

Bab 2

Tinjauan pustaka

2.1 Pengertian Sistem Informasi

Sistem pada dasarnya adalah pengelolaan data untuk menghasilkan informasi yang berguna dan mendukung dalam suatu organisasi untuk memberikan kemudahan terhadap proses-proses yang ada. Secara sederhana sistem adalah suatu kumpulan dari unsur, komponen, atau variabel-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu [2].

Manfaat sistem adalah untuk mengintegrasikan semua unsur yang ada dalam suatu ruang lingkup, dimana komponen-komponen yang ada tidak dapat berdiri sendiri. Komponen atau sub sistem harus saling berintegrasi untuk membentuk suatu kesatuan sehingga tujuan dari sebuah sistem dapat tercapai.

Informasi adalah hasil pengolahan data yang diperoleh dari setiap elemen sistem yang ada dan dibuat dalam bentuk yang mudah dipahami oleh penerimanya. Informasi menggambarkan kejadian-kejadian nyata untuk menambah pemahaman terhadap fakta-fakta yang ada, sehingga dapat digunakan untuk pengambilan suatu keputusan yang bersumber dari data. Kualitas dari suatu informasi tergantung pada tiga hal, yaitu: [2].

a) Akurat (*accurate*)

Informasi harus bebas dari kesalahan dan tidak boleh menyesatkan. Akurat juga berarti bahwa informasi harus jelas mencerminkan maksud pembuatannya.

b) Tepat waktu (*timelines*)

Informasi yang sampai pada penerima tidak boleh tertunda. Informasi yang sudah usang nilainya akan berkurang. Karena informasi merupakan landasan didalam pengambilan suatu keputusan.

c) Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk penggunanya. Relevansi informasi untuk setiap orang, antara satu dan lainnya pasti memiliki perbedaan.

Manfaat adanya sistem informasi yaitu untuk menyajikan informasi yang tepat dan dapat mendukung pengambilan sebuah keputusan, menyajikan informasi untuk mendukung operasi atau aktivitas harian dan menyajikan informasi yang sesuai dengan kebutuhan. Beberapa komponen sistem informasi dapat diklasifikasikan sebagai berikut:

- a) Perangkat keras (*hardware*) dan perangkat lunak (*software*) yang berfungsi sebagai mesin.
- b) Manusia (*people*) dan prosedur yang merupakan sumber daya manusia dan tata cara penggunaan sebuah mesin.
- c) Data yang merupakan jembatan penghubung antara manusia dan mesin agar dapat terjadi proses pengolahan data.

2.2 Customer relationship management (CRM)

CRM adalah strategi utama dalam bisnis yang mengintegrasikan proses-proses dan fungsi-fungsi internal dengan jaringan eksternal untuk menciptakan dan mewujudkan nilai bagi para konsumen sasaran secara profitable [3]. CRM didukung oleh data konsumen yang berkualitas dan teknologi informasi.

2.2.1 Klasifikasi CRM

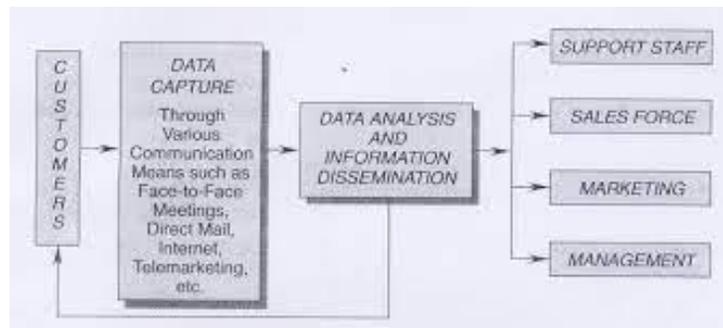
CRM dapat dikajikan dalam 3 tataran, yakni strategis, operasional, dan analitik [3]:

- a) CRM strategis berfokus pada upaya untuk mengembangkan kultur usaha yang berorientasi pada pelanggan atau customer-centric. Ini ditujukan untuk merebut hati konsumen dan menjaga loyalitas mereka dengan menciptakan serta memberikan nilai bagi pelanggan yang unggul para pesaing [3].

