

BAB II

TINJAUAN PUSTAKA

II.1. Kota

Yunus (1987, dalam Helmi et al., 2006) menjelaskan bahwa kota adalah suatu daerah tertentu dengan tata guna lahan non agraris, tata guna lahan mana sebagian terbesar tertutup oleh bangunan yang bersifat *non* residensial maupun residensial (gedung lebih banyak dari vegetasi), pola jaringan jalan yang kompleks, dalam satuan yang kompak dan relatif lebih besar pada satuan permukiman di sekitarnya. Sementara itu sudah dilengkapi dengan fasilitas-fasilitas kota . Lahan yang dekat dengan pusat perdagangan di sekitar atau pusat kota akan mempunyai nilai tinggi dan nilai ini akan semakin berkurang dengan bertambah jauh jaraknya dari pusat kota.

II.2. Sistem Informasi Geografis dalam manajemen profil

Profil dalam Sistem Informasi Geografis ini maksudnya yaitu keterangan secara terperinci yang digunakan untuk mengetahui keadaan yang sebenarnya dalam suatu kota ataupun kabupaten. Dengan memanfaatkan pendekatan spasial dan keunggulan Sistem Informasi Geografis, pengembangan manajemen aset merupakan salah satu investasi Pemerintah Daerah untuk berbagai kepentingan strategis lainnya. Disamping sebagai sistem yang handal untuk membantu pengelolaan manajemen aset serta sebagai salah satu sarana bagi promosi potensi ekonomi daerah, manajemen aset juga merupakan pondasi bagi pengembangan

sistem informasi manajemen wilayah (*estate management*) yang akan sangat dibutuhkan, seiring dengan perkembangan dan pertumbuhan kota. Pengembangan manajemen aset di daerah untuk mendukung pengelolaan kawasan kabupaten/kota akan sangat mudah dilakukan karena manajemen aset dikembangkan dengan berbasis geografis dan memberikan keuntungan ganda, bukan saja sebagai alat (*tools*) untuk membantu pengelolaan aset, namun juga dasar bagi pengelolaan wilayah kabupaten/kota. Dengan demikian pengembangan sistem informasi ini merupakan suatu investasi penting bagi pemerintah daerah kabupaten/kota dalam mewujudkan *good corporate governance*. Dalam aplikasinya terhadap kebutuhan Sistem Informasi Geografis dalam menyediakan data mengenai lingkungan hidup sangat penting. Saat ini kebutuhan akan data dan peta profil yang menggambarkan potensi dan sangat jarang dijumpai terutama di daerah kabupaten/kota kecil. Hal ini juga disebabkan untuk membangun sebuah informasi yang handal memerlukan biaya yang cukup tinggi. (Arif Nur Hidayat : 2010)

II.3. Sistem Informasi Geografis

Sistem informasi geografis atau disingkat SIG merupakan terjemahan dari *Geographic Information System*. Sistem informasi geografi adalah suatu sistem yang berfungsi mengumpulkan, mengatur, mengelola, menyimpan, dan menyajikan segala jenis data (informasi) yang berkaitan dengan kondisi geografis suatu wilayah. (K.Wardiyatmoko ; 95 : 2014)

Menurut Riyanto (2009), SIG adalah sistem informasi yang digunakan untuk memasukkan, menyimpan, memanggil kembali, mengolah, menganalisa dan menghasilkan data bereferensi geografis atau data spasial. Komponen SIG terdiri atas perangkat keras (*hardware*), perangkat lunak (*software*), data geospasial dan pengguna.

II.3.1. Konsep Dasar Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) atau juga dikenal sebagai *Geographic Information System (GIS)* pertama pada tahun 1960 yang bertujuan untuk menyelesaikan permasalahan geografis 40 tahun kemudian. *GIS* berkembang tidak hanya bertujuan untuk menyelesaikan permasalahan geografi saja tetapi sudah merambah ke berbagai bidang seperti analisis penyakit epidemik (demam berdarah) dan analisis kejahatan (kerusuhan) termasuk analisis kepariwisataan.

Sistem informasi geografis mulai dikenal pada awal 1980-an. Sejalan dengan berkembangnya perangkat komputer, baik perangkat lunak maupun perangkat keras, Sistem Informasi Geografis berkembang sangat pesat pada era 1990-an. Secara harafiah, Sistem Informasi Geografis dapat diartikan sebagai : "Suatu komponen yang terdiri dari perangkat keras, perangkat lunak, data geografis dan sumberdaya manusia yang bekerja bersama secara efektif untuk menangkap, menyimpan, memperbaiki, memperbaharui, mengelola, memanipulasi, mengintegrasikan, menganalisa, dan menampilkan data dalam suatu informasi berbasis geografis".

GIS atau sistem informasi berbasis pemetaan dan geografi adalah sebuah alat bantu manajemen berupa informasi berbantuan komputer yang terkait dengan sistem pemetaan dan analisis terhadap segala sesuatu, serta peristiwa-peristiwa yang terjadi di muka bumi. Teknologi GIS mengintegrasikan operasi pengolahan data berbasis database yang biasa digunakan, seperti pengambilan data berdasarkan kebutuhan serta analisis statistik dengan menggunakan visualisasi yang khas serta berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar tertentu.

