

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

II.1.1. Sistem

Sistem adalah sekumpulan komponen yang saling berhubungan, bekerja sama untuk mencapai suatu tujuan dengan menerima input serta menghasilkan output dalam proses transformasi yang teratur. (Kusrini; 2008: 4)

Sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional (dengan tugas/fungsi khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses/pekerjaan tertentu. Sebagai contoh, sistem kendaraan terdiri dari: komponen starter, komponen pengapian, komponen penggerak, komponen pengerem, komponen kelistrikan-spedometer, lampau dan lain-lain. Komponen-komponen tersebut diatas memiliki tujuan yang sama yaitu untuk membuat kendaraan tersebut bisa dikendarai dengan nyaman dan aman. Contoh lain yaitu sistem perguruan tinggi, yang terdiri dari dosen, mahasiswa, kurikulum, dan lain-lain. Sistem ini bertujuan untuk menghasilkan mahasiswa-mahasiswa yang memiliki kemampuan di bidang ilmunya. (Kusrini; 2008: 4).

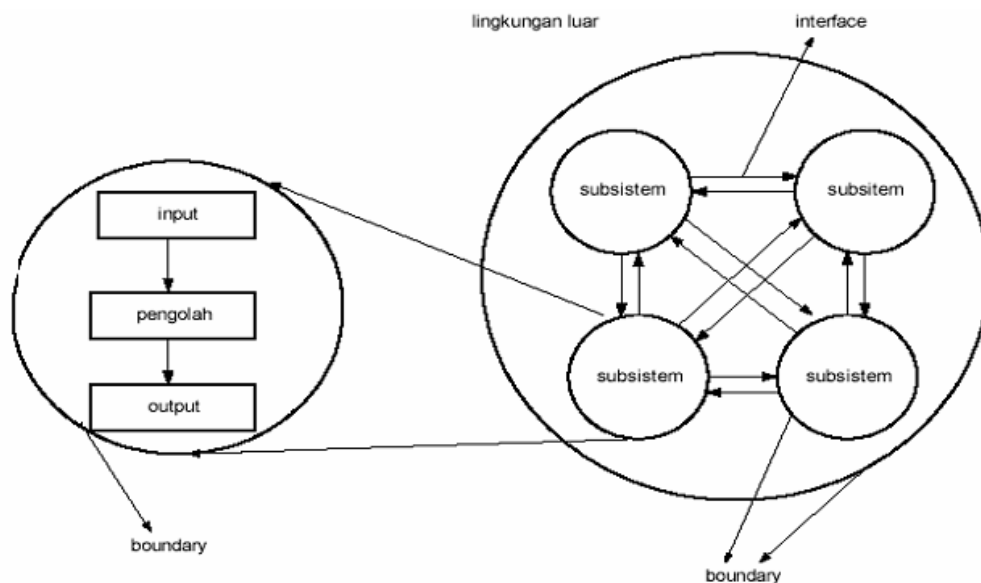
Kesimpulannya, Sistem merupakan sekumpulan komponen/elemen yang saling berinteraksi untuk mencapai tujuan tertentu.

II.1.2. Konsep Dasar Sistem

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Menurut Jerry FithGerald ; sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. (Kusrini; 2008: 4)

II.1.2.1. Karakteristik Sistem

Karakteristik sistem dapatlah digambarkan sebagai berikut :



Gambar II.1. Karakteristik Sistem

(Sumber : Kusrini; 2008:5)

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu

subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Jadi, dapat dibayangkan jika dalam suatu sistem ada subsistem yang tidak berjalan/berfungsi sebagaimana mestinya. Tentunya sistem tersebut tidak akan berjalan mulus atau mungkin juga sistem tersebut rusak sehingga dengan sendirinya tujuan sistem tersebut tidak tercapai.

2. Batas Sistem (*Boundary*)

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environments*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung (*Interface*)

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke yang lainnya. Keluaran (output) dari satu subsistem akan menjadi masukan (input) untuk subsistem lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (signal input). Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah maintenance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

6. Keluaran (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada super sistem. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

7. Pengolah (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan lain yang dibutuhkan oleh manajemen.

8. Sasaran (*Objectives*) atau Tujuan (*Goal*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Perbedaan suatu sasaran (*objectives*) dan suatu tujuan (*goal*) adalah, *goal* biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran dalam ruang lingkup yang lebih sempit. Bila merupakan suatu sistem utama, seperti misalnya sistem bisnis perusahaan, maka istilah *goal* lebih tepat diterapkan. Untuk sistem akuntansi atau sistem-sistem lainnya yang merupakan bagian atau subsistem dari sistem bisnis, maka istilah *objectives* yang lebih tepat. Jadi tergantung dari ruang lingkup mana memandang sistem tersebut. Seringkali tujuan (*goal*) dan sasaran (*objectives*) digunakan bergantian dan tidak dibedakan. (Kusrini; 2008: 8)

II.1.3. Sistem Informasi

II.1.3.1. Informasi

Informasi adalah data yang dapat dianalogikan dengan data – data, yang belum di kelolah dan harus diolah untuk menjadi informasi yang akurat. Agar informasi yang penulis sajikan lebih bermanfaat maka terlebih dahulu dibuat aliran informasi yang lebih jelas dan lengkap. Berkaitannya dengan penyedia informasi bagi manajemen dalam mengambil suatu keputusan, yang diperoleh harus berkualitas, maka kualitas dari informasi tergantung :

1. Akurat

Akurat berarti bahwa informasi harus bebas dari kesalahan - kesalahan dan tidak biasa (menyesatkan) dan jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerimaan informasi kemungkinan banyak terjadi gangguan yang dapat merubah informasi atau merusak informasi tersebut.