- b) CRM Operasional lebih berfokus pada otomatisasi cara-cara perusahaan dalam berhubungan dengan pelanggan. Berbagai aplikasi perangkat lunak CRM memungkinkan fungsi-fungsi pemasaran, penjualan, dan pelayanan dapat berjalan secara otomatis [3].
- c) Beberapa contoh pelayanan yang diberikan melalui CRM operasional, diantaranya
 - 1) Menyediakan pencarian produk. Pelanggan sering kali mengalami kesulitan dalam mencari produk yang mereka inginkan, karena itu diperlukan fasilitas search
 - 2) Menyediakan produk atau pelayanan gratis, sesuatu yang dapat menarik pelanggan untuk mengunjungi web adalah tersedianya produk atau pelayanan gratis
 - 3) Menyediakan pelayanan atau informasi tentang penggunaan produk
 - 4) Menyediakan pemesanan online
 - 5) Menyediakan fasilitas informasi status pemesanan
- d) CRM Analitis digunakan untuk mengeksploitasi data konsumen demi meningkatkan nilai mereka (nilai perusahaan). Sistem ini dikembangkan berdasarkan informasi mengenai konsumen. Data pelanggan dapat diperoleh dari pusat-pusat informasi atau bank data yang dimiliki oleh setiap perusahaan yang relevan, yakni data penjualan (riwayat pembelian barang atau jasa), data financial (riwayat pembayaran atau skor kredit), dan data lainnya [3].

2.2.2 Proses CRM

Sistem CRM yang diintegrasikan secara penuh (gambar 2.1) mengumpulkan data tentang pelanggan dari berbagai jenis data penting pelanggan (contoh: telepon, e-mail, fax, dll) dan menyimpan informasi di database yang terpusat. Setelah dikumpulkan, disimpan, dan dikelola, informasi ini dapat dianalisa, dan diakses oleh sejumlah user yang berbeda dalam perusahaan. Sistem CRM yang lengkap mempunyai kemampuan untuk mengintegrasikan informasi pelanggan dari seluruh perusahaan kedalam format yang user friendly, real time, dan mentransformasikan seluruh perusahaan kedalam sebuah jaringan sales [4].



Gambar 2.1 Proses CRM (Khalid, 2002)

2.2.3 Manfaat Penggunaan Sistem CRM

Manfaat Penggunaan sistem CRM membantu perusahaan untuk mengembangkan produk baru berdasarkan pengetahuan yang lengkap tentang keinginan pelanggan, dinamika pasar dan pesaing dengan cara (feby.blog.perbanas.ac.id):

- a) Menjaga pelanggan yang sudah ada
- b) Menarik pelanggan baru
- c) Cross Selling: menjual produk lain yang mungkin dibutuhkan pelanggan berdasarkan pembeliannya
- d) Upgrading: menawarkan status pelanggan yang lebih tinggi (gold card vs. silver card)
- e) Identifikasi kebiasaan pelanggan untuk menghindari penipuan
- f) Mengurangi resiko operasional karena data pelanggan tersimpan dalam satu system
- g) Respon yang lebih cepat ke pelanggan
- h) Meningkatkan efisiensi karena otomasi proses
- i) Meningkatkan kemampuan melihat dan mendapatkan peluang, layanan dan lain sebagainya

2.4 Analisis sistem dan desain

Analisis sistem (*system analyst*) adalah sebuah teknik pemecahan masalah yang menguraikan sebuah sistem menjadi bagian-bagian komponen dengan tujuan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk meraih tujuan mereka [2].

Pada analisis sistem, dapat diketahui solusi seperti apa yang dibutuhkan dan mendukung dalam suatu pekerjaan. Analisis sistem juga berisi tentang Langkah-langkah pemecahan masalah yang dapat menghasilkan suatu informasi.