Sistem Informasi Geografis menghubungkan sebuah lokasi ke informasi (seperti seseorang dengan alamat, bangunan dengan sebuah paket, atau sebuah jalan yang berada dalam jaringan jalan) dan lapisan-lapisan yang memberikan informasi kepada pemahaman yang lebih tentang bagaimana kesemuanya itu dapat saling terhubung.

II.3.2. Definisi Sistem Informasi Geografis

Istilah geografi digunakan karena *GIS* dibangun berdasarkan pada 'geografi' atau 'spasial'. *Object* ini mengarah pada spesifikasi lokasi dalam suatu *space*. Objek bisa berupa fisik, budaya, atau ekonomi alamiah. Penampakan tersebut ditampilkan pada suatu peta untuk memberikan gambaran yang representatif dari spasial suatu objek sesuai dengan kenyataannya di bumi. Simbol, warna dan gaya garis digunakan untuk mewakili setiap spasial yang berbeda pada peta dua dimensi.

Chrisman (1997), SIG adalah sistem yang terdiri dari perangkat keras, perangkat lunak, data, manusia (*brainware*), organisasi dan lembaga yang digunakan untuk mengumpulkan, menyimpan, menganalisis, dan menyebarkan informasi-informasi mengenai daerah-daerah di permukaan bumi.

Geografi adalah informasi mengenal permukaan bumi dan semua obyek yang berada di atasnya, sedangkan sistem informasi geografis (SIG) atau dalam bahasa Inggris disebut *Geographic Information System (GIS)* adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Sistem informasi geografis adalah bentuk sistem informasi yang menyajikan informasi dalam bentuk grafis dengan menggunakan peta sebagai antar muka. SIG tersusun atas konsep beberapa lapisan (*layer*) dan relasi. (Prahasta, 2002)

II.3.3. Manfaat Sistem Informasi Geografis

Menurut (I Wayan Eka Swastikayana, 2011), SIG mampu memberikan kemudahan-kemudahan yang diinginkan. Dengan SIG kita akan dimudahkan dalam melihat fenomena kebumiharian dengan perspektif yang lebih baik. SIG mampu mengakomodasi penyimpanan, pemrosesan, dan penayangan data spasial digital bahkan integrasi data yang beragam, mulai dari citra satelit, foto udara, peta bahkan data statistik. SIG juga mengakomodasi dinamika data, pemutakhiran data yang akan menjadi lebih mudah.

II.3.4. Subsistem Sistem Informasi Geografis

Menurut (*Prahasta, 2005*), SIG dapat diuraikan menjadi beberapa subsistem sebagai berikut :

1. Data Input

Subsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spasial dan atribut dari berbagai sumber. Subsistem ini juga bertanggung jawab dalam mengkonversi atau mentransformasikan format-format data aslinya ke dalam format yang dapat digunakan oleh SIG.

2. Data Output

Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basisdata baik dalam bentuk *softcopy* maupun dalam bentuk *hardcopy* seperti : tabel, grafik, peta, dan lain-lain.

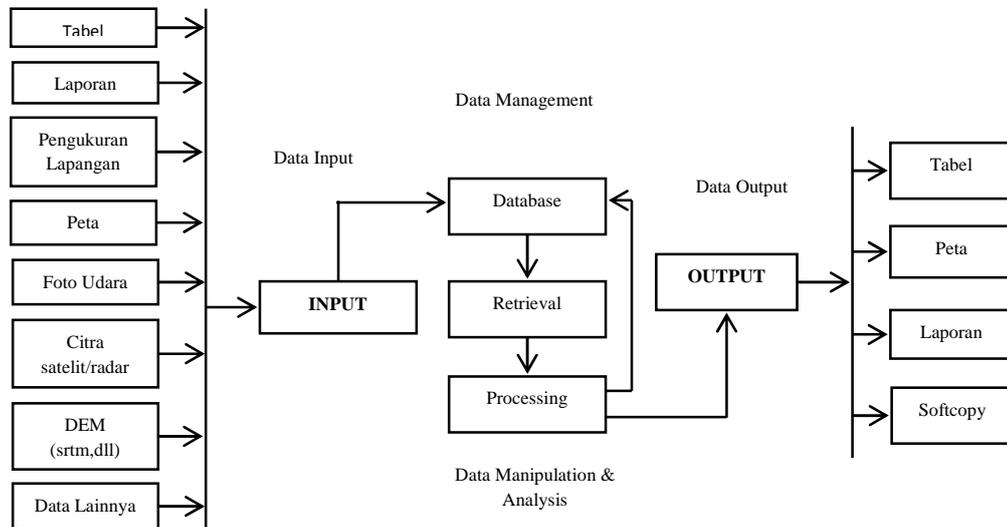
3. Data Manajemen

Subsistem ini mengorganisasikan baik data spasial maupun atribut kedalam sebuah basisdata sedemikian rupa sehingga mudah dipanggil, di-*update* dan di-*edit*.

