

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Pengertian dan definisi sistem pada berbagai bidang berbeda-beda, tetapi meskipun istilah sistem yang digunakan bervariasi, semua sistem pada bidang-bidang tersebut mempunyai beberapa persyaratan umum, yaitu sistem harus mempunyai elemen, lingkungan, interaksi antar elemen, interaksi antara elemen dengan lingkungannya, dan yang terpenting adalah sistem harus mempunyai tujuan yang akan dicapai.

Sistem menurut McLeod yang dikutip (2010:34) dalam bukunya yang berjudul "*Management Information System*" adalah sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai tujuan.

Menurut Satzinger, Jackson, dan Burd (2010:6) sistem merupakan sekumpulan komponen yang saling berhubungan dan bekerja bersama untuk mencapai suatu tujuan.

2.2 Informasi

Menurut Gordon B. Davis dalam bukunya Bambang Hartono (2013:15), *information is data has been processed into a form that is meaningful to the recipient and is of real or perceived value in current or prospective decision*. (Informasi adalah data yang telah diolah menjadi suatu bentuk yang berguna bagi penerimanya dan memiliki nilai bagi pengambilan keputusan saat ini atau di masa yang akan datang).

Menurut Henry C. Lucas dalam bukunya Bambang Hartono (2013:15), *information is the interpretation of data to provide meaning by an individual*. (Informasi adalah data yang telah ditafsirkan agar memberikan makna tertentu bagi seseorang)

2.3 Sistem Informasi

Sesungguhnya yang dimaksud sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (*Computer Based Information System* atau CBIS). Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer, walaupun dalam kenyataannya komputer merupakan bagian yang penting. Di buku ini, yang dimaksudkan dengan sistem informasi adalah sistem informasi berbasis komputer. Ada beragam definisi sistem informasi, yaitu :

- a. Alter, sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
- b. Bodnar dan Hopwood, sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
- c. Gelinas, Oram dan Wiggins, sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi keluaran kepada para pemakai.
- d. Hall, Sistem informasi adalah sebuah rangkaian prosedur formal, dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada para pemakai.
- e. Turban, McLean dan Wetherbe, Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi untuk tujuan yang spesifik.
- f. Wilkinson, Sistem informasi adalah kerangka kerja yang mengoordinasikan sumber daya (manusia dan komputer) untuk mengubah masukan (input) menjadi keluaran (informasi) guna mencapai sasaran-sasaran perusahaan.

Berdasarkan berbagai definisi tersebut, dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi) dan dimaksudkan untuk mencapai suatu sasaran atau tujuan (Kadir, 2014).

2.4 Sistem Informasi Geografis

Sistem Informasi Geografis atau Geographic Information System (GIS) adalah sebuah sistem yang didesain untuk menangkap, menyimpan, memanipulasi, menganalisa, mengatur dan menampilkan seluruh jenis data geografis.

Akronim GIS terkadang dipakai sebagai istilah untuk *geographical information science* atau *geospatial information studies* yang merupakan ilmu studi atau pekerjaan yang berhubungan *Geographic Information System*. Dalam artian sederhana sistem informasi geografis dapat kita simpulkan sebagai gabungan kartografi, analisis statistik dan teknologi sistem basis data (*database*).

Pengertian sistem informasi geografis menurut beberapa ahli :

Menurut (Erna Kharistiani, 2013) Pada dasarnya, istilah Sistem Informasi Geografis (SIG) merupakan gabungan dari tiga unsur pokok yaitu sistem, informasi dan geografis. Dengan melihat unsur - unsur tersebut, maka jelas SIG merupakan salah satu sistem informasi yang menekankan pada unsur “informasi geografis”. SIG terdiri dari data spasial dan aspasial.

Menurut (Jurnal Mulawarman, Vol.10 No.1, 2015) SIG adalah kumpulan yang terorganisir dari perangkat keras komputer, perangkat lunak, data geografi dan personal yang dirancang secara efisien untuk memperoleh, menyimpan, mengupdate, memanipulasi, menganalisa dan memanipulasi semua bentuk informasi yang bereferensi geografi.

