

BAB II

TINJAUAN PUSTAKA

II.1 Sistem Informasi Geografis

Menurut Depdiknas di dalam website resminya (www.sig.depdiknas.go.id, 07/02/13 13:46) *Geographic Information System (GIS)* atau *Sistem Informasi Geografis (SIG)* diartikan sebagai sistem informasi yang digunakan untuk memasukkan, menyimpan, memanggil kembali, mengolah, menganalisis dan menghasilkan data bereferensi geografis atau data geospasial, untuk mendukung pengambilan keputusan dalam perencanaan dan pengelolaan penggunaan lahan, sumber daya alam, lingkungan transportasi, fasilitas kota, dan pelayanan umum lainnya.

Pendekatan-pendekatan kelokasian atau lebih dikenal dengan istilah pendekatan keruangan/spasial sangat penting di dalam melakukan analisis-analisis fenomena yang terjadi di bumi ini, baik itu yang sifatnya fisik maupun yang bersifat sosial kemasyarakatan seperti ekonomi, politik, lingkungan, budaya, dsb. Karena jika fenomena itu bisa ditangkap informasinya secara utuh berikut lokasi dan polanya, hal tersebut bisa membantu dalam menyelesaikan atau mencari solusi dari permasalahan terkait muka bumi.

Tentunya untuk mendapatkan informasi yang utuh tersebut diperlukan satu metode yang tepat dan akurat. Antara lain metode itu adalah pemetaan, yang bisa menggambarkan obyek-obyek di muka bumi ini ke dalam media yang lebih kecil sehingga lebih mudah dipahami. Namun pemetaan saja belumlah cukup, data-data

atau informasi lainnya pun (yang terkait) perlu digambarkan atau tersaji secara lokasional pula. Sehingga peta dan data (*database*) nya perlu dihubungkan dengan alat yang tepat. Alat tersebut adalah GIS (*Geographical Information System*) yang dalam bahasa Indonesia berarti SIG (*Sistem Informasi Geografis*). GIS ini merupakan teknik berbasis komputer untuk memasukan, mengolah, dan menganalisis data-data obyek permukaan bumi dalam bentuk grafis, koordinat, dan database, di mana hasilnya bisa menggambarkan sebuah fenomena keruangan (spasia) yang bisa digunakan sebagai basis informasi untuk pengambilan keputusan di berbagai bidang.

Dalam SIG terdapat berbagai peran dari berbagai unsur, baik manusia sebagai ahli dan sekaligus operator, perangkat alat (lunak/keras) maupun objek permasalahan. SIG adalah sebuah rangkaian sistem yang memanfaatkan teknologi digital untuk melakukan analisis spasial. Sistem ini memanfaatkan perangkat keras dan lunak komputer untuk melakukan pengolahan data seperti:

1. Perolehan dan verifikasi
2. Kompilasi
3. Penyimpanan
4. Pembaruan dan perubahan
5. Manajemen dan pertukaran
6. Manipulasi
7. Penyajian
8. Analisis

Pemanfaatan SIG secara terpadu dalam sistem pengolahan citra digital adalah untuk memperbaiki hasil klasifikasi. Dengan demikian, peranan teknologi SIG dapat diterapkan pada operasionalisasi penginderaan jauh satelit.

Mengingat sumber data sebagian besar berasal dari data penginderaan jauh baik satelit maupun *terrestrial*, maka teknologi SIG erat kaitannya dengan teknologi penginderaan jauh. Namun demikian, penginderaan jauh bukanlah satu-satunya ilmu pendukung bagi sistem ini. Sumber data lain berasal dari hasil *surveyterrestrial* (uji lapangan) dan data-data sekunder lain seperti sensus, catatan, dan laporan yang terpercaya.

II.2 GeoTools

Menurut pengembang *GeoTools* di dalam website nya (www.geotools.org/about.html, 07/02/12 15:47). *GeoTools* adalah sebuah pustaka kode *Java* bersifat terbuka yang menyediakan metode standar yang sesuai untuk memanipulasi data geospasial, Misalnya untuk menerapkan *Sistem Informasi Geografis* (SIG). Pustaka kode *GeoTools* mengimplementasikan *Open Geospatial Consortium* (OGC) dengan spesifikasi yang mereka kembangkan.