3. Analisis dan Manipulasi Data

Subsistem ini menentukan informasi – informasi yang dapat dihasilkan oleh SIG. Selain itu, subsistem ini juga melakukan manipulasi dan pemodelan data untuk menghasilkan informasi yang diharapkan.

Sub-sistem SIG di atas dapat diilustrasikan sebagai berikut :



Gambar II.1. Ilustrasi Uraian Sub-Sistem SIG

Sumber (Jurnal Sistem Informasi Geografis (AK-011225))

II.3.5. Cara Kerja SIG

SIG dapat menyajikan *real world* (dunia nyata) pada monitor sebagaimana lembaran peta dapat merepresentasikan dunia nyata diatas kertas. Tetapi, SIG memiliki kekuatan lebih dan fleksibilitas dari pada lembaran pada kertas. Peta merupakan representasi grafis dari dunia nyata, obyek-obyek yang dipresentasikan di atas peta disebut unsur peta atau map features (contohnya adalah sungai, taman, kebun, jalan dan lain-lain). Karena peta mengorganisasikan unsur-unsur berdasarkan lokasi-lokasinya.

SIG menyimpan semua informasi deksriptif unsur-unsurnya sebagai atribut-atribut didalam basis data. Kemudian, SIG membentuk dan menyimpannya didalam tabel-tabel (*relasional*) dengan demikian, atribut-atribut ini dapat diakses melalui lokasi-lokasi unsur-unsur peta dan sebaliknya, unsur-unsur peta juga dapat diakses melalui atribut-atributnya. (Prahasta Eddy, 2005)

II.3.6. Kemampuan SIG

Kemampuan dasar dari SIG adalah mengintegrasikan berbagai operasi basis data seperti query, menganalisisnya serta menampilkannya dalam bentuk pemetaan berdasarkan letak geografisnya. Inilah yang membedakan SIG dengan sistem informasi lain. (Nuryadin, Ruslan; 2005) Sistem Informasi Geografis (SIG) adalah sistem informasi yang mampu memvisualisasikan aspek-aspek spasial dan *non* spasial untuk dapat diketahui dan dianalisa.

Sistem informasi geografis mempunyai kemampuan untuk menghubungkan berbagai data pada suatu titik tertentu di bumi, menggabungkannya, menganalisis dan akhirnya memetakan hasilnya : (Prahasta, Eddy, 2009)

1. Memasukkan dan mengumpulkan data geografis (spasial dan atribut)
2. Mengintegrasikan data geografis.
3. Memeriksa, meng-*update* (meng-*edit*) data geografis.
4. Menyimpan atau memanggil kembali data geografis.
5. Mempresentasikan atau menampilkan data geografis.
6. Mengelola, memanipulasi dan menganalisis data geografis.
7. Menghasilkan *output* data geografis dalam bentuk peta tematik (*view* dan *layout*), tabel, grafik (*chart*) laporan, dan lainnya baik dalam bentuk *hardcopy* maupun *softcopy*.

II.3.7. Komponen SIG

Menurut (*John E. Harmon, Steven J. Anderson, 2003*), secara rinci SIG dapat beroperasi dengan komponen- komponen sebagai berikut :

- a. Orang yang menjalankan sistem meliputi orang yang mengoperasikan, mengembangkan bahkan memperoleh manfaat dari sistem. Kategori orang yang menjadi bagian dari SIG beragam, misalnya operator, analis, *programmer, database administrator* bahkan *stakeholder*.
- b. Aplikasi merupakan prosedur yang digunakan untuk mengolah data menjadi informasi. Misalnya penjumlahan, klasifikasi, rotasi, koreksi geometri, *query, overlay, buffer, jointable*, dsb.
- c. Data yang digunakan dalam SIG dapat berupa data grafis dan data atribut.
 - Data posisi/koordinat/grafis/ruang/spasial, merupakan data yang merupakan representasi fenomena permukaan bumi/keruangan yang memiliki referensi (koordinat) lazim berupa peta, foto udara, citra satelit dan sebagainya atau hasil dari interpretasi data-data tersebut.
 - Data atribut/*non*-spasial, data yang merepresentasikan aspek-aspek deskriptif dari fenomena yang dimodelkannya. Misalnya data sensus penduduk, catatan survei, data statistik lainnya.
- d. *Software* adalah perangkat lunak SIG berupa program aplikasi yang memiliki kemampuan pengelolaan, penyimpanan, pemrosesan, analisis dan penayangan data spasial (contoh : *ArcView, Idrisi, ARC/INFO, ILWIS, MapInfo*, dll).
- e. *Hardware*, perangkat keras yang dibutuhkan untuk menjalankan sistem berupa perangkat komputer, printer, scanner, dan perangkat pendukung lainnya.