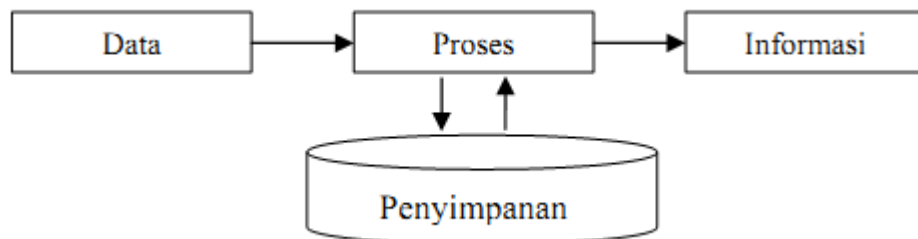
2. Relevansi

Relevansi berarti bahwa informasi benar – benar berguna bagi suatu tindakan dan keputusan oleh seseorang

3. Tepat waktu

Tepat waktu berarti bahwa informasi yang datang pada penerimaan tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan dalam pengambilan keputusan. Dewasa ini mahalnya nilai informasi disebabkan harus cepatnya informasi itu di dapat,

Untuk lebih jelasnya informasi merupakan hasil atau output dari proses informasi data (Kusrini; 2008: 15) . Hal ini dapat dilihat seperti gambar II.2



Gambar II.2. Proses Data Menjadi Informasi

(Sumber : Kusrini; 2008:15)

II.2. Sistem Informasi Geografis

II.2.1. Geography

Geografi adalah ilmu tentang lokasi serta persamaan dan perbedaan (variasi) keruangan atas fenomena fisik dan manusia di atas permukaan bumi. Kata geografi berasal dari Bahasa Yunani yaitu ge ("Bumi") dan graphein ("menulis", atau "menjelaskan"). Geografi juga merupakan nama judul buku bersejarah pada subyek ini, yang terkenal adalah Geographia tulisan Klaudios Ptolemaios (abad kedua). Geografi lebih dari sekedar kartografi, studi tentang peta. Geografi tidak hanya menjawab apa dan dimana di atas muka bumi, tapi juga mengapa di situ dan tidak di tempat lainnya, kadang diartikan dengan "lokasi pada ruang." Geografi mempelajari hal ini, baik yang disebabkan oleh alam atau manusia. Juga mempelajari akibat yang disebabkan dari perbedaan yang terjadi itu. Istilah ini digunakan karena GIS dibangun berdasarkan pada 'geografi' atau 'spasial'. (Eko Budiyanto; 2009 : 1)

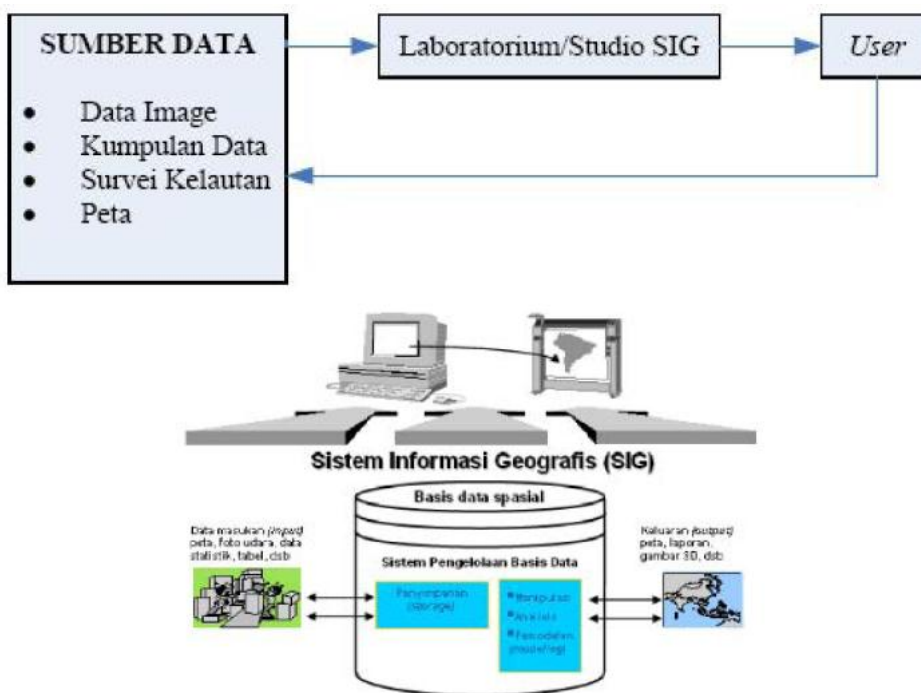
Dalam GIS, informasi memiliki volume terbesar. Setiap object geografi memiliki setting data tersendiri karena tidak sepenuhnya data yang ada dapat terwakili dalam peta. Jadi, semua data harus diasosiasikan dengan objek spasial yang dapat membuat peta menjadi intelligent. Ketika data tersebut diasosiasikan dengan permukaan geografis yang representatif, data tersebut mampu memberikan informasi dengan hanya mengklik mouse pada objek. Perlu diingat bahwa semua informasi adalah data tapi tidak semua data merupakan informasi. (Eko Budiyanto; 2009 : 2)

II.2.2. Geographical information system (GIS)

Sistem informasi berbasis pemetaan dan geografis adalah sebuah alat bantu manajemen berupa informasi berbantuan komputer yang berkait erat dengan sistem pemetaan dan analisis terhadap segala sesuatu serta peristiwa-peristiwa yang terjadi permukaan bumi. Teknologi GIS mengintegrasikan operasi pengolahan data berbasis database yang digunakan saat ini pengambilan data berdasarkan kebutuhan, serta analisa statistik dengan menggunakan visualisasi yang khas berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar petanya. Teknologi Sistem Informasi Geografis dapat digunakan untuk investigasi ilmiah, pengelolaan sumber daya, perencanaan pembangunan, kartografi dan perencanaan rute. Misalnya, SIG bisa membantu perencana untuk secara cepat menghitung waktu tanggap darurat saat terjadi bencana alam, atau SIG dapat digunakan untuk mencari lahan basah (wetlands) yang membutuhkan perlindungan dari polusi. (Eko Budiyanto; 2009 : 6-7)