GeoTools merupakan penemu proyek *Geospastial* yang *Open-source* dan memegang hak cipta pada pustaka kode dengan bantuan dari komunitas. Pustaka *GeoTools* dibuat dibawah lisensi LGPL yang dimana *source-code GeoTools* dapat diakses oleh public secara bebas melalui repositori subversi.

II.3 UML (Unified Modeling Language)

UML (*Unified Modeling Language*) adalah suatu alat Bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek (Munawar, 2005).

Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Meskipun UML sudah banyak menyediakan diagram yang bisa membantu mendefinisikan suatu aplikasi, tidak berarti bahwa semua diagram tersebut akan bisa menjawab persoalan yang ada. Adapun tipe diagram UML yang ada seperti pada Tabel II.1 :

Tabel II.1 Tipe Diagram UML

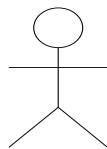
Diagram	Tujuan	Keterangan
Activity	Prilaku prosedural dan parallel	Sudah ada di UML 1
Class	Class, fitur dan relasinya	Sudah ada di UML 1
Communication	Interaksi diantara objek. Lebih menekankan kepada link	Di UML 1 disebut collaboration
Component	Struktur dan koneksi dari komponen	Sudah ada di UML 1
Composite Structure	Dekomposisi sebuah class saat runtime	Baru untuk UML 2
Deployment	Penyebaran/instalasi ke klien	Sudah ada di UML 1
Interaction Overview	Gabungan dari activity dan sequence diagram	Baru untuk UML 1
Object	Contoh konfigurasi instance	Tidak resmi ada di UML 1
Package	Struktur hierarki saat kompilasi	Tidak resmi ada di UML 1
Sequence	Interaksi antara objek. Lebih menekankan pada urutan.	Sudah ada di UML 1
State Machine	Bagaimana event mengubah sebuah objek	Sudah ada di UML 1
Timing	Interaksi antar objek. Lebih menekankan pada waktu	Sudah ada di UML 1
Use Case	Bagaimana user berinteraksi dengan sebuah sistem	Sudah ada di UML 1

Sumber : " Pemodelan Visual dengan UML (Munawar ; 2009 : 23) "

II.3.1 Notasi Dasar UML

II.3.1.1 Actor

Actor adalah *abstraction* dari orang dan *system* yang lain yang mengaktifkan fungsi dari target *system*. Orang atau *system* bisa muncul dalam beberapa peran. Perlu dicatat bahwa *actor* berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Berikut notasi *actor* dalam UML :

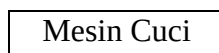


Gambar II.1 Notasi Actor pada UML

Sumber : "Pemodelan Visual dengan UML (Munawar ; 2009 : 64)"

II.3.1.2 Class

Class dalam notasi UML digambarkan dengan kotak. Nama class menggunakan huruf besar di awal kalimatnya dan diletakkan di atas kotak. Bila class mempunyai nama yang terdiri dari 2 suku kata atau lebih, maka semua suku kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar. Berikut notasi class dalam UML :



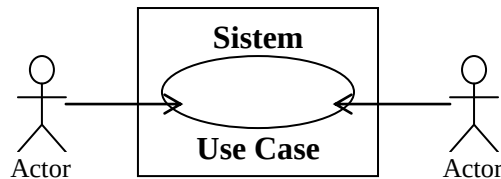
Gambar II.2 Notasi Class di UML

Sumber : "Pemodelan Visual dengan UML (Munawar ; 2009 : 35)"

II.3.1.3 Usecase

Usecase adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu *system* dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud

menggunakan sebuah *system*. Karena *system* pengembangan tradisional sering ceroboh dalam melakukan analisis, akibatnya pengguna seringkali susah menjawabnya tatkala dimintai masukan tentang sesuatu. Notasi *use case* dapat dilihat pada gambar II.3 :