2.4.1 Model Driven

Analisis terstruktur, teknik informasi dan analisis berorientasi objek adalah contoh-contoh dari analisis model-driven. Analisis model-driven adalah analisis yang menggunakan gambar untuk mengkomunikasikan masalah-masalah, persyaratan dan solusi-solusi bisnis [2].

Analisis model-driven ini biasanya melakukan analisis sistem terstruktur dalam bentuk *flowchart* yang dibuat dengan menggunakan program aplikasi *grafis generatepurpose*, seperti *Microsoft Visio*. Hal ini dapat memudahkan *user* dalam mengetahui dan menganalisis komponen pembangun sistem.

2.4.2 Pengembangan Metodologi SDLC

Pengembangan dibutuhkan untuk merancang dan membangun suatu sistem informasi agar menjadi lebih baik dari sebelumnya. Pengembangan sistem informasi dibutuhkan metodologi untuk pengembangannya yaitu menggunakan SDLC (*system development life cycle*). Menurut Sukamto dan Shalahuddin mendefinisikan SDLC sebagai proses mengembangkan atau mengubah suatu sistem dengan menggunakan model dan metodologi yang berdasarkan cara yang teruji baik [5]. Menurut Nugroho SDLC memiliki kelebihan dan kekurangan sebagai berikut:

a) Kelebihan SDLC

Pengembang berfokus terhadap suatu tahap terlebih dahulu untuk pindah ke tahap selanjutnya, dikarenakan tidak bisa dilakukan perpindahan tahap ketika satu tahapan belum terselesaikan [5].

b) Kekurangan SDLC

SDLC memakan banyak waktu dan biaya saat terjadi pembaruan maka saat melakukan pengembangan dilakukan dengan asumsi setiap tahap tidak boleh salah [5].

2.4.3 Metode Waterfall

Metode penelitian yang diterapkan pada penelitian ini adalah dengan pengembangan metode waterfall. Metode waterfall merupakan model pengembangan sistem informasi yang sistematis dan sekuensial [6]. Metode Waterfall memiliki tahapan-tahapan sebagai berikut [7]:

a) Analisis definisi dan kebutuhan (*Requirements analysis and definition*)

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

b) Perencanaan desain sistem dan perangkat lunak (*System and software design*)

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

c) Implementasi (*Implementation and unit testing*)

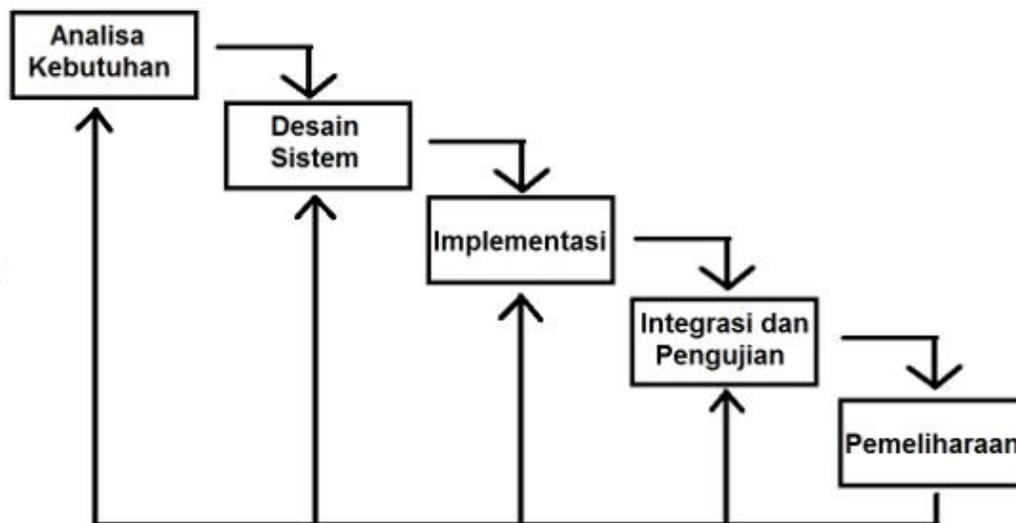
Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

d) Integrasi dan pengujian (*Integration and system testing*)

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke customer

e) Pemeliharaan (*Operation and maintenance*)

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. Maintenance melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.