II.2.3. Konsep Sistem Informasi Geografis

Sumber data untuk keperluan SIG dapat berasal dari data citra, data lapangan, survey kelautan, peta, sosial ekonomi, dan GPS. Selanjutnya diolah di laboratorium atau studio SIG dengan software tertentu sesuai dengan kebutuhannya untuk menghasilkan produk berupa informasi yang berguna, bisa berupa peta konvensional, maupun peta digital sesuai keperluan user, maka harus ada input kebutuhan yang diinginkan user, dapat dilihat pada gambar II.3



Gambar II.3. Konsep Sistem Informasi Geografis

(Sumber : Eko Budiyanto; 2009:8-9)

Untuk mendukung suatu Sistem Informasi Geografis, pada prinsipnya terdapat dua jenis data, yaitu:

1. Data spasial

Data yang berkaitan dengan aspek keruangan dan merupakan data yang menyajikan lokasi geografis atau gambaran nyata suatu wilayah di permukaan bumi. Umumnya direpresentasikan berupa grafik, peta, atau pun gambar dengan format digital dan disimpan dalam bentuk koordinat x,y (vektor) atau dalam bentuk image (raster) yang memiliki nilai tertentu.

2. Data non-spasial

Disebut juga data atribut, yaitu data yang menerangkan keadaan atau informasi-informasi dari suatu objek (lokasi dan posisi) yang ditunjukkan oleh data spasial. Salah satu komponen utama dari Sistem Informasi Geografis adalah perangkat lunak (software). (Eko Budiyo; 2009 :9)

II.2.3. Komponen Sistem Informasi Geografis

Secara umum, Sistem Informasi Geografis bekerja berdasarkan integrasi komponen, yaitu: Hardware, Software, Data, Manusia, dan Metode. Kelima komponen tersebut dapat dijelaskan sebagai berikut:

- a. Hardware.

Sistem Informasi Geografis memerlukan spesifikasi komponen hardware yang sedikit lebih tinggi dibanding spesifikasi komponen sistem informasi lainnya. Hal tersebut disebabkan karena data-data yang digunakan dalam SIG, penyimpanannya membutuhkan ruang yang besar dan dalam proses

analisnya membutuhkan memory yang besar dan processor yang cepat. Beberapa Hardware yang sering digunakan dalam Sistem Informasi Geografis adalah: Personal Computer (PC), Mouse, Digitizer, Printer, Plotter, dan Scanner. (Eko Budiyo; 2009 :10)

b. Software

Sebuah software SIG haruslah menyediakan fungsi dan tool yang mampu melakukan penyimpanan data, analisis, dan menampilkan informasi geografis. Dengan demikian elemen yang harus terdapat dalam komponen software SIG adalah:

- a. Tools untuk melakukan input dan transformasi data geografis
- b. Sistem Manajemen Basis Data.
- c. Tools yang mendukung query geografis, analisis, dan visualisasi.
- d. Geographical User Interface (GUI) untuk memudahkan akses pada tool geografi.

c. Data

Hal yang merupakan komponen penting dalam SIG adalah data. Secara fundamental, SIG bekerja dengan 2 tipe model data geografis, yaitu model data vector dan model data raster. Dalam model data vector, informasi posisi point, garis, dan polygon disimpan dalam bentuk koordinat x,y. Bentuk garis, seperti jalan dan sungai dideskripsikan sebagai kumpulan dari koordinat-koordinat point. Bentuk polygon, seperti daerah penjualan disimpan sebagai pengulangan koordinat yang tertutup. Data raster terdiri dari sekumpulan grid atau sel seperti peta hasil scanning maupun gambar

atau image. Masing-masing grid memiliki nilai tertentu yang bergantung pada bagaimana image tersebut digambarkan.

d. Manusia

Komponen manusia memegang peranan yang sangat menentukan, karena tanpa manusia maka sistem tersebut tidak dapat diaplikasikan dengan baik. Jadi manusia menjadi komponen yang mengendalikan suatu sistem sehingga menghasilkan suatu analisa yang dibutuhkan.

e. Metode

SIG yang baik memiliki keserasian antara rencana desain yang baik dan aturan dunia nyata, dimana metode, model dan implementasi akan berbeda untuk setiap permasalahan. (Eko Budiyanto; 2009 :11)



Gambar II.4. Komponen Sistem Informasi Geografis

(Sumber : Eko Budiyanto; 2009:11)

II.3. PHP CodeIgniter

PHP disebut script atau bisa juga disebut program, tapi umumnya disebut script. File PHP tidak dikompilasi, tapi diinterpretasi oleh server (misal apache ataupun IIS). PHP tidak umum dipakai untuk aplikasi web. PHP populer karena memiliki kode yang sangat mirip C atau C++. Relatif lebih mudah daripada Perl. Dan konfigurasi relatif lebih mudah daripada Python. Anda sudah terbiasa dengan C, C++, ataupun Java akan dengan mudah melompat ke PHP.

Posisi PHP sebagai bahasa scripting di dunia web sangat kuat, bahasa ini memiliki apresiasi dari komunitas web. Kita bisa jumpai banyak teknologi MVC (Model-view-Controller) untuk PHP yang salah satunya CodeIgniter. Kita bisa jumpai berbagai ORM tools, yang terkenal adalah Doctrine. Kita juga bisa menemukan templating engine seperti Twig. (Adam Pahlevi; 2013 : viii)

Jika ini adalah pertama kali kita membuat proyek dalam PHP, maka proses instalasi tidak akan berjalan terlalu lama. Proyek dalam webmatrix otomatis akan dieksekusi oleh server IIS jika kita mengklik tombol “start” maupun “run”. URL di atas adalah URL localhost, yang merupakan loophole ke komputer kita sendiri. Jika kita mengklik URL, maka safari akan terbuka namun tidak menampilkan apa pun karena kita belum melakukan apapun. Sekarang, klik “Files” pada tab di bawah. Klik pada “index.php” ketik kode berikut :

