

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

Penelitian serupa pernah dilakukan oleh Rhyca Putri Ardy, dkk. Mereka pernah melakukan riset tentang memanfaatkan *SMS Gateway* sebagai layanan informasi siswa pada TK Xaverius 5 Palembang [2]. *Framework* yang mereka gunakan sama-sama memakai *gammu*, namun untuk *platform* aplikasi yang mereka kembangkan adalah aplikasi *desktop* dengan menggunakan *Microsoft Visual Studio 2008*, kelemahan dari aplikasi *desktop* adalah tidak dapat digunakan di komputer lain, sebelum memindahkan program tersebut ke komputer baru.

Penelitian lainnya juga dilakukan oleh Dennis Eko Harbiyanto, yang membahas mengenai bagaimana meningkatkan pembayaran sekolah dengan memanfaatkan sistem informasi berbasis *Gammu SMS Gateway* dengan *platform* aplikasi web pada SMK Bhinneka Karya 1 Boyolali [3].

**Tabel 2.1 Penelitian Terkait**

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
1.	R. Putri, R. Atista, dan A. Wahyu, 2013	Pemanfaatan <i>SMS Gateway</i> dalam pelayanan informasi aktifitas siswa	Metodologi RUP ( <i>Rational Unified Process</i> ) dengan menggunakan UML 2 dimensi.	Mebutuhkan beberapa fasilitas perangkat lagi untuk meningkatkan efisiensi kinerja guru.

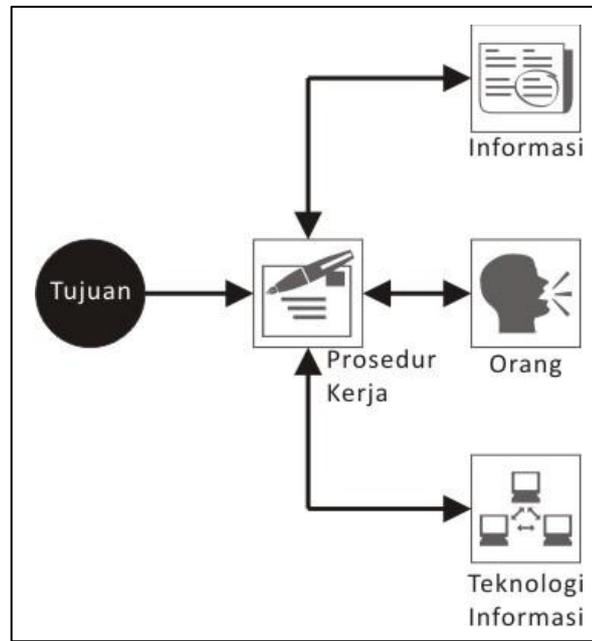
No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
2.	D.E. Harbiyanto, 2015	Sistem Informasi Pembayaran Sekolah Berbasis SMS Gateway	Metode R&D ( <i>Research &amp; Development</i> ) dengan pengujian model ISO 9126.	Hasil pengujian dapat memenuhi berbagai aspek sehingga layak digunakan.

## 2.2 Sistem Informasi

Awalnya, yang dimaksud sistem informasi sebenarnya tidak selalu menggunakan komputer. Sistem informasi yang dalam pengoperasiannya menggunakan komputer biasanya disebut dengan sistem informasi berbasis komputer (*Computer-Based Information Systems* atau CBIS); walaupun dalam kehidupan sehari-hari, kini komputer merupakan sesuatu yang penting dalam menjalankan berbagai kegiatan maupun bisnis. Terdapat banyak ragam definisi sistem informasi menurut ahli. Gelinas, Oram, dan Wiggins mengatakan, sistem informasi merupakan suatu sistem buatan manusia yang secara umum terdiri dari komponen-komponen manual manual dan otomatis (terkomputerisasi) dibuat dengan tujuan untuk menyimpan, mengelompokkan, dan mengelola data, serta menyediakan keluaran berupa informasi untuk para pengguna [4].