Gambar 2.2 – Metode WaterFall

Gambar 2.2 adalah bagan metode waterfall yang merupakan metode pengembangan sistem yang digunakan pada penelitian ini.

2.5. UML (*Unified Modelling Language*)

Notasi UML dibuat sebagai kolaborasi dari Grady Booch, DR. James Rumbaugh, Ivar Jacobson, Rebecca Wirfs-Brock, Peter Yourdon, dan lainnya. Semuanya bekerja di *Rational Software Corporation* dan berfokus pada standarisasi dan perbaikan ulang UML. Simbol UML mirip dengan Boch, notasi OMT dan juga ada kemiripan dengan

notasi lainnya [8]. UML muncul karena adanya kebutuhan pemodelan visual untuk memberi spesifikasi, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Ada beberapa diagram yang disediakan dalam UML antara lain [8].

- 1) *Diagram use case (use case diagram).*
- 2) *Diagram aktivitas (activity diagram).*
- 3) *Diagram sekuensial (sequence diagram).*
- 4) *Diagram kelas (class diagram).*
- 5) *Diagram kolaborasi (collaboration diagram).*
- 6) *Diagram statechart (statechart diagram).*
- 7) *Diagram komponen (component diagram).*
- 8) *Diagram deployment (deployment diagram).*

2.5.1 Use Case Diagram

Diagram *use case* menyajikan interaksi antara *use case* dan aktor. Dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai atau pengguna. Pelajari setiap penjelasan simbol *use case diagram* yang dijelaskan dalam Tabel 2.1.

Tabel 2.1. Use case diagram

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

Tabel 2.1. *Use Case Diagram (Lanjutan)*

No	Simbol	Nama	Keterangan
2		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
3		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .

Sumber: Sholiq (2006)

2.5.2 Activity Diagram

Diagram aktivitas menggambarkan aliran fungsionalitas sistem. Dapat juga digunakan untuk menggambarkan aliran kejadian (*flow of events*) dalam *use case*. aktivitas dalam *diagram* dipresentasikan dengan bentuk bujur sangkar bersudut tidak lancip, yang di dalamnya berisi langkah-langkah apa saja yang terjadi dalam aliran kerja. Ada sebuah keadaan mulai (*start state*) yang menunjukkan dimulainya aliran kerja dan sebuah keadaan selesai (*end state*) yang menunjukkan akhir *diagram*, titik keputusan dipresentasikan dengan *diamond*. *Diagram* aktivitas tidak perlu dibuat untuk setiap aliran kerja, tetapi *diagram* ini akan sangat berguna untuk aliran kerja yang kompleks dan melebar. Penjelasan simbol-simbol *activity diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2. *Activity Diagram*

No	Simbol	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Decision</i>	<i>State</i> dari sistem yang mencerminkan pengambilan keputusan.

Tabel 2.2. *Activity Diagram (Lanjutan)*

No	Simbol	Nama	Keterangan
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
6		<i>Control Flow</i>	Arus aktivitas.
7		<i>Receive</i>	Tanda Penerimaan.

Sumber: Sholiq (2006)

2.5.3 Sequence Diagram

Diagram sekuensial digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Penjelasan simbol-simbol yang digunakan pada *sequence diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3. *Sequence Diagram*

No	Simbol	Nama	Keterangan
1		<i>Lifeline</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

Tabel 2.3. Sequence Diagram (lanjutan)

No	Simbol	Nama	Keterangan
3		<i>Self-message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
4		<i>Activation</i>	Indikasi dari sebuah objek yang melakukan suatu aksi.