II.4. Peta

Peta adalah gambaran konvensional dari permukaan bumi yang diperkecil sesuai kenampakannya dari atas. Apeta umumnya digambarkan dalam bidang datar dan dilengkapi skala, orientasi, dan simbol-simbol. Dengan kata lain peta adalah gambaran permukaan bumi yang diperkecil sesuai dengan skala. Supaya dapat dipahami oleh pengguna atau pembaca, peta harus diberi tulisan dan simbol-simbol. (K.Wardiyatmoko ; 68 : 2014)

Peta merupakan bagian permukaan bumi yang disajikan dalam berbagai cara yang berbeda, mulai dari peta konvensional yang tercetak hingga peta digital yang tampil di layar komputer. Peta dapat digambarkan dengan berbagai gaya, masing-masing menunjukkan permukaan yang berbeda untuk subjek yang sama untuk memvisualisasikan dunia dengan mudah, informatif dan fungsional. Peta berbasis komputer (*digital*) lebih serba guna dan dinamis karena bisa menunjukkan banyak *view* yang berbeda dengan subjek yang sama. Peta ini juga memungkinkan perubahan skala, animasi gabungan, gambar, suara, dan bisa terhubung ke sumber informasi tambahan melalui internet. Peta digital dapat *update* ke peta tematik baru dan bisa menambahkan detail informasi geografi lainnya. (Denny Carter, Irma Agtrisari, 2003)

II.5. PHP

PHP singkatan dari *PHP Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *Web* yang disisipkan pada dokumen *HTML*. *PHP* memungkinkan *Web* dapat dibuat dinamis sehingga *maintenance* situs *Web* tersebut menjadi lebih mudah dan efisien. *PHP* merupakan

software Open-Source yang disebar dan dilisensikan secara gratis serta dapat di *download* secara bebas situ resminya. (Kasiman Peranginangin ; 2006 : 2)

Salah satu kelebihan dari *PHP* adalah mampu berkomunikasi dengan berbagai *database* yang terkenal. Dengan demikian, menampilkan data yang bersifat dinamis, yang diambil dari *database*, merupakan hal yang mudah untuk diimplementasikan. Itulah sebabnya sering dikatakan bahwa *PHP* sangat cocok untuk membangun halaman-halaman *web* dinamis. (Abdul Kadir ; 2008 : 6)

II.5.1. Konsep Kerja PHP

Model kerja *HTML* diawal dengan permintaan suatu halaman *web* oleh *browser*. Berdasarkan *URL (Uniform Resource Locator)* atau dikenal dengan sebutan alamat internet, *browser* mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*.

Selanjutnya, *web server* akan mencarikan file yang diminta dan memberikan isinya ke *web browser* (atau yang biasa disebut browser saja). *Browser* yang mendapatkan isinya segera melakukan proses penerjemahan kode *HTML* dan menampilkannya ke layar pemakai. (Abdul Kadir ; 2008 : 4)

II.6. WebGIS

Sistem Informasi Geografis merupakan sistem yang dirancang untuk bekerja dengan data yang terreferensi secara spasial atau koordinat-koordinat geografi. GIS memiliki kemampuan untuk melakukan pengolahan data dan melakukan operasi-operasi tertentu dengan menampilkan dan menganalisa data.

Applikasi GIS saat ini tumbuh tidak hanya secara jumlah aplikasi namun juga bertambah dari jenis keragaman aplikasinya.

Pengembangan aplikasi GIS kedepannya mengarah kepada aplikasi berbasis *web* yang dikenal dengan *WebGIS*. Hal ini disebabkan karena pengembangan aplikasi di lingkungan jaringan telah menunjukkan potensi yang besar dalam kaitannya dengan *geo* informasi. Sebagai contoh adalah adanya peta *online* sebuah kota dimana pengguna dapat dengan mudah mencari lokasi yang diinginkan secara *online* melalui jaringan intranet tanpa mengenal batas geografi penggunaannya.

Secara umum Sistem Informasi Geografis dikembangkan berdasarkan pada prinsip *input*/masukan data, manajemen, analisis dan representasi data. Di lingkungan *web* prinsip-prinsip tersebut di gambarkan dan di implementasikan seperti pada tabel berikut :

Tabel II.1 Prinsip-prinsip Pengembangan SIG

<i>GIS</i> Prinsip	Pengembangan <i>Web</i>
Data <i>Input</i>	<i>Client</i>
Manajemen Data	<i>DBMS</i> dengan komponen spasial
Analisis Data	<i>GIS Library</i> di <i>Server</i>
Representasi Data	<i>Client/Server</i>

Sumber (Arif Nur Hidayat ; 2010)

Untuk dapat melakukan komunikasi dengan komponen yang berbeda-beda di lingkungan *web* maka dibutuhkan sebuah *web server*. Karena standart dari geografis data berbeda-beda dan sangat spesifik maka pengembangan arsitektur sistem mengikuti arsitektur '*Client Server*'.

II.7. *MySql*

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *databasenya*. Selain itu ia bersifat *Open Source* (Anda tidak perlu membayar untuk menggunakannya) pada pelbagai *platform* (kecuali untuk jenis *Enterprise*, yang bersifat komersial).

MySQL termasuk jenis *RDBMS* (*Relational Database Management System*). Itulah sebabnya istilah seperti tabel, baris, dan kolom digunakan pada *MySQL*. Pada *MySQL*, sebuah *database* mengandung satu atau beberapa kolom. (Abdul Kadir ; 2008 : 348)

II.8. *Database*

Database Management System (DBMS) adalah perangkat lunak yang digunakan untuk mengelola dan mengontrol pengaksesan *database*. Contoh DBMS terkenal adalah *Microsoft Access* dan *Oracle*. (Abdul Kadir;2010:4)

Database Management System (DBMS) merupakan satu koleksi data yang saling berelasi dan satu *set* program untuk mengakses data tersebut. Jadi DBMS terdiri dari *database* dan *set* program pengelola untuk menambah, menghapus data, mengambil data dan membaca data.

