dosen	Tampil atribut id_Dosen dan nama_Dosen pada entitas Dosen	✓
					Menggambar atribut id_Mahasiswa dan nama_Mahasiswa untuk entitas Mahasiswa	Tampil atribut id_Mahasiswa dan nama_Mahasiswa pada entitas Mahasiswa	✓
					Menggambar relationship Ajar	Tampil relationship Ajar	✓
					Menggambar kardinalitas Many to Many	Tampil kardinalitas Many to Many	✓

					Menggambar kardinalitas One to One	Tampil kardinalitas One to One	✓
					Menggambar kardinalitas One untuk entitas Dosen dan kardinalitas Many untuk entitas Mahasiswa	Tampil kardinalitas One pada entitas Dosen dan kardinalitas Many pada entitas Mahasiswa	✓
					Menggambar PK pada atribut id_Dosen pada entitas Dosen dan atribut id_Mahasiswa pada entitas Mahasiswa	Tampil PK pada atribut id_Dosen pada entitas Dosen dan atribut id_Mahasiswa pada entitas Mahasiswa	✓
3	kontrolerKonversi	Konversi	Mengkonversi ER-Diagram ke Skema Relational	User mengkonversi ER-Diagram	Konversi gambar ER-Diagram dengan kardinalitas Many to Many	Tampil Skema Relasional: -Dosen(<u>id_Dosen</u> , nama_Dosen) -Mahasiswa(<u>id_Mahasiswa</u> , nama_Mahasiswa) -Ajar(<u>id_Dosen</u> , <u>id_Mahasiswa</u>)	✓
					Konversi gambar ER-Diagram dengan kardinalitas One to One	Tampil Skema Relasional: -Dosen(<u>id_Dosen</u> , nama_Dosen) -Mahasiswa(<u>id_Mahasiswa</u> ,	✓

						nama_Mahasiswa)	
					Konversi gambar ER-Diagram dengan kardinalitas One to Many	Tampil Skema Relasional: -Dosen(<u>id_Dosen</u> , nama_Dosen) -Mahasiswa(<u>id_Mahasiswa</u> , nama_Mahasiswa, <u>id_Dosen</u>)	✓
4	kontrolerSimpan	Insert	Menyimpan informasi basisdata	User menyimpan gambar	Simpan gambar ER-Diagram dengan nama Berhasil	Gambar ER-Diagram disimpan dengan nama Berhasil	✓
5	kontrolerSimpan	Select	Menampilkan ER-Diagram	User menampilkan gambar	Membuka file ER-Diagram dengan nama Berhasil	Gambar ER-Diagram dengan nama Berhasil dapat ditampilkan	✓

Bab V Kesimpulan dan Saran

V.1 Kesimpulan

Setelah melakukan pengembangan terhadap aplikasi Pemodelan Basisdata berdasarkan tujuan dan implementasi yang telah dilakukan maka didapat kesimpulan:

1. Aplikasi dapat menggambar suatu ER-Diagram yang dapat dikonversi menjadi Skema Relational.
2. Berdasarkan implementasi yang dilakukan dibutuhkan suatu database untuk menyimpan data ER-Diagram dan skema relational sehingga dapat dilakukan penyimpanan data.
3. Terdapat perubahan antarmuka, karena terdapat penambahan fitur pendukung, seperti toolbar dan menubar.

V.2 Saran

Sebagai langkah pengembangan aplikasi ini pada masa yang akan datang, maka saran yang dapat diberikan adalah:

1. Aplikasi Pemodelan Basisdata ini dapat dikembangkan dengan menambahkan output berupa kalimat SQL sehingga dapat langsung digunakan pada DBMS.
2. Aplikasi ini dapat melakukan konversi gambar ER-Diagram dari media lain (Microsoft Visio).

Daftar Pustaka

Berisi daftar Tugas Akhir dan buku referensi yang diacu pada bagian Tinjauan Pustaka.

1. Date, C. J.(2004). *Pengenalan Sistem Basisdata edisi ketujuh jilid 1*. Jakarta : PT. Indeks Group Gramedia diakses tanggal 11/10/2011
2. [http://blog.unsri.ac.id/userfiles/59081003017_\(6\).doc](http://blog.unsri.ac.id/userfiles/59081003017_(6).doc) diakses pada 20 oktober 2011
3. <http://achsan.staff.gunadarma.ac.id/Downloads/files/4184/Struktur+Data.pdf> diakses pada 13 oktober 2011
4. <http://www.scribd.com/doc/25209413/Entity-Relationship-Diagram> diakses pada 13 oktober 2011
5. http://courseware.politekniktelkom.ac.id/BUKU_KA/Semester%202/IS143%20Perancangan%20Basis%20Data/IS143%20Perancangan%20Basis%20Data.docx diakses pada 13 oktober 2011
6. http://imam_muiz.staff.gunadarma.ac.id/Downloads/files/6535/BASIS+DATA.pdf diakses pada 13 oktober 2011
7. <http://www.jrgaph.com/> diakses pada 13 oktober 2011
8. http://202.91.15.14/upload/files/8511_Appendix_2_-_Desain_Basis_Data.pdf diakses pada 13 oktober 2011
9. http://staff.uny.ac.id/sites/default/files/BAB_IV%20basis%20data.doc diakses pada 10 oktober 2011