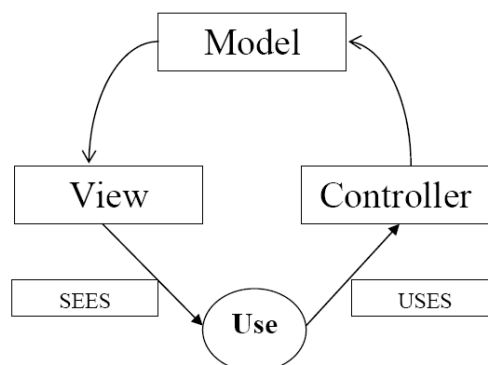
```
“<? PHP
Echo”Hello World”;
?>
```

Sekarang klik tobo, “Run” pada toolbar. Safari akan dibuka dan anda akan melihat “hello world” (Adam Pahlevi; 2013 : 11)

II.3.1. Model View Controller PHP

Model-View-Controller. MVC adalah sistem pengembangan aplikasi dalam mode 3 tier. MVC digunakan untuk memisahkan responsibility dari masing-masing komponen. Jadi, akan ada komponen yang urusannya Cuma masalah tampilan, ada juga yang Cuma masalah koneksi ke database, setiap komponen memiliki tanggung jawabnya masing-masing. Tidak masalah jika pada awalnya terdengar abstrak, memang kenyataannya konsep MVC akan lebih masuk akal jika dipelajari secara praktikal. MVC dibagi menjadi 3 komponen sebagai berikut :

1. Controller : menentukan apa yang akan dikerjakan
2. Model : melakukan manipulasi data, menghasilkan data dan apapun yang ada hubungannya dengan data, kecuali menampilkan data.
3. View : menampilkan data. (Adam Pahlevi; 2013 : 18)



Gambar II.5. Komponen Sistem Informasi Geografis

(Sumber : Adam Pahlevi; 2013:18)

Inilah salah satu bagian dari dunia berorientasi objek. Buku ini berharap untuk bisa membahas pengaplikasian pemrograman berbasis object, menggunakan mvc di PHP dengan framework CodeIgniter.

Pertanyaan mungkin muncul, “Bisakah tanpa menggunakan Frame-work bantuan, kita melakukan MVC di PHP secara native, menjadi argumen di atas tidak relevan lagi. (Adam Pahlevi; 2013 : 19)

II.4. Sekilas Tentang MySQL

II.4.1. Pengenalan Database, DBMS dan RDBMS

Basis data (atau database) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut (Achmad Solihin; 2010 : 6)

Database digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer.

Untuk mengelola database diperlukan suatu perangkat lunak yang disebut DBMS (Database Management System). DBMS merupakan suatu sistem perangkat lunak yang memungkinkan user (pengguna) untuk membuat, memelihara, mengontrol, dan mengakses database secara praktis dan efisien. Dengan DBMS, user akan lebih mudah mengontrol dan memanipulasi data yang ada.

Sedangkan RDBMS atau Relationship Database Management System merupakan salah satu jenis DBMS yang mendukung adanya relationship atau

hubungan antar tabel. Di samping RDBMS, terdapat jenis DBMS lain, misalnya Hierarchy DBMS, Object Oriented DBMS, dan sebagainya.(Achmad Solihin; 2010 : 6)

1. Beberapa Istilah Database.

a. Table

Sebuah tabel merupakan kumpulan data (nilai) yang diorganisasikan ke dalam baris (record) dan kolom (field). Masing-masing kolom memiliki nama yang spesifik dan unik.

b. Field

Field merupakan kolom dari sebuah table. Field memiliki ukuran type data tertentu yang menentukan bagaimana data nantinya tersimpan.

c. Record

Field merupakan sebuah kumpulan nilai yang saling terkait.

d. Key

Key merupakan suatu field yang dapat dijadikan kunci dalam operasi tabel. Dalam konsep database, *key* memiliki banyak jenis diantaranya *Primary Key*, *Foreign Key*, *Composite Key*, dan lain-lain.

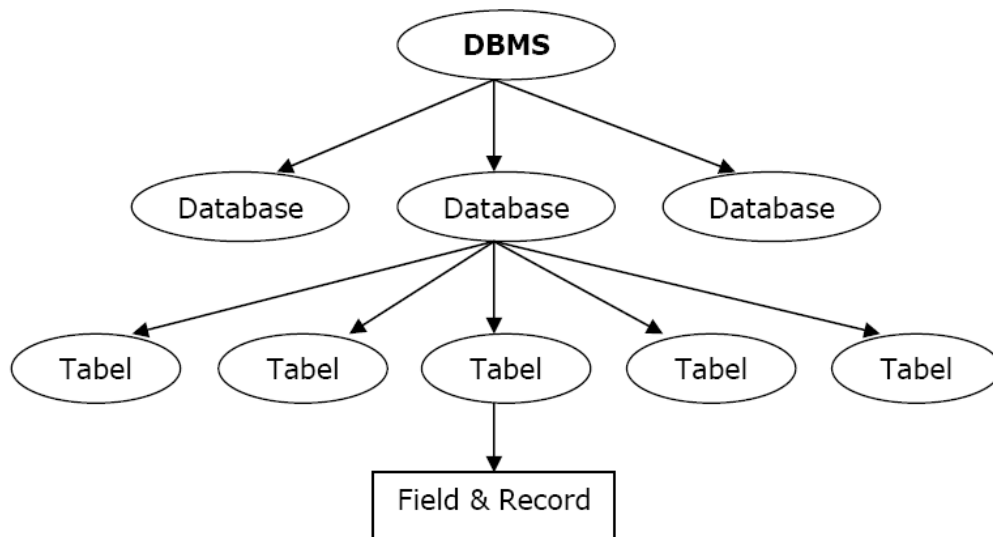
e. SQL

SQL atau Structured Query Language merupakan suatu bahasa (language) yang digunakan untuk mengakses database. SQL sering disebut juga sebagai query. (Achmad Solihin; 2010 : 7)

2. Hierarki Database

Dalam konsep database, urutan atau hierarki database sangatlah penting.

