

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi Geografis (SIG)

Defenisi SIG berkembang, bertambah, dan bervariasi. Hal ini terlihat dari banyaknya defenisi SIG yang beredar. Lebih dari itu, SIG juga merupakan kajian ilmu & teknologi yang belum lama dikembangkan, digunakan oleh berbagai bidang/ilmu, dan berkembang dengan cepat. Sehubungan dengan hal ini, sebagai ilustrasi, berikut adalah beberapa defenisi SIG yang telah beredar di berbagai sumber pustaka :

1. SIG adalah sistem komputer untuk memasukkan (*capturing*), menyimpan (*store/record*), memeriksa, mengintegrasikan, memanipulasi, menganalisis, dan menampilkan data yang berhubungan dengan posisinya di permukaan bumi [Rice20].
2. SIG adalah kombinasi perangkat keras dan perangkat lunak; sistem komputer yang memungkinkan penggunaanya untuk mengelola (*manage*), menganalisa, dan memetakan informasi spasial berikut atributnya dengan akurasi kartografis [Basic20].
3. SIG adalah sistem yang berbasis komputer untuk menyimpan & memanipulasi informasi geografis. SIG dirancang untuk mengumpulkan, menyimpan, dan menganalisis objek & fenomena dimana lokasi geografis merupakan karakteristik penting atau kritis untuk dianalisis. Dengan demikian, SIG merupakan sistem komputer yang memiliki empat kemampuan dalam

menangani data yang bereferensi geografis: (a) masukan, (b) manajemen data (penyimpanan & pemanggilan data), (c) analisis & manipulasi data, dan (d) keluaran [Aronoff89].

4. SIG adalah sistem yang terdiri dari perangkat keras, perangkat lunak, data, manusia, organisasi & lembaga yang digunakan untuk mengumpulkan, menyimpan, menganalisis, dan menyebarkan informasi-informasi mengenai daerah-daerah di permukaan bumi [Chrisman97].
5. SIG adalah sistem komputer untuk memanipulasi data geografis. Sistem ini diimplementasikan dengan menggunakan perangkat keras dan perangkat lunak komputer yang berfungsi untuk : (a) akuisisi & verifikasi data, (b) kompilasi data, (c) penyimpanan data, (d) perubahan & *updating* data, (e) manajemen & pertukaran data, (f) manipulasi data, (g) pemanggilan & presentasi data, dan (h) analisa data [Bern92].
6. SIG adalah sistem komputer untuk mengumpulkan, memeriksa, mengintegrasikan, dan menganalisis informasi yang berhubungan dengan permukaan bumi [Demers97].
7. SIG adalah kumpulan yang terorganisir dari perangkat keras komputer, perangkat lunak, data geografis dan personil yang dirancang secara efisien untuk memperoleh, menyimpan, meng*update*, memanipulasi, menganalisis dan menampilkan semua bentuk informasi yang bereferensi geografis [Esri90].
8. SIG adalah sistem mendukung proses pengambilan keputusan spasial dan mampu mengintegrasikan deskripsi lokasi dengan karakteristik fenomena

yang ditemukan. SIG yang lengkap mencakup metodologi & teknologi yang diperlukan; yaitu data spasial, perangkat keras, perangkat lunak, dan struktur organisasi [Gistut94].

9. SIG merupakan sistem informasi yang dirancang untuk bekerja dengan data tereferensi secara spasial/ geografis. Dengan kata lain, SIG merupakan sistem basis data dengan kemampuan khusus data yang tereferensi secara geografi berikut sekumpulan operasi yang terkait dengan pengelolaan data tersebut [Foote95].
10. SIG adalah teknologi informasi yang dapat menganalisis, menyimpan, dan menampilkan baik data spasial maupun non spasial. SIG mengkombinasikan kekuatan perangkat lunak basis data relasional (*DBMS*) dan paket perangkat lunak *CAD* [Guo20].
11. SIG adalah fasilitas untuk mempersiapkan, mempresentasikan, dan menginterpretasikan fakta-fakta yang terdapat di permukaan bumi. Untuk defenisi yang lebih khusus, SIG adalah konfigurasi perangkat keras & perangkat lunak sistem komputer yang secara khusus dirancang untuk proses akuisisi, pengelolaan, dan penggunaan data kartografi [Tomlin90].
12. SIG adalah sistem informasi yang dirancang untuk bekerja dengan data yang tereferensikan secara spasial/ geografis. Dengan kata lain, SIG merupakan sistem basis data dengan kemampuan khusus dalam menangani data yang tereferensi secara spasial; selain merupakan sekumpulan operasi yang dikenakan terhadap data tersebut [Star90]. (Eddy Prahasta : 2014, Hal : 100).