II.8.1. ERD (Entity Relationship Diagram)

Menurut (Wawang Juwandi, 2007) ERD (*Entity Relationship Diagram*) adalah suatu objek yang dapat didefinisikan dalam lingkungan pemakai.

II.8.1.1. Komponen Utama ERD

1. Entitas

Merupakan sebuah objek yang keberadaannya dapat dibedakan terhadap objek lain. Entitas dapat berupa orang, benda, tempat, kejadian, konsep.

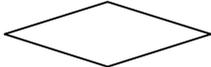
2. Atribut

properti/karakteristik yg melekat dlm suatu entitas, dimana properti itu bermakna bagi organisasi/perusahaan.

3. Kerelasian

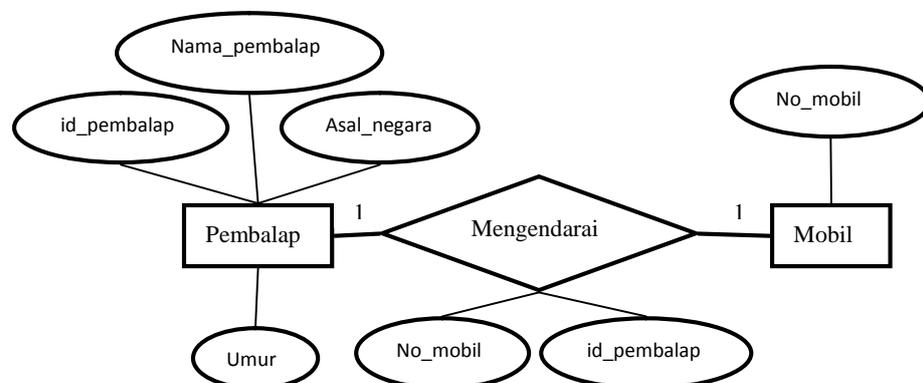
Hubungan antara beberapa tipe entitas. (*Ema Utami ; 2015*)

Table II.2. Simbol dalam ERD (Entity Relation Diagram)

Gambar	Fungsi
	Entitas
	Hubungan
	Atribut
	Link

Sumber (Ema Utami ; 2015)

Contoh ERD dapat digambarkan sebagai berikut :



Gambar II.2. ERD (Entity Relation Diagram)

Sumber (Ema Utami ; 2015)

II.8.2. Normalisasi

Normalisasi adalah suatu teknik dengan pendekatan *bottom-up* yang digunakan untuk membantu mengidentifikasi hubungan. Dimulai dari menguji hubungan, yaitu *functional dependencies* antara atribut. Pengertian lainnya adalah suatu teknik yang menghasilkan sekumpulan hubungan dengan sifat-sifat yang diinginkan dan memenuhi kebutuhan pada perusahaan. (Indrajani;7:2015)

II.8.2.1. Tujuan Normalisasi

Tujuan utama normalisasi adalah mengidentifikasi kesesuaian hubungan yang mendukung data untuk memenuhi kebutuhan perusahaan. Adapun karakteristik hubungan tersebut mencakup:

1. Minimal jumlah atribut yang diperlukan untuk mendukung kebutuhan perusahaan.
2. Atribut dengan hubungan logika yang menjelaskan mengenai *functional dependencies*.
3. Minimal duplikasi untuk tiap atribut.

II.8.2.2. Proses Normalisasi

Beberapa hal yang perlu diperhatikan dalam proses normalisasi adalah:

1. Suatu teknik formal untuk menganalisa relasi berdasarkan *primary key* dan *functional dependencies* antar atribut.
2. Dieksekusi dalam beberapa langkah. Setiap langkah mengacu ke bentuk normal tertentu, sesuai dengan sifat yang dimilikinya.
3. Setelah normalisasi diproses, relasi menjadi rencana bertahap lebih terbatas atau kuat mengenai bentuk formatnya dan juga mengurangi tindakan *update* yang anomali. (Indrajani;8:2015)

1. Unnormalized Form (UNF)

Merupakan suatu tabel yang berisikan satu atau lebih grup yang berulang. Membuat tabel yang unnormalized, yaitu dengan memindahkan data dari sumber informasi.

Contoh: nota penjualan yang disimpan kedalam format tabel dengan baris dan kolom.

2. First Normal Form (1NF)

Merupakan sebuah relasi dimana setiap baris dan kolom berisikan satu dan hanya satu nilai.