Urutan atau hierarki database digambarkan seperti pada gambar II.6



Gambar II.6. Hirarki Database

(Sumber : Achmad Solihin; 2010:7)

II.4.2. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak seperti PHP atau Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya

masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. (Achmad Solihin; 2010 : 8)

1. Fitur – Fitur MySQL

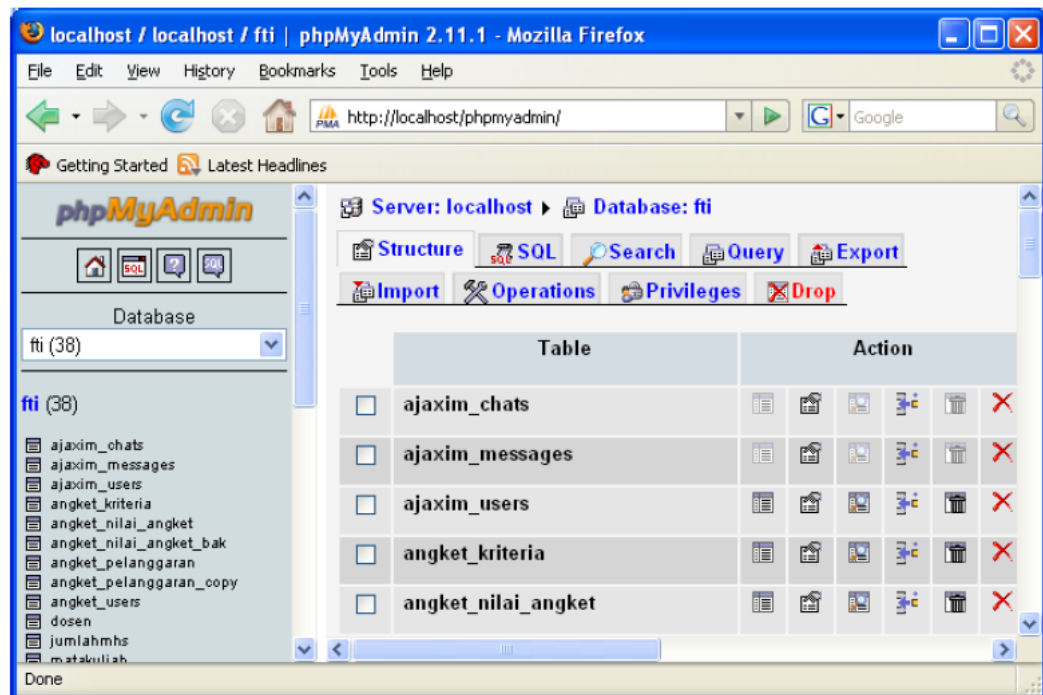
MySQL mempunyai beberapa fitur diantaranya adalah :

- a. Relational Database System. Seperti halnya software database lain yang ada di pasaran, MySQL termasuk RDBMS.
- b. Arsitektur Client-Server. MySQL memiliki arsitektur client-server dimana server database MySQL terinstal di server. Client MySQL dapat berada di komputer yang sama dengan server, dan dapat juga di komputer lain yang berkomunikasi dengan server melalui jaringan bahkan internet.
- c. Mengenal perintah SQL standar. SQL (Structured Query Language) merupakan suatu bahasa standar yang berlaku di hampir semua software database. MySQL mendukung SQL versi SQL: 2003.
- d. Mendukung Sub Select. Mulai versi 4.1 MySQL telah mendukung select dalam select (sub select).
- e. Mendukung Views. MySQL mendukung views sejak versi 5.0
- f. Mendukung Stored Prosedured (SP). MySQL mendukung SP sejak versi 5.0

- g. Mendukung Triggers. MySQL mendukung trigger pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan trigger pada versi 5.1.
- h. Mendukung replication.
- i. Mendukung transaksi.
- j. Mendukung foreign key.
- k. Tersedia fungsi GIS.
- l. Free (bebas didownload)
- m. Stabil dan tangguh
- n. Fleksibel dengan berbagai pemrograman
- o. Security yang baik (Achmad Solihin; 2010 : 9)

II.4.2.1. PHPMyAdmin

PHPMyAdmin merupakan front-end MySQL berbasis web. PHPMyAdmin dibuat dengan menggunakan PHP. Saat ini, PHPMyAdmin banyak digunakan dalam hampir semua penyedia hosting yang ada di internet. PHPMyAdmin mendukung berbagai fitur administrasi MySQL termasuk manipulasi database, tabel, index dan juga dapat mengekspor data ke dalam berbagai format data. PHPMyAdmin juga tersedia dalam 50 bahasa lebih, termasuk bahasa Indonesia. (Achmad Solihin; 2010 : 22-23).



Gambar II.7. Tampilan halaman PHPMyAdmin

(Sumber : Achmad Solihin; 2010:23)

II.5. UML (*Unified Modelling Language*)

UML singkatan dari Unified Modeling Language yang berarti bahasa pemodelan standar. (Chonoles; 2003: bab 1) mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model – model yang kita buat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya ? Bagaimana sistem mengatasi error yang terjadi ? Bagaiman

kemanan terhadap sistem yang kita buat ? Dan sebagainya dapat dijawab dengan UML. (Prabowo Pudjo Widodo & Herlawati; 2011: 6)

UML dapat diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

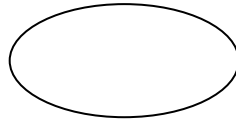
II.5.1. Diagram – Diagram UML

Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis.(Prabowo Pudjo Widodo & Herlawati; 2011: 10)

1. Use Case Diagram

Menurut (Pilone; 2005 : bab 7.1) use case menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan (Whitten; 2004: 258) mengartikan use case sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. Use case

digambarkan dalam bentuk ellips/oval. (Prabowo Pudjo Widodo & Herlawati; 2011: 21)