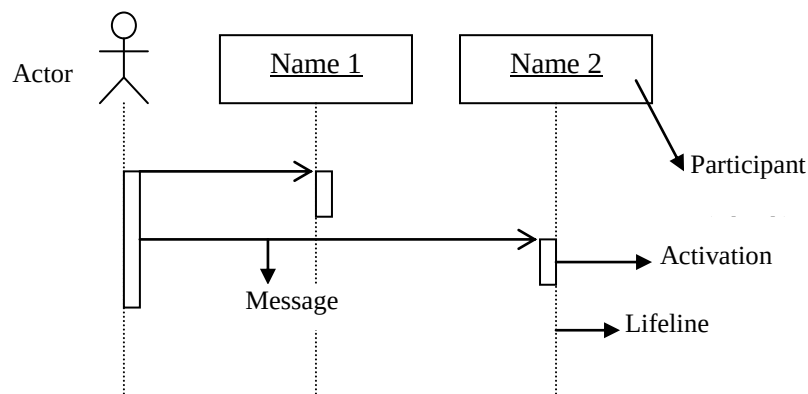
Gambar II.3 : Notasi Use Case pada UML

Sumber : "Pemodelan Visual dengan UML (Munawar ; 2009 : 64)"

II.3.1.4 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini dalam *use case*.

Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. Berikut Contoh *sequence diagram* :



Gambar II.4 : Simbol-simbol yang ada pada sequence diagram

Sumber : " Pemodelan visual dengan UML (Munawar ; 2009 : 89)"

II.3.1.5 Activity Diagram

Activity diagram adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activitydiagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa.

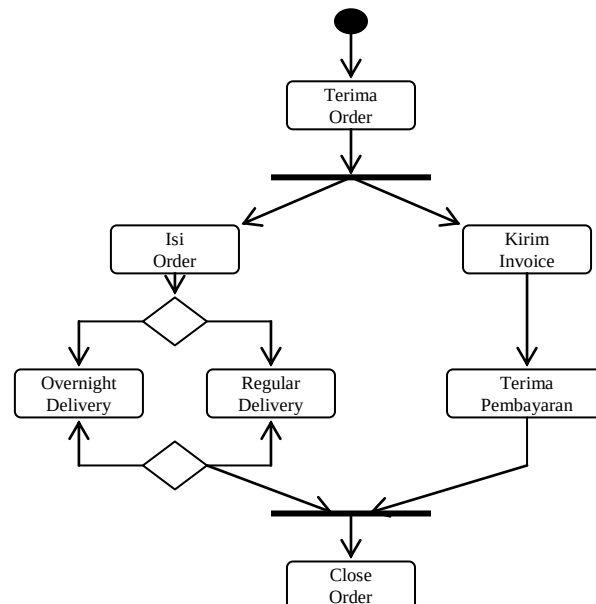
Berikut adalah simbol-simbol yang sering digunakan pada saat pembuatan *activity diagram*.

Tabel II.2 Simbol-simbol yang sering dipakai pada activity diagram

Simbol	Keterangan
●	Titik awal
●	Titik akhir
▭	Activity
◇	Pilihan untuk pengambilan keputusan
—	Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
⌋	Rake; menunjukkan adanya dekomposisi
⌋	Tanda waktu
▭	Tanda pengiriman
⌋	Tanda penerimaan
⊗	Aliran akhir (Flow Final)

Sumber : " Pemodelan visual dengan UML (Munawar ; 2009 : 109) "

Adapun contoh dari *Activity Diagram* dapat di lihat pada Gambar II.5 :



Gambar II.5 : Contoh Activity Diagram Sederhana

Sumber : " Pemodelan visual dengan UML (Munawar ; 2009 : 111)"

II.4 Pengertian Database

Database merupakan komponen terpenting dalam pembangunan SI, karena menjadi tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat dieksplorasi untuk menyusun informasi-informasi dalam berbagai bentuk. *Database* merupakan himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau di eksplorasi secara cepat dan mudah untuk menghasilkan informasi.