3. Second Normal Form (2NF)

Berdasarkan pada konsep *full functional dependency*, yaitu A dan B merupakan atribut sebuah relasi. B dikatakan *fully dependent* terhadap A jika B *functionally dependent* pada A tetapi tidak pada proper subset dari A. 2NF merupakan sebuah relasi dalam 1NF dan setiap atribut *nonprimary-key* bersifat *fully functionally dependent* pada *primary key*. (Indrajani;9:2015)

4. Third Normal Form (3NF)

Berdasarkan pada konsep *transitive dependency*, yaitu suatu kondisi dimana A, B dan C merupakan atribut sebuah relasi, maka $A \rightarrow B$ dan $B \rightarrow C$, maka *transitively dependent* pada A melalui B (jika A tidak *functionally dependent* pada *primary key*). (Indrajani;10:2015)

Tabel II.3. Tabel Kirim-1 (Unnormal)

<u>No-pem</u>	<u>Kode-kota</u>	<u>Kota</u>	<u>No-bar</u>	<u>Jumlah</u>
P01	1	Jakarta	B01	1000
			B02	1500
			B03	2000
P02	3	Bandung	B03	1000
P03	2	Surabaya	B02	2000

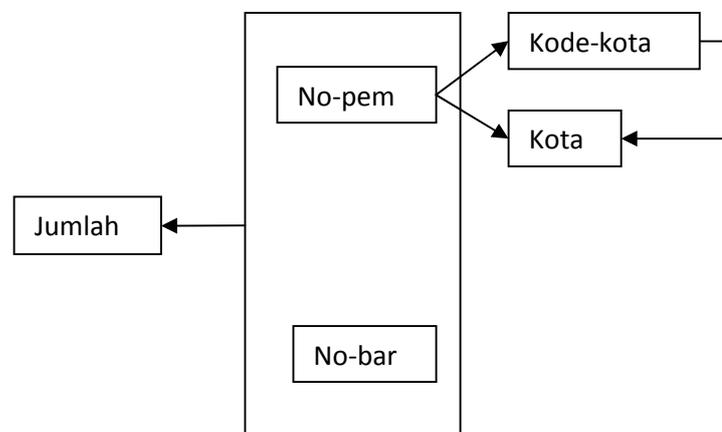
Sumber (Indrajani:2015)

Tabel II.4. Tabel Kirim-2 (1NF)

<u>No-pem</u>	Kode-kota	Kota	<u>No-bar</u>	Jumlah
P01	1	Jakarta	B01	1000
P01	1	Jakarta	B02	1500
P01	1	Jakarta	B03	2000
P02	3	Bandung	B03	1000
P03	2	Surabaya	B02	2000

Sumber (Indrajani:2015)

Diagram Ketergantungan Fungsional

**Gambar II.3. Diagram Ketergantungan Fungsional**

Sumber (Indrajani:2015)

Tabel II.5. Tabel Pemasok-1 (2NF)

<u>No-pem</u>	Kode-kota	Kota
P01	1	Jakarta
P02	3	Bandung
P03	2	Surabaya

Sumber (Indrajani:2015)

Tabel II.6. Tabel Kirim-3 (3NF)

<u>No-pem</u>	<u>No-bar</u>	Jumlah
P01	B01	1000
P01	B02	1500
P01	B03	2000
P02	B03	1000
P03	B02	2000

Sumber (Indrajani:2015)

Tabel II.7. Tabel Pemasok-2 (3NF)

<u>No-pem</u>	Kode-kota
P01	1
P02	3
P03	2

Sumber (Indrajani:2015)

Tabel II.8. Tabel Pemasok-3 (3NF)

<u>Kode-kota</u>	Kota
1	Jakarta
2	Surabaya
3	Bandung

Sumber (Indrajani:2015)

II.9. Website

Website (Situs *Web*) merupakan kumpulan dari halaman-halaman *web* yang berhubungan dengan *file-file* lain yang terkait. Dalam sebuah *website* terdapat suatu halaman yang dikenal dengan sebutan *home page*. *Home page* adalah sebuah halaman yang pertama kali dilihat ketika seseorang mengunjungi

website. Dari *home page*, pengunjung dapat mengklik *hyperlink* untuk pindah kehalaman lain yang terdapat dalam website tersebut.

Secara umum *website* mempunyai fungsi sebagai berikut :

1. Fungsi Komunikasi

Beberapa fasilitas yang memberikan fungsi komunikasi, seperti : *chatting*, *web base email* dan lain-lain.

2. Fungsi Informasi

Fungsi informasi *website* seperti : *News*, *Profile*, *Library*, referensi dan lain-lain.

3. Fungsi *Intertainment*

Website mempunyai fungsi hiburan. Misalnya *web-web* yang menyediakan *game on-line*, *music on-line* dan lain-lain. (Yoni Widhiarso ; 2013 : 3)

II. 10. UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan

piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*). Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: *metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, metodologi wirfs-brock*, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek.

Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan. Dimulai pada bulan Oktober 1994 *Booch, Rumbaugh dan Jacobson*, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease *draft* pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group* (OMG –

<http://www.omg.org>). Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. *Booch, Rumbaugh* dan *Jacobson* menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek. (*Yuni Sugiarti ; 2013 : 33*)

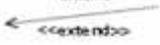
Dalam pembuatan skripsi ini penulis menggunakan diagram *Use Case* yang terdapat di dalam UML. Adapun maksud dari *Use Case* Diagram diterangkan dibawah ini.