II.1.1. Sub Sistem SIG

Jika defenisi-defenisi di atas diperhatikan, SIG dapat diuraikan menjadi beberapa sub-sistem sebaagai berikut :

1. *Data Input* : mengumpulkan, mempersiapkan, dan menyimpan data spasial & atributnya. Sub-sistem ini bertanggung jawab dalam mengonversikan format data aslinya ke dalam format SIG nya.
2. *Data Output* : menampilkan & menghasilkan data keluaran basis data spasial *softcopy* & *hardcopy* seperti halnya tabel, grafik, *report*, peta, dan lain sebagainya.
3. *Data Management* : mengorganisasikan data spasial & tabel atribut ke dalam sistem basis data hingga mudah untuk dipanggil kembali, *update*, dan *edit*.
4. *Data Manipulation & Analisis* : menentukan informasi yang dihasilkan oleh SIG. Selain itu, sub sistem ini memanipulasi dan memodelkan data untuk menghasilkan informasi yang diharapkan. (Eddy Prahasta, 2014, Hal : 102).

II.1.2. Komponen SIG

SIG merupakan sistem kompleks yang umumnya terintegrasi dengan sistem komputer lainnya di tingkat fungsional & jaringan. Jika diuraikan, SIG terdiri dari komponen dengan berbagai karakteristiknya.

1. Perangkat keras

SIG tersedia diberbagai *platform* perangkat keras; mulai dari kelas *PC desktop*, *workstation*, hingga *multi-user host*. Walaupun demikian, fungsionalitas SIG tidak terikat ketat pada karakteristik fisik perangkat kerasnya hingga

keterbatasan memori pada *PC* dapat diatasi. Adapun perangkat keras yang sering digunakan untuk aplikasi SIG adalah komputer (*PC/CPU*), *mouse*, *keyboard*, monitor (plus *VGA-Card* grafik) yang beresolusi tinggi, *digitizer*, *printer*, *plotter*, *receiver GPS*, dan *scanner*.

2. Perangkat Lunak

SIG merupakan sistem perangkat lunak dimana sistem basis datanya memegang peranan kunci. Pada SIG lama, sub sistem diimplementasikan oleh modul-modul perangkat lunak hingga tidak mengherankan jika ada perangkat SIG yang terdiri dari ratusan modul program (*.exe) yang dapat dieksekusi tersendiri.

3. Data & informasi geografis

SIG dapat mengumpulkan & menyimpan data/ informasi yang diperlukan baik tidak langsung (dengan meng-*import*-nya) maupun langsung dengan mendijitasi data spasialnya (*on-screen/ heads-ups* pada layar monitor atau cara manual dengan *digitizer*) dari peta analog dan memasukkan data atributnya dari tabel/ laporan dengan menggunakan *keyboard*.

4. Manajemen

Proyek SIG akan berhasil jika dikelola dengan baik & dikerjakan oleh orang yang memiliki keahlian yang tepat pada semua tingkatan. (Eddy Prahasta, 2014, Hal : 104).

II.1.3. Perangkat Keras SIG

Perangkat keras yang mendukung kebutuhan analisis spasial & pemetaan (SIG), sebenarnya, tidak jauh berbeda dengan perangkat keras lainnya. Perbedaannya, jika ada, terletak pada kebutuhan perangkat tambahan pendukung presentasi grafis beresolusi & kecepatan tinggi, dan mempercepat operasi-operasi manajemen basis data dengan volume data yang besar. Perangkat keras ini pada umumnya mencakup :

1. *CPU*

Perangkat ini bagian dari sistem komputer sebagai pemroses instruksi & program. *CPU* mengendalikan operasi di lingkungan sistem komputernya. Kebutuhan *CPU* bervariasi dari yang sederhana hingga yang canggih. Pada SIG yang agak besar diperlukan *CPU* dengan *processor* berkemampuan tinggi/ganda (*dual-core, core-2-duo, quad core*, dan yang lebih baru lagi).

2. *RAM*

Perangkat ini digunakan *CPU* untuk menyimpan data & program. Kebutuhan SIG mengenai *RAM* juga bervariasi. SIG sederhana memerlukan *RAM* 4 MB hingga 8 MB. Sementara SIG besar direkomendasikan untuk menggunakan *RAM* hingga 8 Gb.