Menurut Wilkinson, sistem informasi yaitu skema kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mentransformasikan masukan (*input / data*) menjadi sebuah keluaran (*output / informasi*), dengan tujuan mencapai sasaran-sasaran perusahaan [5]. Sedangkan menurut definisi Hall, sistem informasi merupakan sebuah rangkaian urutan yang formal di mana data diklasifikasikan, diolah menjadi suatu hasil berupa informasi, dan diteruskan kepada pengguna [6].

Sedangkan definisi dari Alter, sistem informasi yaitu campuran antar prosedur kerja, orang, informasi, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi [7].



**Gambar 2.1 Definisi Sistem Informasi [7]**

Tujuan dapat didefinisikan sebagai apa yang ditargetkan dan akan dicoba dilakukan untuk sistem, didukung dengan prosedur kerja atau cara kerja yang dilakukan oleh orang yang memasukkan, mengolah, dan memakai data, dengan bantuan teknologi informasi berupa perangkat keras dan lunak yang digunakan untuk memproses data yang terformat seperti teks, gambar, suara, dan video.

Berdasarkan berbagai penjelasan tersebut, ditarik kesimpulan bahwa sistem informasi mencakup beberapa komponen (komputer, manusia, prosedur kerja, dan teknologi informasi), mengolah sesuatu (data menjadi informasi), dan ditujukan untuk mencapai suatu tujuan atau sasaran [8].

### 2.3 Teknologi Informasi

IT atau *Information Technology*, dulunya lebih dikenal dengan istilah teknologi komputer atau EDP (*Electronic Data Processing* atau Pengolah Data Elektronik). Istilah teknologi seringkali terdengar rancu dengan istilah sistem informasi itu

sendiri. Turban, McLean, dan Wetherbe menjelaskan bahwa teknologi informasi adalah sekumpulan sistem informasi, pemakai, dan manajemen. Akan tetapi berdasarkan pada pernyataan Alter mengenai sistem informasi, teknologi informasi merupakan sebuah bagian dari sistem informasi.

Secara umum, teknologi informasi dibedakan menjadi dua kelompok, yaitu *software* (perangkat lunak) yang terdiri berupa perintah-perintah untuk dijalankan oleh perangkat keras agar berjalan sesuai dengan apa yang ditugaskan dari instruksi tersebut; dan *hardware* (perangkat keras) yang merupakan *tools* / peralatan yang mempunyai wujud fisik, dan berfungsi untuk menjalankan instruksi yang diperintah oleh perangkat lunak.

Namun menurut Haag, teknologi informasi dibagi menjadi enam kategori, yang terdiri dari:

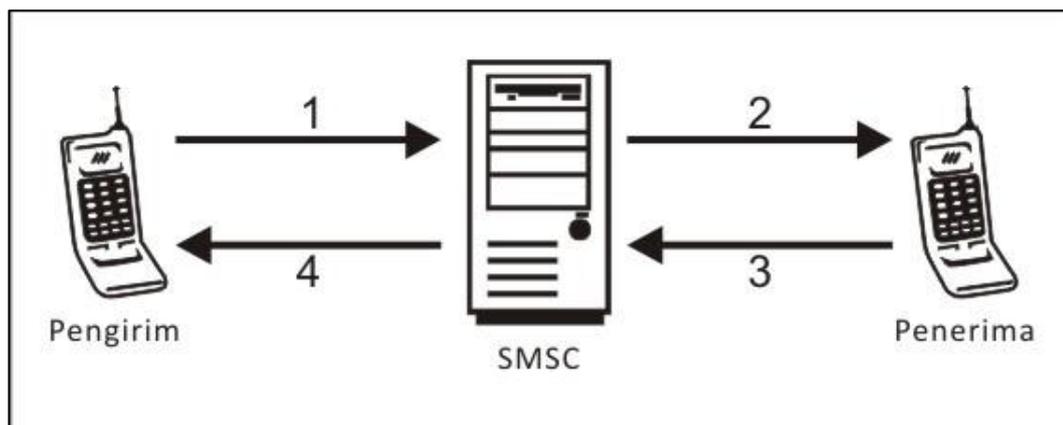
- *Input technology* (teknologi masukan) perangkat berbentuk fisik yang berfungsi untuk menangkap dan merekam data informasi yang dimasukan oleh pengguna;
- *Output technology* (teknologi keluaran) perangkat berbentuk fisik yang berfungsi untuk menampilkan dan menyajikan data informasi dari apa yang diinstruksikan teknologi masukan;
- *Software technology* (teknologi perangkat lunak) sekumpulan instruksi yang dijalankan oleh perangkat keras sesuai dengan yang ditugaskan oleh perintah tersebut;
- *Storage technology* (teknologi penyimpanan) menyangkut berbagai media perangkat yang berfungsi untuk menyimpan data. Bisa berupa fisik seperti *hard disk drive*, maupun non-fisik seperti *database*;
- *Telecommunication technology* (teknologi telekomunikasi) adalah teknologi yang memiliki tujuan untuk menghubungkan dua atau lebih pengguna secara jarak jauh, contohnya seperti internet atau telepon;
- *Processing machine* (mesin pemroses) yang merupakan bagian paling berkontribusi dalam teknologi informasi, karena memiliki fungsi untuk mengingat data hingga program, dan menjalankannya [9].