2.5 Metode Pengembangan Sistem

Menurut Pressman (2015:42), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Nama model ini sebenarnya adalah "*Linear Sequential Model*". Model ini sering disebut juga dengan "*classic life cycle*" atau metode *waterfall*. Model ini termasuk ke dalam model generic pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Berikut adalah tahapan dari metode *waterfall* :

1. Analisa Kebutuhan (*Project Initiation & Requirements Gathering*)
Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.
2. Modeling (*Design System*)
Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.
3. Pembuatan Kode Program (*Construction Code & Test*)
Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

4. Pengujian (*Testing*)

Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

5. Penerapan Program (*Delivery, Support, Feedback*)

Tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.6 UML (*Unified Modeling Language*)

(Adi Nugroho, 2010:6) Bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan yang kompleks sehingga lebih mudah untuk dipelajari dan dipahami.

Tujuan pemodelan dalam kerangka pengembangan sistem adalah sebagai sarana analisis, pemahaman, visualisasi, dan komunikasi antar tim pengembang yang beranggotakan beberapa/banyak anggota.

Beberapa diagram dalam UML yang akan digunakan dalam membantu pengembangan sistem yaitu :

2.6.1 *Use Case Diagram*

Merupakan unit koheren dari fungsionalitas sistem yang tampak dari luar dan diekspresikan sebagai urutan pesan-pesan yang dipertukarkan unit-unit sistem dengan satu atau lebih *actor* yang ada diluar sistem. Kegunaan *use case* sesungguhnya adalah untuk mendefinisikan suatu bagian perilaku sistem yang bersifat koheren tanpa perlu menyiapkan struktur internal sistem yang sedang dikembangkan. Definisi *use case* di dalamnya mencakup semua

prilaku yang ada dalam sistem yang sedang kita kembangkan. Simbol dan keterangan *use case diagram* seperti pada tabel 2.1.

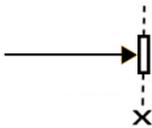
Tabel 2.1 Simbol dan Keterangan *Use Case Diagram*

SIMBOL	KETERANGAN
<p>Aktor</p>  <p>NewClass</p>	<p>Mewakili peran orang, system yang lain atau alat ketika berkomunikasi dengan <i>use case</i>.</p>
<p><i>Use case</i></p>  <p>NewUseCase</p>	<p>Abstraksi dari interaksi antara system dan actor.</p>
<p>Generalisasi</p> 	<p>Menunjukkan spessialisasi actor untuk dapat berpartisipasi dalam <i>use case</i>.</p>
<p><i>Extend</i></p> 	<p>Memspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.</p>
<p><i>Include</i></p> 	<p>Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.</p>

2.6.2 *Sequence Diagram*

Sequence Diagram menekankan pada urutan waktu penerimaan *messege*, kita menjumpai garis hidup objek yaitu garis tegas *vertical* yang mencerminkan ekstensi sebuah objek sepanjang periode waktu. Kebanyakan objek yang hadir pada *interaction diagram* akan eksis sepanjang durasi tertentu dari interaksi, sehingga objek itu diletakkan di atas diagram dengan ‘garis hidup’ digambarkan dari atas hingga ke bagian bawah diagram. Simbol dan keterangan *sqeunce diagram* seperti pada tabel 2.2.

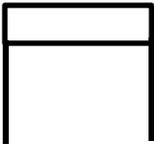
Tabel 2.2 Simbol dan Keterangan *Sequence Diagram*

SIMBOL	KETERANGAN
Objek 	Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan dan ditempatkan di bagian atas diagram.
Garis hidup objek 	Menandakan kehidupan obyek selama urutan dan diakhiri tanda X pada titik dimana kelas tidak lagi berinteraksi.
Waktu aktif 	menandakan ketika suatu objek mengirim atau menerima pesan.
Pesan 	Objek mengirim satu pesan ke objek lainnya.
<i>Create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
<i>Destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah objek yang diakhiri.