Gambar II.8. Simbol Use Case

(Sumber : Prabowo Pudjo Widodo & Herlawati; 2011: 22)

Elemen use case terdiri dari :

- a. Diagram use case, disertai dengan narasi dan skenario.
- b. Aktor (actor), mendefinisikan entitas diluar sistem yang memakai sistem.
- c. Asosiasi (assosiations), mengindikasikan aktor mana yang berinteraksi dengan use case dalam suatu sistem.
- d. << include >> dan << extend >>. Merupakan indikator yang menggambarkan jenis relasi dan interaksi antar use-case.
- e. Generalisasi (generalization), menggambarkan hubungan turunan antar use case atau antar aktor. (Prabowo Pudjo Widodo & Herlawati; 2011:35)

2. Diagram Kelas (Class Diagram)

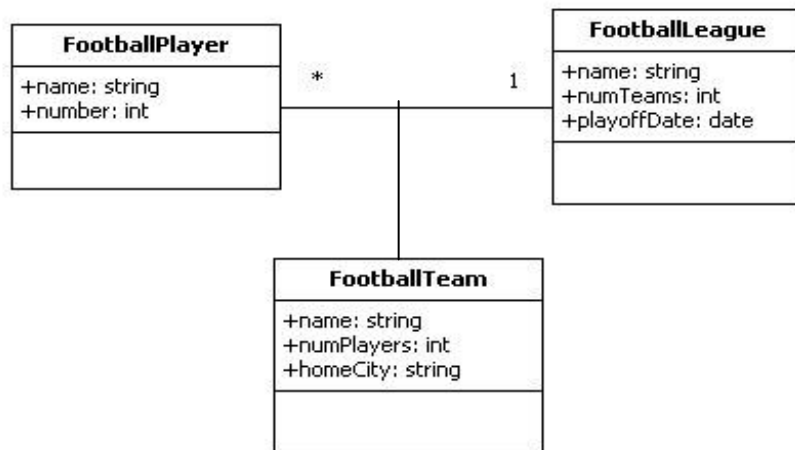
Diagram kelas adalah inti dari proses pemodelan objek. Baik forward engineering maupun reverse engineering memanfaatkan diagram ini. Forward engineering adalah proses perubahan model menjadi kode program sedangkan reverse engineering sebaliknya merubah kode program menjadi model. (Prabowo Pudjo Widodo & Herlawati; 2011: 37).

Diagram kelas merupakan kumpulan kelas-kelas objek. Oleh karena itu pengertian kelas sangat penting sebelum merancang diagram kelas. (Whitten; 2004: 410) mengartikan kelas sebagai satu set objek yang memiliki atribut kelas objek (objek class). Secara alami, objek yang berupa buku analisis

desain dan buku pemrograman terstruktur kita kelompokkan dalam satu kelas, yaitu kelas buku. Kedua objek memiliki atribut dan perilaku yang serupa. Contohnya, kedua objek mungkin memiliki atribut yang serupa seperti nomor ISBN, Judul, tanggal penerbitan, edisi, dan sebagainya. Demikian juga, kedua objek memiliki perilaku yang sama misalnya membuka dan menutup. (Prabowo Pudjo Widodo & Herlawati; 2011: 39).

Kelas menggambarkan suatu group yang memiliki kesamaan dan perilaku. Kelas merupakan cetak biru suatu objek dalam suatu sistem orientasi objek. Dapat dikatakan kelas adalah sejenis alat pengklasifikasikan. Sebagai contoh volkswag, toyota dan ford merupakan kumpulan mobil sehingga kita dapat mengelompokkannya dalam menyatakan konsep yang dapat dilihat maupun abstrak.

Kelas dinyatakan dalam kotak yang terbagi menjadi beberapa kompartemen. Kompartemen adalah area dalam kelas yang berisi informasi. Kompartemen pertama berisikan nama kelas, berikutnya atribut dan yang terakhir operasinya. Kompartemen-kompartemen itu bisa disembunyikan untuk memperjelas diagram yang kita buat. (Prabowo Pudjo Widodo & Herlawati; 2011: 40)



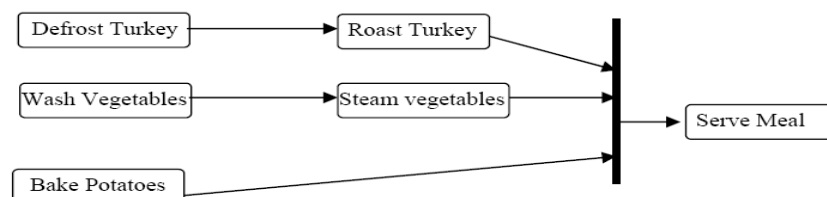
Gambar II.9. Contoh Class Diagram

(Sumber : Prabowo Pudjo Widodo & Herlawati; 2011: 69)

3. Activity diagram

Diagram aktifitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan model bisnis juga. Diagram aktifitas menunjukkan aktifitas sistem dalam bentuk aksi-aksi. Ketika digunakan dalam pemodelan software, diagram aktifitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya call. Sedangkan bila dimodelkan dalam pemodelan bisnis, diagram ini menggambarkan aktifitas yang dipicu oleh kejadian-kejadian di luar seperti pemesanan atau kejadian-kejadian internal. (Prabowo Pudjo Widodo & Herlawati; 2011: 143)

Prepare Meal



Gambar II.10. Contoh Activity Diagram Sederhana

(Sumber : Prabowo Pudjo Widodo & Herlawati; 2011: 163)

4. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sebuah contoh objek dan pesan yang diletakkan diantara objek-objek ini didalam *use case*. Diagram sequence dimaksudkan untuk mengembangkan komunikasi antara objek, bukan memanipulasi data saat berkomunikasi. (Prabowo Pudjo Widodo & Herlawati; 2011: 174)