## 2.4 SMS Gateway

SMS atau *Short Message Service* (Layanan Pesan Singkat) adalah sebuah layanan komunikasi pengiriman suatu pesan maupun penyebaran informasi melalui media teks. SMS juga dapat mengirim gambar, suara, dan film, fitur ini disebut juga sebagai *Multimedia Messaging Service*.

Panjang maksimal pesan per satuan SMS hanya 160 karakter alfanumerik termasuk spasi. Pembatasan ini dipengaruhi oleh pemanfaatan sisa jalur sepanjang 140 oktet atau 1120 bits. Namun tidak menutup kemungkinan pengiriman sebuah pesan melebihi 160 karakter, akan tetapi pengguna tersebut diharuskan membayar lebih dari satu kali tarif SMS yang diterapkan oleh tiap operator penyedia jasa layanan.

Sistem mekanisme cara kerja SMS ini adalah dengan mengirim pesan dari satu terminal pengguna ke terminal pengguna yang lain, dengan bantuan sebuah entitas yang biasa disebut dengan SMSC atau *Short Message Service Center*. Pada saat suatu pesan SMS terkirim dari satu terminal, pesan tersebut tidak bisa langsung sampai menuju terminal tujuan. Sebelumnya pesan tersebut dikirim dahulu menuju SMSC, dan kemudian baru pesan tersebut dikirim menuju terminal tujuan akhir.



Gambar 2.2 Cara Kerja SMS

1. Pengirim mengirimkan pesan ke penerima, melalui SMSC sebagai perantara.
2. SMSC akan meneruskan pesan menuju ke penerima.
3. Setelah pesan masuk, secara otomatis penerima mengirimkan pemberitahuan bahwa pesan telah sampai dan diterima.
4. SMSC mengirimkan laporan pengiriman kepada pengirim yang berisikan bahwa pesan telah terkirim.

Sedangkan, prinsip dari *SMS Gateway* di sini adalah sebuah perangkat yang memanfaatkan teknologi komputer sebagai media pengiriman SMS dan dapat diintegrasikan dengan berbagai aplikasi tergantung dengan apa yang dibutuhkan oleh penggunanya.

Fitur-fitur pada *SMS Gateway* yang biasanya dibutuhkan oleh penggunanya antara lain adalah:

- *SMS Broadcast*, digunakan untuk mengirimkan informasi kepada banyak pengguna secara langsung.
- *SMS Auto-reply*, menggunakan suatu perintah dimana nantinya pengguna akan mendapatkan informasi balasan otomatis sesuai dengan apa yang ingin dicari.
- *SMS Auto-send*, mengirimkan pesan secara otomatis dengan jumlah dan waktu yang ditentukan oleh pengirim itu sendiri.