2.6.3 Activity Diagram

Activity Diagram atau Diagram Aktivitas menggambarkan alur aktivitas dalam aplikasi, menjelaskan proses masing-masing alur berawal dan proses aplikasi berakhir. Diagram aktivitas juga menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Simbol dan keterangan *activity diagram* seperti pada tabel 2.3.

Tabel 2.3 Simbol dan Keterangan *Activity Diagram*

SIMBOL	KETERANGAN
Status Awal 	Status awal aktivitas system, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan system, aktivitas biasanya diawali dengan kata kerja.
Percabangan 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Status akhir 	Status akhir yang dilakukan system, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Action</i> 	Memperlihatkan masing-masing kelas antar muka saling berinteraksi satu sama lain.
<i>Fork Node</i> 	Satu aliran pada tahap tertentu berubah menjadi beberapa aliran.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.6.4 *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. Atribut merupakan variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa, 2011).

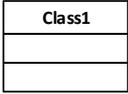
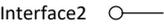
Kelas-kelas yang ada pada struktur sistem, harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

- a. Kelas main
Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
- b. Kelas yang menangani tampilan sistem
Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
- c. Kelas yang diambil dari pendefinisian *use case*
Kelas yang menangani fungsi-fungsi yang baru ada diambil dari pendefinisian *use case*.
- d. Kelas yang diambil dari pendefinisian data
Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas tersebut juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks dan lainnya.

Dalam mengidentifikasi metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain, sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Sebagai aturan secara umum, maka sebuah metode yang dibuat harus memiliki kadar *cohesion* yang kuat dan kadar *coupling* yang lemah. Simbol-simbol yang ada pada diagram kelas adalah seperti pada tabel 2.4.

Tabel 2.4 Simbol dan Keterangan *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem.
<p>Natarmuka/<i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<p>Asosiasi</p> 	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
<p>Kebergantungan</p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
<p>Agregasi</p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

2.7 Bahasa Pemrograman dan Perangkat Lunak Pendukung

Bahasa pemrograman yang digunakan dalam penulisan skripsi ini adalah *PHP* sedangkan perangkat lunak pendukung yang digunakan adalah JQuery mobile, MySQL, Xampp, HTML5, CSS, Google Maps API, dan Sublime Text.

2.7.1 PHP (*Hypertext Preprocessor*)

(Rohi Abdulloh, 2015:3) PHP singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programming*, yaitu Bahasa pemrograman yang diproses di sisi server. Fungsi utama PHP dalam membangun website adalah untuk melakukan pengolahan data pada database. Data website akan dimasukkan ke database, diedit, dihapus, dan ditampilkan pada *website* yang diatur oleh PHP.

Pengembangan demi pengembangan terus berlanjut, ratusan fungsi ditambahkan sebagai fitur dari bahasa PHP, dan di awal tahun 1999, netcraft mencatat, ditemukan 1.000.000 situs di dunia telah menggunakan PHP. Ini membuktikan bahwa PHP merupakan bahasa yang paling populer digunakan oleh dunia web development. Hal ini mengagetkan para developernya termasuk Rasmus sendiri, dan tentunya sangat diluar dugaan sang pembuatnya. Kemudian Zeev Suraski dan Andi Gutsman selaku core developer (programmer inti) mencoba untuk menulis ulang PHP Parser, dan diintegrasikan dengan menggunakan Zend scripting engine, dan mengubah jalan alur operasi PHP. Dan semua fitur baru tersebut di rilis dalam PHP 4.13 Juli 2004, evolusi PHP, PHP telah mengalami banyak sekali perbaikan disegala sisi, dan wajar jika netcraft mengumumkan PHP sebagai bahasa web populer didunia, karena tercatat 19 juta domain telah menggunakan PHP sebagai server side scriptingnya.