II.4.1 Hierarki Data Dalam Database

Data dalam sebuah *database* disusun berdasarkan sistem hierarki yang unik, yaitu:

1. **Database**, merupakan kumpulan file yang saling terkait satu sama lain, misalnya file data induk karyawan, file jabatan, file penggajian, dan lain sebagainya. Kumpulan file yang tidak saling terkait satu sama lain tidak dapat disebut *database*, misalnya file data induk karyawan, file tamu undangan perkawinan, file barang pasar swalayan.
2. **File**, yaitu kumpulan dari *record* yang saling terkait dan memiliki format *field* yang sama dan sejenis.
3. **Record**, yaitu kumpulan *field* yang menggambarkan suatu unit data individu tertentu.
4. **Field**, yaitu atribut dari *record* yang menunjukkan suatu item dari data, seperti nama, alamat, dan lain sebagainya.
5. **Byte**, yaitu atribut dari *field* yang berupa huruf yang membentuk nilai dari sebuah *field*. Huruf tersebut dapat berupa numerik maupun abjad atau karakter khusus.
6. **Bit**, yaitu bagian terkecil dari data secara keseluruhan, yaitu berupa karakter ASCII nol atau satu yang merupakan komponen pembentuk *byte*.

II.5 Bahasa Pemrograman Java

Menurut Hariyanto (2007:1), dalam *Esensi-Esensi Bahasa Java*, bahasa pemrograman *Java* merupakan karya *Suns Microsystem Inc.* Rilis resmi level *beta* dilakukan pada November 1995. Dua bulan berikutnya *Netscape* menjadi

perusahaan pertama yang memperoleh lisensi bahasa *Java* dari *Sun*. *Java* adalah bahasa pemrograman berorientasi objek yang berukuran kecil, sederhana dan aman, diinterpretasi atau dioptimalisasi secara dinamis, ber-*bytecode*, arsitektur yang netral, mempunyai *garbage-collector*, *multithreading*, mempunyai mekanisme penanganan pengecualian, berbasis tipe untuk penulisan program mudah diperluas dinamis serta telah diperuntukan sistem tersebar.

Pada awal pembuatannya, *Java* dinamakan *Oak*, kemudian nama *Oak* dinilai kurang menjual sehingga pada Januari 1995 nama *Oak* diubah menjadi *Java*. Sebagai bahasa yang bersifat terbuka, *Java* didukung oleh banyak *programmer* dari seluruh dunia yang memberikan kontribusinya untuk mengembangkan bahasa *Java*.

II.5.1 MySQL (phpMyAdmin)

Pengertian *MySQL* Menurut DeniSutaji (2012 : 40), *MySQL* adalah *DMBS* yang didistribusikan secara gratis dibawah lisensi dari *General Public Lisence* (GPL), dimana setiap orang bebas untuk menggunakannya tetapi tidak boleh untuk dijadikan program induk turunan yang bersifat tertutup atau komersial.

MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam basis data sejak lama, yaitu *SQL*. *SQL* adalah sebuah konsep pengoperasian basis data terutama untuk proses seleksi, pemasukan, perubahan dan penghapusan data yang dimungkinkan dapat dikerjakan dengan mudah dan otomatis.

Berikut adalah keunggulan dari *MySQL*, diantaranya :

1. Portabilitas, yaitu dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS, Solaris dan Asigma.
2. Sumber Kode Terbuka yang berarti didistribusikan secara gratis dibawah lisensi dari GPL dimana ssetiap orang bebeas untuk menggunakannya tetapi tidak boleh menjadikan *MySQL* sebagai induk turunan yang bersifat tertutup atau komersial.
3. Multi Pengguna yang berarti dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan.
4. Kinerja tinggi yang mempunyai kecepatan dalam menangani *query*.
5. Tipe kolom yang memiliki tipe data sangat kompleks, seperti signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year dan enum.
6. Perintah dan fungsi, memiliki operator dan fungsi penuh yang mendukung *select* dan *where* dalam *query*.
7. Kemananan yang memiliki beberapa lapisan sekuritas seperti tingkat subnet mask, hostname, privilege user dengan sistem perijinan yang mendetail serta password yang terenkripsi.
8. Skalabilitas dan Batasan yang mampu menangani basis data dalam jumlah besar dengan jumlah filed lebih dari 50 juta, 60 ribu tabel, 5 milyar record serta batas indeks mencapai 32 buah per tabel.
9. Lokalisasi yang dapat mendeteksi pesan kesalahan (*error code*) pada klien dengan menggunakan lebih dari 20 bahasa.