II.6. Entity Relationship Diagram – ERD

II.6.1. Model-model Data

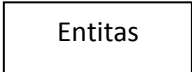
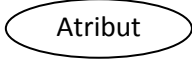
Struktur yang mendasari suatu basisdata adalah model data yang merupakan kumpulan alat-alat konseptual untuk mendeskripsikan data, relasi data, data semantik, dan batasan konsistensi. Untuk mengilustrasikan konsep model data, berikut disajikan dua model data, yaitu *entity relationship model* dan *relational model*. Kedua model menyediakan cara mendeskripsikan rancangan basisdata pada tingkatan logis. (Janner Simarmata & Imam Prayudi; 2006:59)

II.6.2. 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.. (Janner Simarmata & Imam Prayudi; 2006:59)

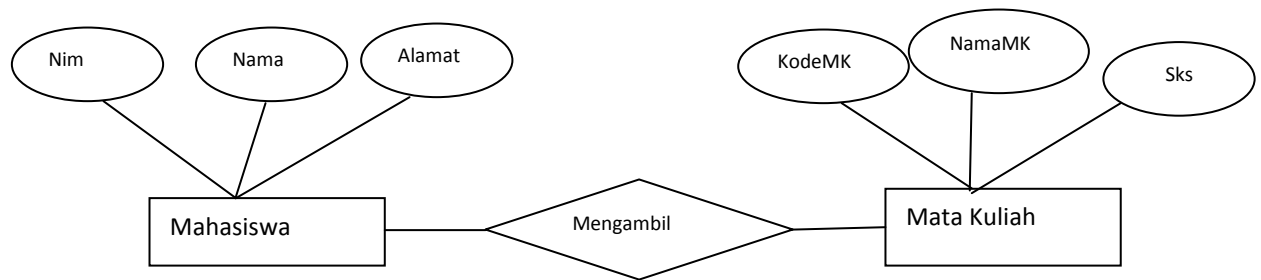
Struktur logis (skema *database* dapat ditunjukkan secara grafis dengan ER yang dibentuk dari komponen-komponen dapat dilihat pada tabel II.1 berikut ini :

Tabel II.1 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 & Imam Prayudi; 2006: 60)

Sebagai ilustrasi, bayangan dan mengambil bagian system basis data universitas yang terdiri dari mahasiswa dan matakuliah. gambar II.11 menunjukkan ER diagram daricontoh. Diagram menunjukkan bahwa ada dua kumpulan entitas, yaitu mahasiswa dan matakuliah, dan bahwa relasi mengambil mahasiswa dan matakuliah.



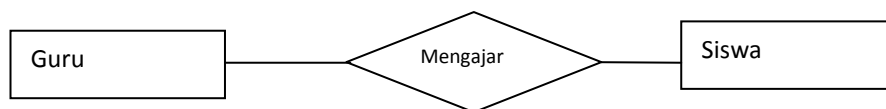
GambarII.11 : Diagram ER

(Sumber : Janner Simarmata dan Iman Prayudi : 2006 : 60)

Ada beberapa model Entitas yaitu :

1. One-to-One

Sebuah entitas pada A berhubungan dengan entitas pada B.

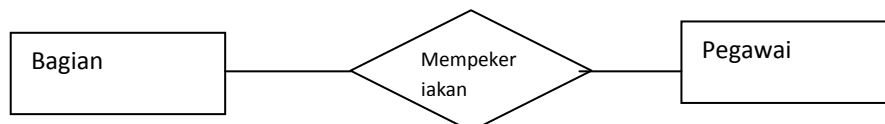


GambarII.12 :Entitas One-to-One

(Sumber : Janner Simarmata dan Iman Prayudi : 2006 : 64)

2. One-to-Many

Sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B.

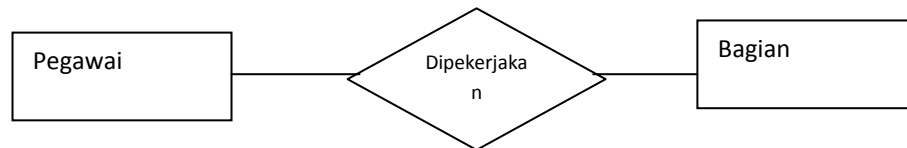


GambarII.13 :Entitas One-to-Many

(Sumber : Janner Simarmata dan Iman Prayudi : 2006 : 65)

3. Many-to-One

Sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B.

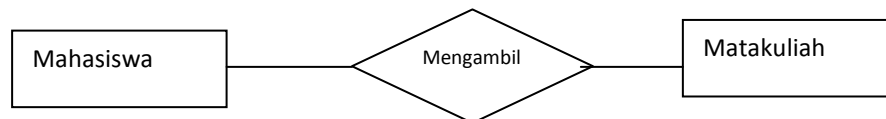


GambarII.14 :EntitasMany-to-One

Sumber : (Janner Simarmata dan Iman Prayudi : 2006 : 65)

4. Many-to-Many

Sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B dan sebuah entitas pada B dapat dihubungkan nol atau lebih entitas pada A.



GambarII.15 :EntitasMany-to-Many

(Sumber : Janner Simarmata dan Iman Prayudi : 2006 : 66)

II.7. Normalisasi

Proses normalisasi merupakan proses pengelompokan data kedalam bentuk table atau relasi atau file untuk menyatakan entitas dan hubungan sehingga terwujud satu bentuk basis data yang mudah dimodifikasi (Janner Simarmata, 2007 : 197)

Terdapat beberapa langkah normalisasi diantaranya :

1. Bentuk Normal Pertama (1NF)

Sebuah tabel relasional secara definisi selalu berada dalam bentuk normal pertama. Sehingga nilai pada kolom-kolomnya adalah *atomic*. Ini berarti kolom-kolom tidak mempunyai nilai berulang.