3. *Storage*

Perangkat ini merupakan penyimpanan data permanen. Bila dibandingkan dengan *RAM*, akses pada media ini agak lambat. *Hardisk, disket, CD-ROM, flash-disk*, dan pita magnetis merupakan contoh *storage*. Kebutuhan *storage* sangat bervariasi mulai dari SIG yang satu ke SIG yang lain. SIG sederhana memerlukan

storage di bawah 5 MB. Sementara SIG besar bisa memerlukan *storage* dengan kapasitas hingga ratusan GB atau bahkan dalam satuan Tera Byte.

4. *Input device*

Perangkat ini merupakan peralatan untuk masukan data ke dalam SIG. Yang termasuk ke dalam perangkat ini adalah *keyboard, mouse, digitizer, receiver GPS, kamera digital, dan sejenisnya*.

5. *Output device*

Perangkat ini merupakan peralatan yang digunakan untuk menyajikan data & informasi SIG. Yang termasuk ke dalam perangkat ini adalah layar *monitor, printer, plotter, dan sejenisnya*.

6. *Peripheral*

Perangkat ini merupakan bagian dari sistem komputer SIG yang tidak termasuk ke dalam perangkat di atas. SIG sederhana tidak memerlukan peripheral, tetapi SIG besar memerlukan kabel jaringan, modem, ISP, *router, ethernet, CPU khusus untuk clients dan server, dan lain sejenisnya*.

Selain di pustaka [Antenucci91], perangkat keras SIG juga dijelaskan di pustaka [Hard20], [HardSoft20], dan [Wgiac20]. Sebagai ilustrasi, WGIAC (*Wyoming geographic information advisory council*) Pernah membuat standard perangkat keras SIG sebagai berikut [Wgiac20] :

- a. CPU: berbasiskan *processor* 32-bit Intel.
- b. RAM: minimal 32 MB.
- c. *Storage: harddisk* berkapasitas 1GB untuk *workstation* dengan jaringan, dan *harddisk* 2GB untuk *workstation* berdiri sendiri.

- d. *Input device: keyboard 101-key, digitizer* berdimensi minimum 24x36 (D size) berakurasi 0.005 *inchi*, *scanner* hitam- putih berukuran minimum 24x36 (D size) beresolusi 400 dpi, *scanner* berwarna berukuran 11x7 (B size) beresolusi 400 dpi.
- e. *Ouput device: monitor* beresolusi 1280x1024 dengan 256 warna dan memori 4 Mb, *printer* dengan teknologi lase/inkjet berukuran minimum kertas 11x7 (B size) dan resolusi minimum 300 dpi, *plotter* dengan teknologi inkjet resolusi minimum 300 dpi untuk ukuran kertas minimum 36x48 (E size).
- f. *Peripheral* lainnya: *receiver* GPS yang ada pada saat *draft* ini dibuat belum direkomendasikan standardnya. Walaupun demikian, ada beberapa spesifikasi *receiver* GPS yang perlu diperhatikan: untuk pemetaan (akurasi horizontal 5 meter) vs. Untuk survey (akurasi horizontal di bawah 1 cm), jumlah *channel* yang dapat diterima (6, 8, atau 12), antena internal vs. eksternal, kapasitas penyimpanan data, kemampuan *post-processing* dan konversi file ke format perangkat lunak standard SIG.

Sebagai ilustrasi tambahan, berikut persyaratan perangkat keras untuk SIG yang pernah beredar pada masa lampau:

- a. IDRISI versi 4.1 (DOS)

SIG ini berjalan *platform* PC dengan *processor* Intel 8086, 80286, 80386, 80486, dan yang lebih baik lagi, dan DEC-Alpha dan MIPS-4000. SIG ini memerlukan RAM 512 KB, card grafik EGA, VGA, super VGA, atau 8514/A, dan *printer* minimal bertipe *dot-matrix*. Walaupun demikian, SIG ini

disarankan dijalankan di PC dengan *harddisk* 80 MB, card grafik super-VGA, *mouse*, *processor* 80486 pentium atau yang lebih baik [Eastman93a].

b. MapInfo versi 4.0 (*Windows*)

SIG ini bekerja pada PC yang ber-processor minimum Intel 486, RAM 8 MB, monitor dengan card-VGA atau yang lebih baik, *harddisk* 19.5 MB untuk program dan 4 MB untuk data (disarankan minilam 100 MB) [Map95]. (Eddy Prahasta, 2014, Hal : 107).

II.1.4. Perangkat Lunak SIG

Pada sistem komputer, perangkat lunak biasanya terdiri dari beberapa *layer*. Model *layer* ini terdiri dari perangkat lunak sistem operasi, program pendukung sistem (*special system utilities*), dan perangkat lunak aplikasi [Antenucci91].