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat

dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. (Yuni Sugiarti ; 2013 : 41)

Tabel II.9. Use Case Diagram

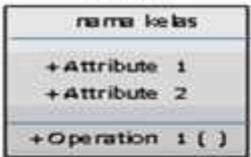
Simbol	Deskripsi
 <p>Use Case</p>	fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case
 <p>Aktor</p>	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
<p>Asosiasi / association</p> <hr/>	komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
 <p>Extend</p>	relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, arah panah menunjukan pada use case yang dituju 
 <p>Include</p>	relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini. Ada dua sudut pandang yang cukup besar mengenai include di use case, include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan, contoh : 

Sumber : (Yuni Sugiarti ; 2013 ; 42)

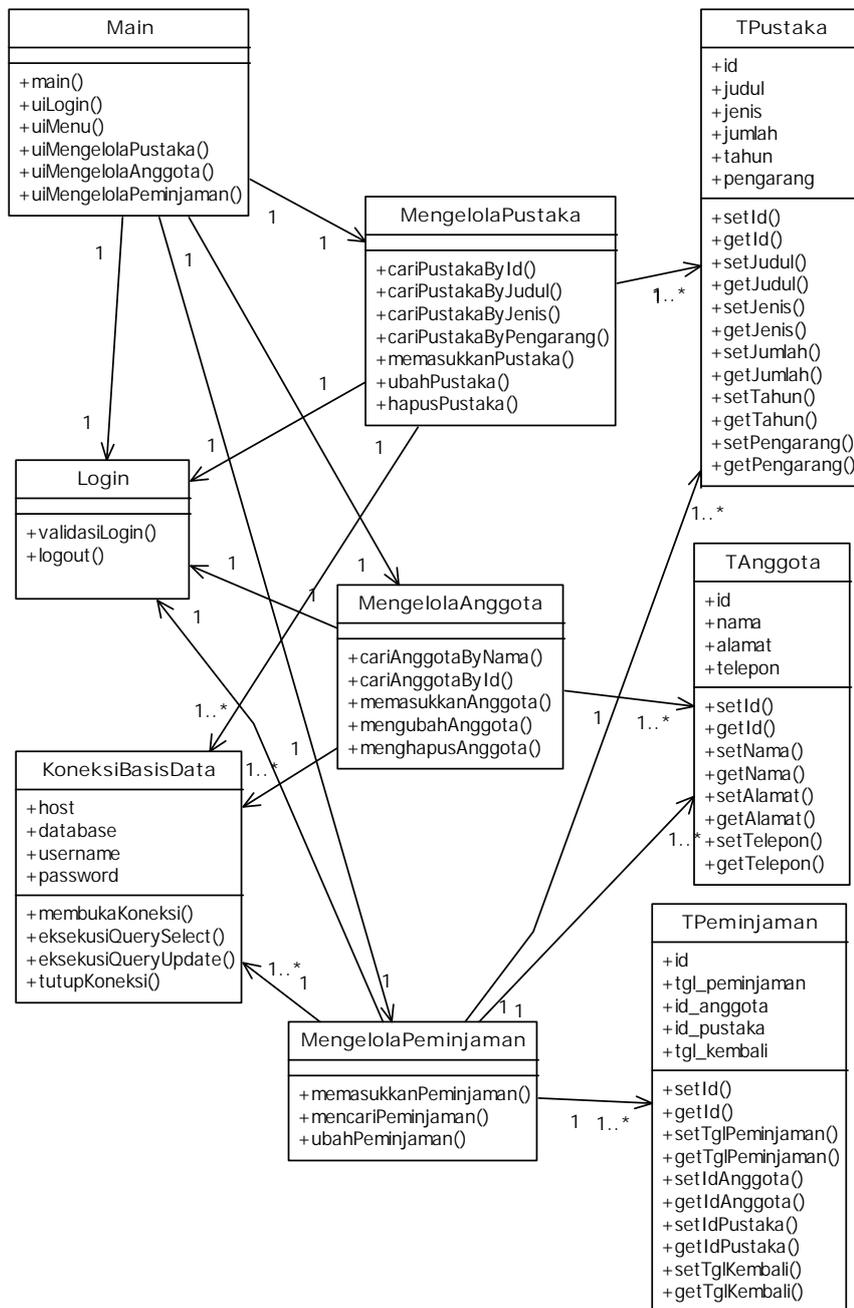
2. Class Diagram

Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol-simbol pada diagram kelas :

Tabel II.10. *Class Diagram*

Simbol	Deskripsi
<p>Package</p> 	Package merupakan sebuah bungkus dari satu atau lebih kelas
<p>Operasi</p> 	Kelas pada struktur sistem
<p>Antarmuka / interface</p> 	sama dengan konsep interface dalam pemrograman berorientasi objek
<p>Asosiasi</p> 	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
<p>Asosiasi berarah/directed asosiasi</p> 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
<p>Generalisasi</p> 	relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
<p>Kebergantungan / defedency</p> 	relasi antar kelas dengan makna kebergantungan antar kelas
<p>Agregasi</p> 	relasi antar kelas dengan makna semua-bagian (whole-part)