2.7.2 jQuery

(Loka Dwiartara) jQuery adalah pustaka kecil dari Javascript bersumber terbuka yang menekankan pada interaksi antara JavaScript dan HTML. Pustaka ini dirilis pada Januari 2006 di BarCamp NYC oleh John Resig dan berlisensi ganda di bawah Lisensi MIT dan GPL. jQuery sangat ringan walaupun hanya sedikit tulisan namun jQuery mampu melakukan hal lebih menakutkan. jQuery mampu membuat tampilan suatu web menjadi lebih menarik. jQuery adalah sebuah

perpustakaan JavaScript berbeban ringan "sedikit menulis, banyak kerja". Perpustakaan jQuery memuat feature-feature berikut:

- Seleksi elemen HTML
- Manipulasi elemen HTML
- Manipulasi CSS
- Fungsi-fungsi event HTML
- Animasi dan JavaScript Effects
- Modifikasi dan HTML DOM traversal

2.7.3 MySQL

(Loka Dwiartara) MySQL bersifat RDBMS (Relational Database Management System) yang memungkinkan seorang admin dapat menyimpan banyak informasi ke table-table, dimana table-table tersebut saling berkaitan satu sama lain. Keuntungan RDBMS sendiri adalah kita dapat memecah database kedalam table-table yang berbeda, setiap table memiliki informasi yang berkaitan dengan table yang lainnya.

MySQL adalah *multiuser database* yang menggunakan bahasa *Structured Query Language (SQL)*. MySQL dalam operasi *client-server* melibatkan *server daemon* MySQL di sisi *server* dan berbagai macam program serta *library* yang berjalan di sisi *client*. MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TeX, mengaku mampu menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data. Informasi selengkapnya dapat dilihat di www.mysql.com

2.7.4 XAMPP

(Andi, 2009:) XAMPP adalah salah satu paket software web server yang terdiri dari Apache, MySQL, PHP dan phpMyAdmin. Mengapa menggunakan XAMPP? Karena XAMPP sangat mudah

penggunaanya, terutama jika Anda seorang pemula. Proses instalasi XAMPP sangat muda, karena tidak perlu melakukan konfigurasi Apache, PHP dan MySQL secara manual, XAMPP melakukan instalasi dan konfigurasi secara otomatis.

2.7.5 HTML (*Hyper Text Markup Language*)

(Rosa A.S, M.Shalahuddin, 2014:2) HTML singkatan dari *Hyper Text Markup Language*, yaitu skrip yang mengatur berupa tag-tag untuk membuat dan mengatur struktur website. Beberapa tugas utama HTML dalam membangun website diantaranya sebagai berikut:

- Menentukan layout website.
- Memformat text dasar seperti pengaturan paragraf, dan format font.
- Membuat list.
- Membuat table.
- Menyisipkan gambar, video, dan audio.
- Membuat link.
- Membuat formulir

2.7.6 *Sublime Text Editor*

Menurut *Wikipedia*, *Sublime text* merupakan sebuah teks editor berbasis *Phyton* yang mempunyai banyak fitur – fitur. *Sublime text* juga mendukung banyak bahasa pemrograman diantaranya *C++*, *C*, *CSS*, *PHP*, *HTML*, *Javascript*, *ASP*, dan masih banyak lagi. *Sublime text* mempunyai beberapa kenggulan – keunggulan yang dapat membantu pengguna dalam membuat sebuah *web development*. Keunggulan-keunggulan yang dimiliki *Sublime Text* :

1. *Multiple Selection*

Multiple selection mempunyai fungsi untuk membuat perubahan pada sebuah kode pada waktu yang sama dan dalam baris yang berbeda. *Multiple selection* ini juga merupakan salah satu fitur

unggulan dari *Sublime Text 3*. Kita dapat meletakkan kursor pada kode yang akan di ubah / edit, lalu tekan *Ctrl + click* atau blok kode yang akan diubah kemudian *Ctrl + D* setelah itu kita dapat merubah kode secara bersamaan.

2. *Drag & Drop*

Dalam teks editor ini pengguna dapat menyeret dan melepas file teks ke dalam editor yang akan membuka *tab* baru secara otomatis.