Sistem operasi terdiri dari program yang mengawasi jalannya operasi kesisteman dan mengendalikan komunikasi di antara perangkat kerasnya. *Special system utilities* & perangkat lunak aplikasi mengakses program sistem operasi untuk mengeksekusi fungsi-fungsinya.

Sistem operasi mengandung program-program yang mengelola memori, akses sistem, pengendalian komunikasi, pengolahan perintah, manajemen data & *file*, dan lain sejenisnya. *Special system utilities* dan program pendukungnya terdiri dari *complier* bahasa pemrograman, *device driver*, *utility*, pustaka fungsi & prosedur, dan perangkat lunak komunikasi khusus. Perangkat lunak aplikasi terdiri

dari *word processing*, *sphread sheet*, *DBMS*, *presentation*, *image processing*, *charting & drawing*, dan aplikasi khusus lainya seperti halnya SIG.

Perangkat lunak aplikasi SIG digunakan untuk menjalankan tugas-tugas SIG. Perangkat lunak tipe ini banyak tersedia dalam bentuk paket perangkat lunak yang terdiri dari multi-program yang terintegrasi untuk mendukung kemampuan khusus pemetaan dijital, manajemen, dan analisi spasial. Perangkatb lunak SIG, secara konseptual, terdiri dari 2 bagian : paket inti yang digunakan untuk pemetaan dijital dasar & manajemen data, dan paket aplikasi untuk menjalankan fungsionalitas pemetaan dijital khusus dan analisis spasial.

Pemilihan SIG bergantung pada beberapa factor: tujuan penggunaan biaya pembelian & pemeliharaan, kesiapan, & kemampuan personil, dan keberadaan agennya. Dalam kaitan ini, WGIAC membuat standard umum perangkat lunak SIG seperti berikut ([Wgiac20]) :

a. Sistem Operasi

Berbasiskan UNIX atau Ms. Windows (Win95, Win98, dan WinNt).

b. Model data spasial

raster & vector, teteapi dengan prioritas tinggi pada vector.

c. Basisdata (DBMS)

jika menggunakan sistem basisdata relasional, maka sistemnya harus sesuai dengan standard SQL (FIPS 127-2) sebagaimana telah dideskripsikan di dalam DBMS bagi standard aplikasi *multi-user*. Jika tidak menggunakan basisdata relasional, maka sistem basisdatanya harus mampu mengekspor/impor ke & dari DBMS relasional. (Eddy Prahasta, 2014, Hal : 109).

II.1.5. Cara Kerja SIG

SIG bisa menyajikan model “*real world*” di monitor sebagaimana peta di atas kertas. Walaupun demikian, SIG lebih memiliki kekuatan & fleksibilitas dari pada peta/kertas. Peta adalah bentuk sajian grafis dunia nyata; objek yang disajikan disebut unsur peta/*map feature*. Karena peta mengorganisasikan unturnya berdasarkan lokasinya, maka ia sangat baik dalam memperlihatkan hubungan unsur-unsurnya.

Peta menggunakan unsur-unsur geometri titik, garis, & polygon untuk menyajikan objek-objek dunia nyata sebagai berikut:

- a. Pada skala besar, sungai ditampilkan sebagai poligon, sementara pada skala kecil, sungai ditampilkan sebagai garis besar.
- b. Jalan bebas hambatan digambarkan sebagai garis-garis.
- c. Pada skala besar, bangunan disajikan sebagai poligon. Sementara pada skala kecil, objek yang sama disajikan sebagai titik.

Peta menggunakan symbol grafis (geometri), ukuran, dan warna untuk mengidentifikasi unsur-unsur spasial berikut deskripsinya. Berikut adalah beberapa diantaranya:

- a. Objek sungai diberi warna biru.
- b. Objek tanaman / kebun / hutan diberi warna hijau.
- c. Objek jalan bebas hambatan diberi warna merah.
- d. Objek jalan kecil/setapak digambarkan dengan garis-garis yang tipis warna hitam.
- e. Objek bangunan digambarkan sebagai geometri poligon.

f. Label & teks (anotasi) digunakan untuk mengidentifikasi unsur-unsur peta dengan menggunakan nama-nama objek-objeknya.

Skala peta menentukan ukuran & bentuk sajian unsur-unsurnya. Makin besar nilainya, makin besar pula ukuran unsurnya. Sebagai ilustrasi, pada peta berskala 1:250,000 atau yang lebih kecil, kota disajikan sebagai titik, sementara jalan & sungai disajikan sebagai garis-garis.