Sumber: Sholiq (2006)

2.5.4 Diagram Komponen

Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.

2.5.5 Diagram Deployment

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (run-time). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram deployment berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (distributed computing).

2.6 Konteks Diagram

Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan antara entity luar, masukan dan keluaran dari sistem. Diagram konteks direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem[9].

2.7 Diagram Dekomposisi

Diagram dekomposisi adalah diagram yang digunakan untuk menggambarkan dekomposisi sebuah sistem disebut juga bagan hierarki menunjukkan dekomposisi fungsional top-down dan struktur sistem. Diagram dekomposisi merupakan alat perencanaan untuk model proses yang lebih detail, yaitu diagram aliran data.

2.8 DFD (*Data flow diagram*)

Data Flow Diagram (DFD) merupakan suatu diagram yang menggambarkan alir data dalam suatu entitas ke sistem atau sistem ke entitas. DFD juga dapat diartikan sebagai teknik grafis yang menggambarkan alir data dari input atau masukan menuju atau output[10].

Tabel 2.4 Simbol-simbol Data Flow Diagram

Nama Simbol	<i>DeMarco and Yourdan Symbols</i>
Entitas eksternal	
Proses	
Aliran Data	
<i>Data Store</i>	

Keterangan simbol yang digunakan dalam DFD:

- a) Entitas eksternal dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi di luar sistem.
- b) Proses adalah orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.

- c) Aliran data dengan arah khusus dari sumber ke tujuan.
- d) Data Store Penyimpanan data atas tempat data di refer oleh proses.

2.9 ERD (*Entity relationship diagram*)

Entity Relationship Diagram (ERD) menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antardata. Pada dasarnya ada 3 macam simbol yang digunakan, yaitu [11]:

a) *Entity*

Entity adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Entitas digambarkan dalam bentuk persegi empat.

b) *Atribut*

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

c) *Hubungan/Relationship*

Sebagaimana halnya entitas, hubungan pun harus dibedakan antara hubungan atau bentuk hubungan antarentitas dengan isi dari hubungan itu sendiri. *Relationship* digambarkan dalam bentuk intan (*diamonds*).

2.10 Website

Website merupakan kumpulan halaman web yang saling terhubung dan berkasnya saling berkaitan. Web terdiri dari halaman, dan kumpulan halaman yang dinamakan beranda. beranda merupakan posisi teratas dengan halaman terkait berada di bawahnya [5].

Jenis-jenis website ada 2 (dua) dibawah ini sebagai berikut:

a) Website Statis

website yang mempunyai halaman yang tidak berubah. Ketika ingin merubah suatu halaman hanya bisa dilakukan secara manual dengan cara merubah kode dalam struktur web keseluruhan [5].

b) Website Dinamis

Website yang dapat dilakukan *update* sesering mungkin. Biasanya selain dimana utamanya yang bisa diakses oleh para pengguna pada umumnya [5].

2.10.1 Bahasa pemrograman website

Bahasa pemrograman website merupakan bahasa komputer yang berisikan suatu proses memberikan intruksi sebuah perintah dalam membuat web sehingga dapat digunakan dan ditampilkan dengan bantuan browser. Jenis Bahasa pemrograman dalam pembuatan web, sebagai berikut [12]

a) *Cascading style sheet (CSS)*

Bahasa pemrograman dalam web yang berarti halaman penataan menurun yang menjelaskan tampilan dari sebuah halaman web dalam *mark-up language*

b) *Javascript*

Bahasa pemrograman yang sangat mudah dalam membuat web karena bersifat *clientside programming*. Tipe bahasa yang dilakukan oleh client dalam pembuatannya seperti google chrome dan mozilla firefox.

c) *Hyper Text markup language (HTML)*

Bahasa pemrograman struktur standar dalam yang mendeskripsikan sebuah halaman dan berfungsi sebagai publikasi dokumen online kemudian dapat diakses oleh web.