3. *Multi Platform*

Sublime text juga mempunyai keunggulan dalam bagian platform. *Sublime text* sendiri sudah tersedia dalam berbagai platform sistem operasi, yaitu *Windows*, *Linux*, dan *MacOS*

2.8 *Google Maps*

Menurut *Wikipedia*, *Google Maps* adalah layanan pemetaan *web* yang dikembangkan oleh *Google*. Layanan ini memberikan citra satelit, peta jalan, panorama 360°, kondisi lalu lintas, dan perencanaan rute untuk berpergian dengan berjalan kaki, mobil, sepeda (versi beta), atau angkutan umum. *Google Maps* dimulai sebagai program *desktop* C++, dirancang oleh Lars dan Jens Eilstrup Rasmussen pada *Where 2 Technologies*. Pada Oktober 2004, perusahaan ini diakuisisi oleh *Google*, yang diubah menjadi sebuah aplikasi *web*. Setelah akuisisi tambahan dari perusahaan visualisasi data geospasial dan analisis lalu lintas, *Google Maps* diluncurkan pada Februari 2005. Layanan ini menggunakan *Javascript*, *XML*, dan *AJAX*. *Google Maps* untuk seluler dirilis pada bulan September 2008. Pada Agustus 2013, *Google Maps* bertekad untuk menjadi aplikasi yang paling populer di dunia untuk ponsel cerdas, dengan lebih dari 54% dari pemilik ponsel cerdas di seluruh dunia menggunakannya setidaknya sekali.

2.8.1 *Google Maps API*

Seperti yang tercatat oleh Svennerberg, *Google Maps API* adalah API yang paling populer di internet. Pencatatan yang dilakukan pada bulan Mei tahun 2010 menyatakan bahwa 43% *mash up* (aplikasi dan situs web yang

menggabungkan dua atau lebih sumber data) menggunakan Google Maps API. Beberapa tujuan dari penggunaan *Google Maps* API adalah untuk melihat lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain sebagainya.

2.9 Basis Data

Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data di maksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas (Kadir,2014). 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.

Umumnya DBMS menyediakan fitur-fitur sebagai berikut :

a. Independensi data program

Karena basis data ditangani oleh DBMS, program dapat ditulis sehingga tidak tergantung pada struktur data dalam basis data. Dengan perkataan lain, program tidak akan terpengaruh 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.

Komponen-komponen yang menyusun lingkungan DBMS terdiri atas:

a. Perangkat keras

Perangkat keras digunakan untuk menjalankan DBMS beserta aplikasi-aplikasinya. Perangkat keras berupa komputer dan periferal pendukungnya. Komputer dapat berupa PC, minikomputer, mainframe, dan lain-lain.

b. Perangkat lunak

Komponen perangkat lunak mencakup DBMS itu sendiri, program aplikasi, serta perangkat lunak pendukung untuk komputer dan jaringan. Program aplikasi dapat dibangun dengan menggunakan bahasa pemrograman seperti *C++*, *Pascal*, *Delphi*, atau *Visual BASIC*.

c. Data

Bagi sisi pemakai, komponen terpenting dalam DBMS adalah data karena dari data inilah pemakai dapat memperoleh informasi yang sesuai dengan kebutuhan masing-masing.

d. Prosedur

Prosedur adalah petunjuk tertulis yang berisi cara merancang hingga menggunakan basis data. Beberapa hal yang dimasukkan dalam prosedur:

1. Cara masuk ke DBMS (*login*).

2. Cara memakai fasilitas-fasilitas tertentu dalam DBMS maupun cara menggunakan aplikasi.
 3. Cara mengaktifkan dan menghentikan DBMS.
 4. Cara membuat cadangan basis data dan cara mengembalikan cadangan ke DBMS.
- e. Orang
- Komponen orang dapat dibagi menjadi tiga kelompok, yaitu :
1. Pemakai akhir (*end-user*).
 2. Pemogram aplikasi.
 3. Administrator basis data.