Sementara itu, pada peta berskala 1:25,000 atau yang lebih besar, kota akan disajikan sebagai poligon, sementara jalan & sungai bisa disajikan sebagai garis-garis / poligon.

SIG menyimpan informasi deskriptif unsur spasialnya sebagai atribut yang disimpan di table tersendiri atau di DBMS-nya. Setelah itu, SIG mengkaitkan (tagging) unsur spasialnya dengan record pada table tersebut. Oleh karena itu, atribut spasial dapat diakses melalui lokasi objeknya. Dan sebaliknya, Objek spasial dapat diakses melalui atributnya. Dengan demikian, objek spasial dapat di-*query* & di-*search* berdasarkan atributnya.

Dengan relasi antar-objek spasial dengan atributnya, maka sebagai misal, pengguna bisa mencari rute terpendek yang menghubungkan hotel “Mulia” dengan rumah sakit “Sumber Waras”. Pekerjaan ini dapat dilakukan dengan beberapa langkah berikut:

a. Identifikasilah lokasi hotel “Mulia” & lokasi rumah sakit “Sumber Waras” pada peta. Tentukan salah satunya sebagai titik awal, sementara yang lain sebagai akhir (tujuan).

- b. Identifikasilah jalan yang terdekat dengan titik awal, dan telusurilah kombinasi segmen jalan yang menghubungkan titik awal ke jalan yang terdekat dengan titik akhir.
- c. Periksalah apakah segmen-segmen jalan yang terpilih benar-benar menghubungkan titik awal ke titik akhir dengan mempertimbangkan aturan lalu-lintas, belokan, dan arah-arah pemakaian jalannya (searah, dua-arah, berlawanan, *perboden*, jalan buntu, dan lain sejenisnya). Jika tidak, ulangi langkah b).
- d. Hitunglah setiap jarak (panjang segmen) beserta akumulasinya.
- e. Pilihlah rute yang berjarak total (segmen) terkecil (terpendek).

SIG menghubungkan objek spasial dengan atributnya yang tersimpan. Kumpulan objek “sungai”, “bangunan”, “jalan”, “laut”, “perkebunan”, dan “hutan” merupakan layer. Kumpulan layer & tabel atributnya akan membentuk basisdata spasial SIG. Dengan demikian, perancangan basisdata merupakan esensial di dalam SIG. Rancangan basisdata menentukan efektifitas & efisiensi proses masukan, pengolahan, dan keluaran SIG itu sendiri. (Eddy Prahasta, 2014, Hal : 111).

II.1.6. Kemampuan SIG

Pada dasarnya, kemampuan/fitur SIG sudah dikenali. Kemampuan ini dinyatakan dalam fungsi analisis spasial & atributnya, jawaban, atau solusi yang diberikan terhadap pertanyaan yang diajukan. (Eddy Prahasta, 2014, Hal : 115).

II.2. Basis Data Dan DBMS

Basis data dapat didefinisikan sebagai koleksi dari data-data yang terorganisasi sedemikian rupa sehingga data mudah disimpan dan dimanipulasi (diperbarui, dicari, diolah dengan perhitungan-perhitungan tertentu, serta dihapus). Secara teoritis, basis data tidak harus berurusan dengan komputer (misalnya, catatan belanja hari ini yang dibuat oleh seorang ibu rumah tangga juga merupakan basis data dalam bentuk yang sangat sederhana). (Adi Nugroho, 2011, Hal : 4).

Menurut Abdul Kadir (2014) basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktifitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut *Database Management System (DBMS)*. *DBMS* adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol dan mengakses basis data dengan cara yang praktis dan efisien. *DBMS* dapat digunakan untuk mengakomodasikan berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda-beda. (Abdul Kadir, 2014, Hal : 218).