d) *HyperText Preprocessor (PHP)*

Bahasa pemrograman yang dijalankan melalui laman web yang umumnya digunakan mengolah informasi di internet. PHP bersifat gratis dan menyatu dengan HTML tetapi berbeda server dalam script nya.

e) Bootstrap

Dalam perancangan sistem atau aplikasi web yang responsive cepat, gratis dan mudah maka digunakannya tools atau framework bootstrap, bootstrap sendiri terdiri dari HTML maupun CSS didalamnya dan dapat dengan mudah menghasilkan atau menampilkan layout, grid, table, form, navigation dan lain-lain. Didalam bootstrap juga terdapat jquery plugins yang dapat menghasilkan UI komponen seperti Tab, Dropdown, Tooltip, Button dan lain-lain. Bootstrap dapat membantu dalam membuat responsive website dengan cepat dan mudah dan dapat berjalan sempurna pada browser yang tersedia saat ini diantaranya internet explorer, opera firefox, dan google chrome.

2.11. Basis Data

Sistem basis data merupakan sistem yang terintegrasi bertujuan untuk memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Dapat diartikan, basis data merupakan data yang aman penyimpanannya yang memuat informasi dengan akses yang cepat disaat dibutuhkan. Kebutuhan basis data dalam sistem informasi meliputi [13].

- a) Kebutuhan memasukan, mengambil dan simpan data.
- b) Kebutuhan membuat berdasarkan data yang telah disimpan.

2.11.1 Database Management System (DBMS)

DBMS merupakan sebuah perangkat lunak yang berfungsi untuk mengelola *database* secara efektif dan efisien. menurut Connolly dan Begg, Sistem manajemen basis data adalah sistem perangkat lunak yang memungkinkan pengguna untuk menentukan, membuat, memelihara, dan mengelola akses ke basis data. Tujuannya adalah mengelola kumpulan data sehingga lebih aman dan akses yang cepat untuk digunakan secara bersamaan untuk menangani data yang beskala banyak [7]

2.11.2 Keuntungan DBMS

Keuntungan DBMS yaitu kontrol terhadap data didalam suatu wadah penyimpanan sehingga memudahkan, data yang didalam penyimpanan konsisten tidak berubah, pembagian data tersebut dimiliki satu orang kemudian bisa dibagikan dengan perizinan oleh satu pihak, integritas data tersebut baik dikarenakan melalui validasi, dan meningkatkan produktifitas kinerja karyawan dalam penggunaannya karna mudah.

2.11.3 Kerugian DBMS

Kerugian dari DBMS yaitu kompleksitas data, ukuran penyimpanan yang sangat besar, biaya dari DBMS yang bervariasi, lalu ada biaya tambahan dari perangkat keras sehingga perlu biaya tambahan dan performa dari aplikasi tersebut bisa dibilang tidak selalu cepat.

2.12 Implementasi sistem informasi

Pada tahap implementasi sistem, adalah tahap yang menentukan pada perancangan sistem secara keseluruhan, karena apabila desain sistem yang sudah kita buat secara detail yang meliputi desain sistem, desain *input* dan desain *output* yang kita tidak terapkan serasa percuma. Berikut merupakan hal-hal yang berkaitan dengan implementasi sistem[14].

a. Pengkodean

Pengkodean merupakan suatu tahapan dari analisis kebutuhan sistem dan desain sistem yang dituliskan dalam suatu bahasa pemrograman komputer tertentu yang biasanya oleh pabrik komputer yang telah ditentukan spesifikasinya.

b. Testing

Testing merupakan suatu proses dari eksekusi program dengan secara intensif untuk menemukan suatu kesalahan-kesalahan atau *error* pada sistem.

c. Dokumentasi

Dokumentasi adalah suatu kegiatan pencatatan terhadap setiap langkah-langkah pekerjaan mulai dari awal sampai akhir.