II.2 Tabel Bentuk Normal Pertama (1NF)

PEMASOK

p#	Status	Kota	b#	Qty
p1	20	Yogyakarta	b1	300
p1	20	Yogyakarta	b2	200
p1	20	Yogyakarta	b3	400
p1	20	Yogyakarta	b4	200
p1	20	Yogyakarta	b5	100
p1	20	Yogyakarta	b6	100
p2	10	Medan	b1	300
p2	10	Medan	b2	400
p3	10	Medan	b2	200
p4	20	Yogyakarta	b2	200
p4	20	Yogyakarta	b4	300
p4	20	Yogyakarta	b5	400

2. Bentuk Normal Kedua (2NF)

Sebuah table relasional berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama.

Tabel II.3 Tabel Bentuk Normal Kedua (2NF)

PEMASOK 2

p#	Status	Kota
p1	20	Yogyakarta
P2	10	Yogyakarta
P3	20	Yogyakarta
P4	30	Yogyakarta
P5	20	Yogyakarta

BARANG

p#	b#	qyt
p1	b1	200
p1	b2	300
p2	b3	400
p2	b4	300
p3	b5	200
p4	b6	100
p4	b7	200

3. Bentuk Normal Ketiga (3NF)

Secara definisi sebuah table berada pada bentuk normal ketiga (3NF) jika table sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama.

Tabel II.4 Tabel Bentuk Normal Ketiga (3NF)

PEMASOK_KOTA

KOTA_STATUS

p#	kota
p1	Yogyakarta
p2	Medan
p3	Medan
p4	Yogyakarta
p5	Bandung

p#	kota
p1	Yogyakarta
p2	Medan
p3	Medan
p4	Yogyakarta
p5	Bandung

4. Bentuk Normal Boyce-Codd (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom lebih dan semua kolom adalah kunci. Bentuk

normal Boyce-Codd (BCNF) adalah versi 3NF yang lebih teliti dan berhubungan dengan table relasional yang mempunyai banyak kunci kandidat, kunci kandidat gabungan, dan kunci kandidat yang saling tumpang tindih. (Jannner Simarmata dan Iman Paryudi, 2006:79-84)

II.8. Kamus Data

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

Tabel II.5 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 di dalam 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 : Jogiyanto; 2005: 730)

Bentuk dari form kamus data dapat dilihat pada sebagai berikut ini.

Barang : KodeBarang + NamaBarang + Satuan + HargaSatuan +

JenisBarang

Supplier : KodeSupplier + NamaSupplier + Alamat + Telepon

Pembelian : NoFaktur + Tanggal + KodeSupplier

Detail Pembelian : NoFaktur + KodeBarang + Jumlah

Admin : UserName + Password + NamaLengkap + StatusAdmin

II.9. Google Maps API

Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer. Google Maps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, Google Maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Kita dapat menambahkan fitur Google Maps dalam web yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan Google Maps API. Google Maps API adalah suatu library yang berbentuk JavaScript.

Cara membuat Google Maps untuk ditampilkan pada suatu web atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta JavaScript, serta koneksi Internet yang sangat stabil. Dengan menggunakan

Google Maps API, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik Google sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia. (myusro.info/2013/02/pengertian-google-maps-api.html).

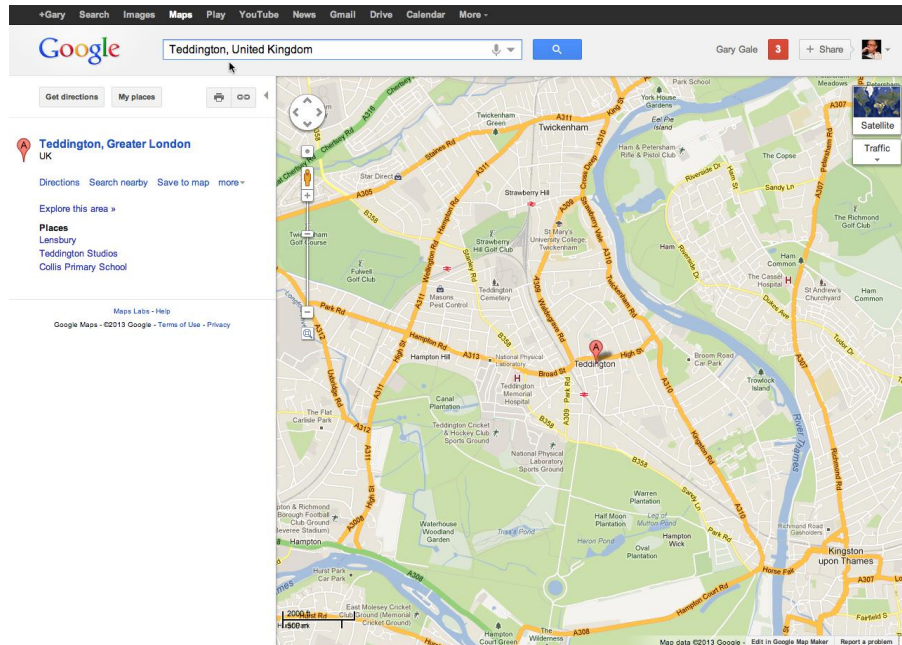
Dalam pembuatan program Google Map API menggunakan urutan sebagai berikut:

1. Memasukkan Maps API JavaScript ke dalam HTML kita.
2. Membuat element div dengan nama map_canvas untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan property-property pada peta.
4. Menuliskan fungsi JavaScript untuk membuat objek peta.
5. Meng-inisiasi peta dalam tag body HTML dengan event onload.

Pada Google Maps API terdapat 4 jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah:

1. *ROADMAP*, ini yang saya pilih, untuk menampilkan peta biasa 2 dimensi
2. *SATELLITE*, untuk menampilkan foto satelit
3. *TERRAIN*, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai

4. *HYBRID*, akan menunjukkan foto satelit yang diatasnya tergambar pula apa yang tampil pada ROADMAP (jalan dan nama kota).



Gambar II.11. Tampilan fitur titik, garis, dan poligon pada Google Maps

(Sumber : Google/picturenews.com)