Umumnya *DBMS* menyediakan fitur-fitur sebagai berikut :

a. Independensi data-program

Karena basis data ditangani oleh *DBMS*, program dapat dipilih sehingga tidak tergantung pada struktur data dalam basis data. Dengan perkataan lain, program tidak akan terpenaruh sekiranya bentuk fisik data diubah.

b. Keamanan

Keamanan dimaksudkan untuk mencegah pengaksesan data oleh orang yang tidak berwenang.

c. Integritas

Hal ini ditujukan untuk menjaga agar data selalu dalam keadaan yang valid dan konsisten.

d. Konkurensi

Konkurensi memungkinkan data dapat diakses oleh banyak pemakai tanpa menimbulkan masalah.

e. Pemulihan (*recovery*)

DBMS menyediakan mekanisme untuk mengembalikan basis data ke keadaan semula yang konsisten sekiranya terjadi gangguan perangkat keras atau kegagalan perangkat lunak.

f. Katalog Sistem

Katalog Sistem adalah deskripsi tentang data yang terkandung dalam basis data yang dapat diakses oleh pemakai.

g. Perangkat Produktivitas

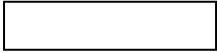


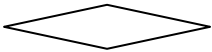
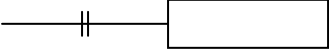
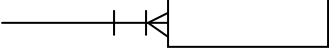
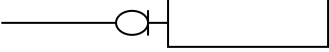

Untuk menyediakan kemudahan bagi pemakai dan meningkatkan produktivitas, *DBMS* menyediakan sejumlah perangkat produktivitas seperti pembangkit *query* dan pembangkit laporan. (Abdul Kadir, 2014, Hal : 219).

II.3. Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah bagian yang menunjukkan hubungan antara entity yang ada dalam sistem. Simbol-simbol yang digunakan

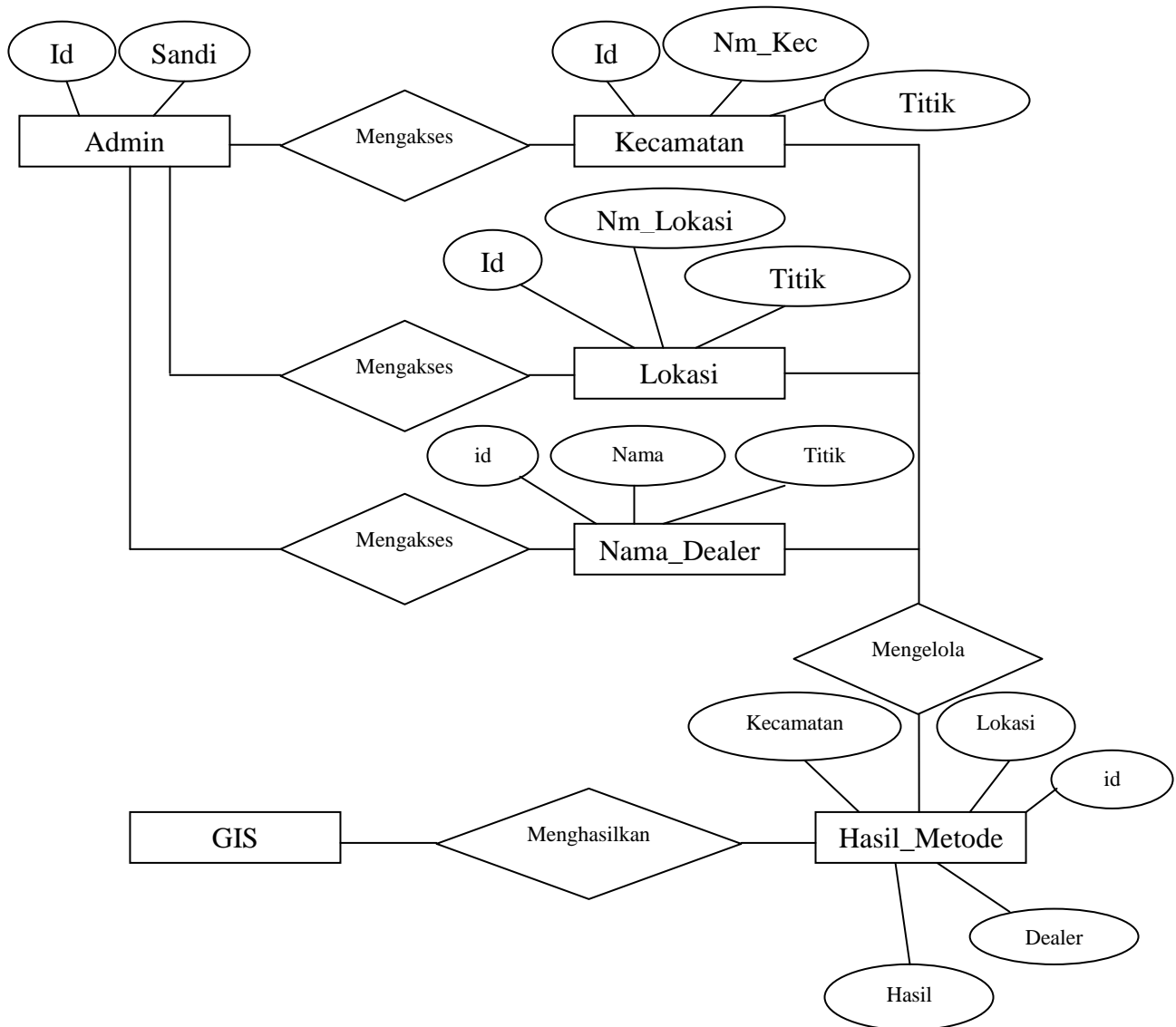
dapat dilihat dari tabel II.1. (Yuhendra, M.T, Dr. Eng dan Riza Eko Yulianto, 2015, Hal : 70).