10. Keterhubungan yang dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix Socket dan Named Pipes.
11. Antarmuka yang dimiliki terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API.
12. Klien dan Peralatan yang dilengkapi dengan berbagai peralatan yang dapat digunakan untuk administrasi basis data sekaligus dokumen petunjuk online.
13. Struktur tabel yang dimiliki lebih fleksibel dalam menangani *altertabel* dibandingkan dengan PostgreSQL dan Oracle.

II.5.2 Tipe Data MySQL

Tipe data yang dikenal oleh MySQL sampai dengan versi 5 keatas adalah String, Number, Boolean dan Date dan secara detail didistribusikan dalam bentuk tabel II.3 berikut :

Tabel II.3 Tipe data MySQL

Tipe Data	Keterangan
Number Type	
TINYINT	Integer yang sangat kecil
SMALLINT	Integer yang kecil
MEDIUMINT	Integer dengan ukuran menengah
INT	Integer dengan ukuran normal
INTEGER	Integer dengan ukuran normal
BIGINT	Integer yang sangat besar
FLOAT	Bilangan floating point yang kecil atau presisi tunggal
DOUBLE	Bilangan floating point dengan ukuran normal atau presisi ganda
DOUBLE PRECISION	Bilangan floating point dengan ukuran normal atau presisi ganda
REAL	Bilangan floating point dengan ukuran normal atau presisi ganda
DECIMAL	Bilangan unpacked floating point (bilangan disimpan dalam String) yang tidak dapat dijadikan unsigned
NUMERIC	Bilangan unpacked floating point (bilangan disimpan dalam String) yang tidak dapat dijadikan unsigned

Tipe Data	Keterangan
Date Type	
DATE	Data berupa tanggal dengan format YYYY-MM-DD
DATETIME	Kombinasi dari tanggal dan waktu dengan format YYYY-MM-DD HH:MM:SS
TIME	Data berupa waktu dengan format HH:MM:SS
YEAR [(2 4)]	Data dengan bentuk tahun dalam format 2 atau 4 digit
String Type	
CHAR	Sebuah String dengan panjang tetap, sisa jumlah karakter yang belum diisi akan digantikan oleh spasi akan tetapi spasi akan dibuang saat data dipanggil
VARCHAR	Sebuah String dengan panjang berupa variable
TINYTEXT	Sebuah BLOB atau teks dengan panjang karakter sampai dengan 255 karakter
TEXT	Sebuah BLOB atau teks dengan panjang karakter 65535 karakter
MEDIUMBLOB	Sebuah BLOB atau teks dengan panjang karakter maksimum 16777215 karakter
LONGBLOB	Sebuah BLOB atau teks dengan panjang karakter maksimum 4294967295 karakter
ENUM	Sebuah enumeration berupa obyek string yang hanya boleh memiliki salah satu nilai dari 'value1', 'value2', ..., NULL
SET	Sebuah set dengan obyek string yang dapat memiliki nilai nol atau lebih yang diambil dari 'value1', 'value2',

Sumber : “Sistem inventory mini market dengan php & jquery(Deni Sutaji ; 2012)”

II.5.3 SQL (Structured Query Language)

SQL adalah bahasa yang digunakan untuk berkomunikasi dengan basis data. Menurut *American National Standart Institute* (ANSI) bahasa ini adalah sebuah standar untuk *Relational Database Managemen System* (RBDMS). Terdapat dua bagian dalam SQL yaitu *Data Defenition Language* (DDL) dan *Data Manipulation Language* (DML).

DDL digunakan untuk mendefenisikan struktur basis data dan tabel, sintaksnya adalah CREATE, ALTER, DROP dan RENAME.

DML digunakan untuk melakukan manipulasi terhadap basis data, sintaksnya adalah SELECT, UPDATE DELETE, INSERT, TRUNCATE, dan lain-lain.