Terdapat beberapa elemen basis data, yaitu :

a. *Database*

Database atau basis data adalah kumpulan tabel yang mempunyai kaitan antara suatu tabel dengan tabel lainnya sehingga membentuk suatu bangunan data.

b. Tabel

Tabel adalah kumpulan *record-record* yang mempunyai panjang elemen yang sama dan atribut yang sama namun berbeda data *valuenya*.

c. Entitas

Entitas adalah sekumpulan objek yang terdefiniskan yang mempunyai karakteristik sama dan bisa dibedakan satu dengan lainnya. Objek dapat berupa barang, orang, tempat atau suatu kejadian.

d. Atribut

Atribut adalah deskripsi data yang bisa mengidentifikasi entitas yang membedakan entitas tersebut dengan entitas yang lain. Seluruh atribut harus cukup untuk menyatakan identitas objek atau dengan kata lain, kumpulan atribut dari setiap entitas dapat mengidentifikasi keunikan suatu individu.

e. *Data Value* (Nilai Data)

Data value adalah data aktual atau informasi yang disimpan pada tiap data, elemen atau atribut. Atribut nama pegawai menunjukkan tempat dimana

informasi nama karyawan disimpan, nilai datanya misalnya adalah Anjang, Arif, Suryo dan lain-lain yang merupakan isi data nama pegawai tersebut.

f. File

File adalah kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda nilai datanya.

g. Record/Tuple

Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu *record* mewakili satu data atau informasi.

2.10 Mobile Application

Menurut Wikipedia, pengertian aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon mobile berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi mobile merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA.

2.11 Pengertian Kuliner

Menelusuri kuliner sebagai sebuah tradisi dan warisan yang orisinal sukar dilakukan karena pengaruh dan sentuh budaya dari luar. Mengenai cara atau teknik pengolahan makanan, setiap daerah memiliki kekhasan yang pada akhirnya memunculkan “cita rasa”.

Menurut Soekresno (2012, p16) kuliner tersebut perlu dikelola secara profesional dengan adanya usaha restoran/rumah makan. adapun jenis-jenis restoran yaitu

1. Restoran Formal adalah industri jasa pelayanan makanan dan minuman yang dikelola secara komersial dan profesional dengan pelayanan eksklusif. Contoh : *Main Dining Room*.
2. Restoran Informal adalah industri jasa pelayanan makanan dan minuman yang dikelola secara komersial dan profesional dengan lebih mengutamakan kecepatan pelayanan, kepraktisan dan percepatan frekuensi yang silih berganti. Contoh : *cafe, coffee shop, canteen*.
3. Spesial Restoran adalah industri jasa pelayanan makanan dan minuman yang dikelola secara komersial dan profesional dengan menyediakan makanan khas dan diikuti dengan penyajian yang khas dari suatu negara tertentu. Contoh : *japanese restaurant, chinese restaurant*.

2.12 Penelitian Sebelumnya

Penelitian ini mengacu pada penelitian terdahulu oleh Dewanto R.A tahun 2006 judul “Rancang bangun informasi geografis wisata kuliner berbasis web dengan Google API”. Penelitian ini membahas tentang sebuah aplikasi web yang menampung dan mempublikasikan informasi pusat - pusat kuliner suatu daerah. Informasi yang ditampilkan dalam bentuk peta yang memudahkan pengguna dalam mencari informasi.

Penelitian ini juga mengacu berdasarkan penelitian terdahulu oleh Dimas Maulana tahun 2010 judul “Membangun sistem informasi kuliner Bandung berbasis Web”. Dalam penelitian ini user yang berkunjung ke dalam sistem informasi ini bisa melihat informasi objek wisata kuliner dimulai dari lokasi, penjelasan kuliner dan menu yang ditawarkan dan terakhir juga tersedia informasi set layout dari tempat jualan objek kuliner yang bersangkutan. Selain itu informasinya dikelompokkan berdasarkan objek kulinernya yang bersangkutan.