Tabel II.1. Simbol Yang Digunakan Pada *Entity Relationship Diagram* (ERD)

SIMBOL	KETERANGAN
	<i>Entity</i>
	Atribut Dan <i>Entity</i>
	Atribut Dan <i>Entity</i> Dengan <i>Key</i> (Kunci)
	Relasi Atau Aktifitas Antar <i>Entity</i>
	Hubungan Satu Dan Pasti
	Hubungan Banyak Dan Pasti
	Hubungan Satu Tapi Tidak Pasti
	Hubungan Banyak Tapi Tidak Pasti

(Sumber : Yuhendra, M.T, Dr. Eng dan Riza Eko Yulianto; 2015)

Berikut adalah contoh penggunaan *Entity Relationship Diagram (ERD)* :



Gambar II.1. Contoh *Entity Relationship Diagram (ERD)*

(Sumber : Yuhendra, M.T, Dr. Eng dan Riza Eko Yulianto; 2015)

II.4. Kamus Data

Kamus data merupakan sebuah daftar yang terorganisasi dari elemen data yang berhubungan dengan sistem, dengan definisi yang tegas dan teliti sehingga

pemakai dan analisis sistem akan memiliki pemahaman yang umum mengenai *input*, *output*, komponen penyimpanan. (Yusi Ardi Binarso, 2012, Hal : 74).

Contoh kamus data untuk entitas wisuda dan alumni yang digunakan berdasarkan sistem informasi alumni teknik informatika adalah sebagai berikut :

1. Data Wisuda

Wisuda = @idWisuda + bulan + tahun + jumlahPeserta

@Wisuda = { integer }

Bulan = [januari|...|Desember]

Tahun = [2000|...|2999]

Jumlah Peserta = { integer }

Integer = [0-9]

2. Data Alumni

Alumni = @NIM + namaLengkap + password + email + tglLahir +
 jenisKelamin + noTelp + alamatAsal + kotaAsal +
 alamatSekarang + kotaSekarang + instansi + jabatan + judulTA +
 lamaTA + tglLulus + lamaStudi + IPK + foto + jawabanPK +
 kunciAktivasi + statusAktif + idWisuda + idPertanyaan

@NIM = 1 { character } 14

namaLengkap = 1 { character } 50

password = 1 { character } 50

email = 1 { character } 25

tglLahir = date

jenisKelamin = { L | P }

noTelp	= 1 { character } 20
alamatAsal	= 1 { character } 100
kotaAsal	= 1 { character } 20
alamatSekarang	= 1 { character } 100
kotaSekarang	= 1 { character } 20
instansi	= 1 { character } 50
jabatan	= 1 { character } 50
judulTA	= 1 { character } 250
lamaTA	= { integer }
tglLulus	= date
lamaStudi	= { integer }
IPK	= decimal
Foto	= 1 { character } 50
JawabanPK	= 1 { character } 20
kunciAktivasi	= 1 { character } 65
statusAktif	= [aktif non aktif]
idWisuda	= *dapat dilihat pada data Wisuda
idPertanyaan	= *dapat dilihat pada data Pertanyaan_Keamanan
character	= [A-Z a-z 0-9]
integer	= [0-9]

II.5. Normalisasi

Normalisasi dapat dipahami sebagai tahapan-tahapan yang masing-masing berhubungan dengan bentuk normal. Bentuk normal adalah keadaan relasi yang

dihasilkan dengan menerapkan aturan sederhana berkaitan dengan konsep kebergantungan fungsional pada relasi yang bersangkutan. Kita akan menggambarkannya secara garis besar sebagai berikut :

1. Bentuk Normal Pertama (1NF/ *First Normal Form*)

Bentuk normal pertama adalah suatu bentuk relasi dimana atribut bernilai banyak (*multivalued attribute*) telah dihilangkan sehingga kita akan menjumpai nilai tunggal (mungkin saja nilai *null*) pada perpotongan setiap baris dan kolom.