Sumber : (Yuni Sugiarti ; 2013 : 59)



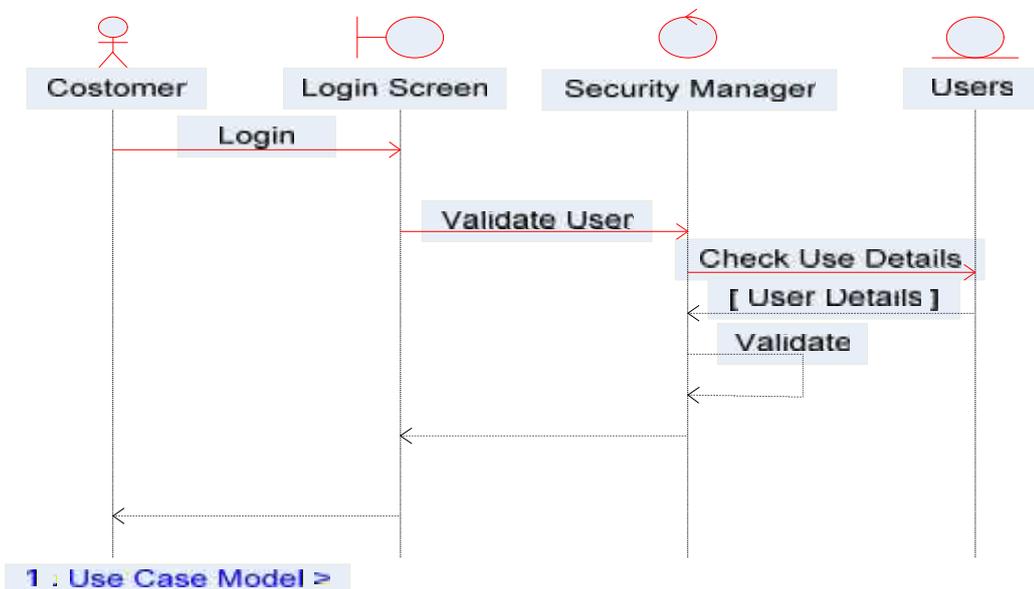
Gambar II.4. Contoh Class Diagram

Sumber : (Yuni Sugiarti ; 2013 : 63)

3. *Sequence Diagram*

Diagram *Sequence* menggambarkan kelakuan/prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram *sequence* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak.



Gambar II.5. Contoh *Sequence Diagram*

Sumber : (Yuni Sugiarti ; 2013 : 63)

4. *Activity Diagram*

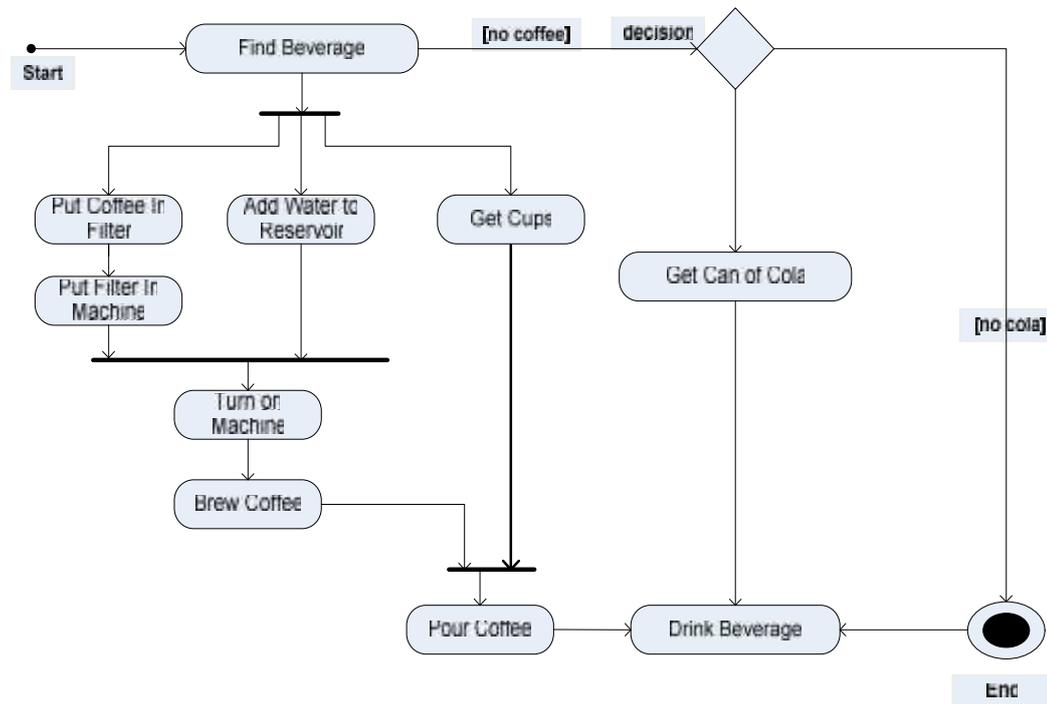
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



Gambar II.6. Activity Diagram

Sumber : (Yuni Sugiarti ; 2013 : 76)