II.6 Kamus Data

Kamus data atau *data dictionary* (DD) disebut juga dengan istilah *systems data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perencanaan sistem (Jogiyanto: 2005: 725).

Tabel II.4 Notasi Kamus Data

Notasi	Arti
=	Terbentuk dari (is composed) atau terdiri dari (consist of) atau sama dengan (is equivalent of)
+	AND
[]	Salah satu dari (memilih salah satu dari elemen-elemen data di dalam kurung bracket ini)
	Sama dengan simbol []
M{ }M	Intensi (elemen data didalam kurung brace berinteraksi mulai minimum N kali dan maksimum M kali)
()	Optional (elemen data di dalam kurung parenthesis sifatnya optional, dapat ada dan dapat tidak ada)
*	Keterangan setelah tanda ini adalah komentar

Sumber : " Pengantar Sistem Informasi (Jogiyanto: 2005: 730)"

II.7 ERD (Entity Relationship Diagram)

II.7.1 Model-model Data

Struktur yang mendasari suatu basisdata adalah model data yang merupakan kumpulan alat-alat konseptual untuk mendeskripsikan data, relasi data, data

semantik, dan batasan konsistensi. Untuk mengilustrasikan konsep model data, berikut disajikan dua model data, yaitu *entity relationship model* dan *relational model*. Kedua model menyediakan cara mendeskripsikan rancangan basis data pada tingkatan logis.

II.7.2 Entity Relationship Model

Entity Relationship (ER) data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek. Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain. Sebagai contoh, masing-masing mahasiswa adalah entitas dan mata kuliah dapat pula dianggap sebagai entitas.

Entitas digambarkan dalam basis data dengan kumpulan atribut. Misalnya atribut nim, nama, alamat dan kota bisa menggambarkan data mahasiswa tertentu dalam suatu universitas. Atribut-atribut membentuk entitas mahasiswa. Demikian pula, atribut kodeMK, namaMK, dan SKS mendeskripsikan entitas mata kuliah.

Atribut NIM digunakan untuk mengidentifikasi mahasiswa secara unik karena dimungkinkan terhadap dua mahasiswa dengan nama, alamat, dan kota yang sama. Pengenal unik harus diberikan pada masing-masing mahasiswa.

Relasi adalah hubungan antara beberapa entitas. Sebagai contoh, relasi menghubungkan mahasiswa dengan mata kuliah yang di ambilnya. Kumpulan semua entitas bertipe sama disebut kumpulan entitas (*entity set*), sedangkan kumpulan semua relasi bertipe sama disebut kumpulan relasi (*relationship set*).

Struktur logis (*skemadatabase*) dapat ditunjukkan secara grafis dengan diagram ER yang dibentuk dari komponen-komponen pada tabel II.5 berikut :

Tabel II.5 Notasi ERD (*Entity Relationship Diagram*)

	Persegi panjang mewakili kumpulan entitas
	Elips mewakili atribut
	Belah ketupat mewakili relasi
	Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

Sumber : "*Basis Data (Janner Simarmata : 2006: 59)*"

II.8 Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relational (www.utexas.edu).

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

1. Bentuk Normal Pertama (1NF)
2. Bentuk Normal Kedua (2NF)
3. Bentuk Normal Ketiga (3NF)
4. Bentuk Normal Boyce-Code (BCNF)
5. Bentuk Normal Keempat (4NF)
6. Bentuk Normal Kelima (5NF)

Tujuan normalisasi adalah membuat kumpulan tabel relasional yang bebas dari data berulang yang dapat dimodifikasi secara benar dan konsisten. Ini berarti bahwa semua tabel pada basisdata relasional harus berada pada bentuk normal ketiga (3NF). Sebuah tabel relasional berada pada 3NF jika dan hanya jika semua kolom bukan kunci adalah (a) saling independen dan (b) sepenuhnya tergantung pada kunci utama. Saling independen berarti bahwa tidak ada kolom bukan kunci yang tergantung pada senbarang kombinasi kolom lainnya. Dua bentuk normal pertama adalah langkah mencapai tujuan, yaitu mempunyai semua tabel dalam 3NF. (Janner Simarmata : 2006: 77).