2. Bentuk Normal Kedua (2NF/ *Second Normal Form*)

Semua kebergantungan fungsional yang bersifat sebagian (*partial functional dependency*) telah dihilangkan.

3. Bentuk Normal Ketiga (3NF/ *Third Normal Form*)

Semua kebergantungan transitif (*transitive dependency*) telah dihilangkan.

4. Bentuk Normal *Boyce-Codd* (BCNF/ *Boyce-Codd Normal Form*)

Semua anomaly yang tersisa dari hasil penyempurnaan kebergantungan fungsional sebelumnya telah dihilangkan.

5. Bentuk Normal Keempat (4NF/ *Fourth Normal Form*)

Semua kebergantungan bernilai banyak telah dihilangkan.

6. Bentuk Normal Kelima (5NF/ *Fifth Normal Form*)

Semua anomaly yang tertinggi telah dihilangkan. (Adi Nugroho, 2011, Hal : 199).

Berikut ini adalah contoh normalisasi :

Bentuk tidak normal

Tabel II.2. Bentuk Tidak Normal

ID	Admin	Kecamatan	Titik
1	Anto	Medan Denai	3,66666
2	Andi	Medan Area	3,65843

Bentuk normal pertama

Tabel II.3. Bentuk Normal Pertama

D	Kecamatan	Titik
1	Medan Denai	3,66666
2	Medan Area	3,65843

Bentuk normal kedua

Tabel II.4. Bentuk Normal Kedua

Kecamatan	Titik
Medan Denai	3,66666
Medan Area	3,65843

II.6. Database Mysql

Database MySQL merupakan aplikasi yang bersifat *daemon* atau menetap dalam memori yang berjalan bersama dengan sistem operasi *Microsoft Windows*. *Interface* utama *MySQL database server* adalah *command line* atau berbasis *DOS* sehingga diperlukan pengetahuan khusus mengenai penggunaan perintah atau *commandi* dalam *command shell MySQL*. (Wahana Komputer, 2011, Hal : 2).

II.7. Hypertext Preprocessor (PHP)

Hypertext Preprocessor adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML*. *PHP* banyak dipakai untuk memrogram situs *web* dinamos. *PHP* dapat digunakan untuk membangun sebuah *CMS*. Pada awalnya

PHP merupakan kependekan dari *Personal Home Page* (situs personal). *PHP* pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu *PHP* masih bernama *Form Interpreted (FI)*, yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya *PHP/FI*. Dengan perilisannya kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan *PHP*. (Jurnal Momentum, Anisya, 2013, Vol.15 No.2. Agustus 2013).

II.8. *Unified Modeling Language (UML)*

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. *UML* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.


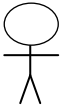


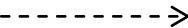
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Gellysa Urva dan Helmi Fauzi Siregar, 2015).

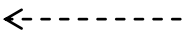
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis *UML* adalah sebagai berikut:

1. Use case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.5 dibawah ini:

Tabel II.5. Simbol Use Case

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.




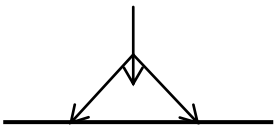
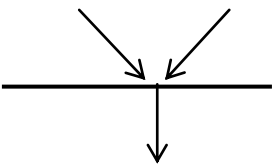
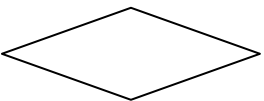

	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.
---	---

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.6 dibawah ini:

Tabel II.6. Simbol *Activity Diagram*

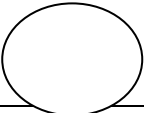
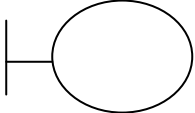
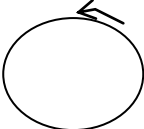
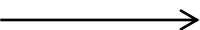
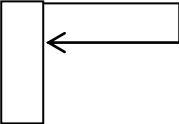

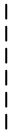
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.7 dibawah ini :

Tabel II.7. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.8 dibawah ini:

Tabel II.8. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015)

II.9. *Algoritma Greedy*

Algoritma *greedy* merupakan algoritma yang paling populer untuk memecahkan masalah optimasi. Algoritma ini membentuk solusi langkah per langkah. Pada setiap langkah, banyak pilihan yang perlu dieksplorasi. Oleh karena itu, pada setiap langkah harus dibuat keputusan terbaik dalam menentukan pilihan. (Yongke Yoswara, 2011).