Cara kerja *SMS Gateway* yaitu dengan memanfaatkan perangkat seperti modem untuk dijadikan server pengiriman SMS. Ada berbagai macam aplikasi perangkat lunak yang dapat digunakan dalam penggunaan *SMS Gateway* itu sendiri, misalnya seperti *SMS Service Gammu*, *PlaySMS*, *NowSMS*, dan lain-lain.

## 2.5 Gammu

Gammu adalah sebuah aplikasi *open source* berupa kumpulan kode / *script* yang berguna untuk mengatur fungsi pada telepon. Gammu sendiri merupakan satu dari sekian banyak aplikasi *SMS Gateway* yang cukup populer di kalangan penggunanya karena kelebihan dari Gammu itu sendiri yang antara lain adalah:

1. Gammu dapat dioperasikan di berbagai sistem operasi seperti Windows dan Linux.
2. Gammu adalah sebuah aplikasi *Open Source* atau gratis.
3. Gammu bersifat fleksibel, yaitu kompatibel dengan berbagai macam perangkat yang terhubung dengan kabel USB hingga serial.
4. Gammu dapat menggunakan berbagai macam aplikasi seperti MySQL dan ODBC sebagai *database*.

Berikut adalah sekumpulan perintah-perintah dasar yang biasa digunakan dalam Gammu.

**Tabel 2.2 Perintah Dasar Dalam Gammu**

No	Perintah	Deskripsi
1.	gammu -identify	Untuk mengecek koneksi dengan HP / Modem
2.	gammu-smsd -c smsdrc -i	Untuk meng- <i>install</i> service Gammu
3.	gammu-smsd -c smsdrc -s	Untuk menjalankan <i>service</i> Gammu
4.	gammu -sendsms TEXT 08xxxxxxxxxxx -text "coba"	Untuk mengirimkan pesan singkat
5.	gammu-smsd -c smsdrc -k	Untuk menghentikan <i>service</i> Gammu
6.	gammu-smsd -c smsdrc -u	Untuk meng- <i>uninstall</i> service Gammu

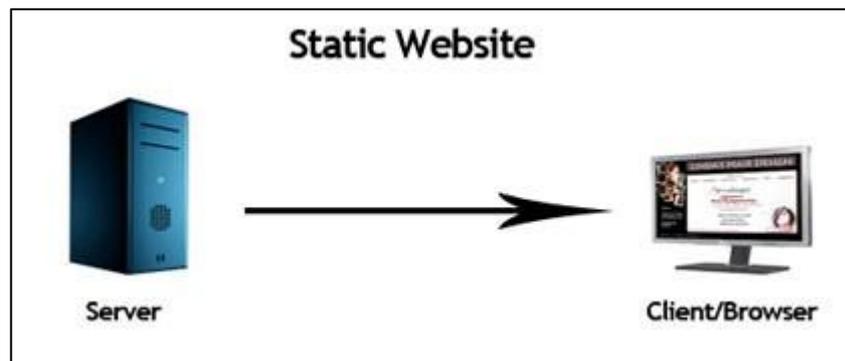
Untuk memudahkan pengguna dalam pengoperasiannya, dibutuhkanlah suatu kerangka kerja. Dikarenakan sifat Gammu yang fleksibel, terdapat banyak aplikasi yang dapat digunakan sebagai penyedia fungsionalitas secara umum seperti Microsoft Visual Studio, Java, maupun PHP.

## 2.6 Website

*Website* atau yang biasa ditulis dengan *web site*, sesuai dengan namanya yaitu merupakan sekumpulan halaman jaringan yang saling berhubungan, berisikan kumpulan informasi berupa konten multimedia seperti teks, gambar, audio, maupun video, dan diidentifikasi dengan suatu nama domain sebagai alamat jaringan. Sebuah situs jaringan dapat diakses melalui perangkat protokol jaringan seperti internet maupun jaringan wilayah local (*Local Area Network*), dengan menuliskan alamatnya atau biasa disebut dengan URL (*Uniform Resource Locator*).

Secara garis besar, situs jaringan dikategorikan menjadi tiga, yaitu:

1. Situs jaringan statis

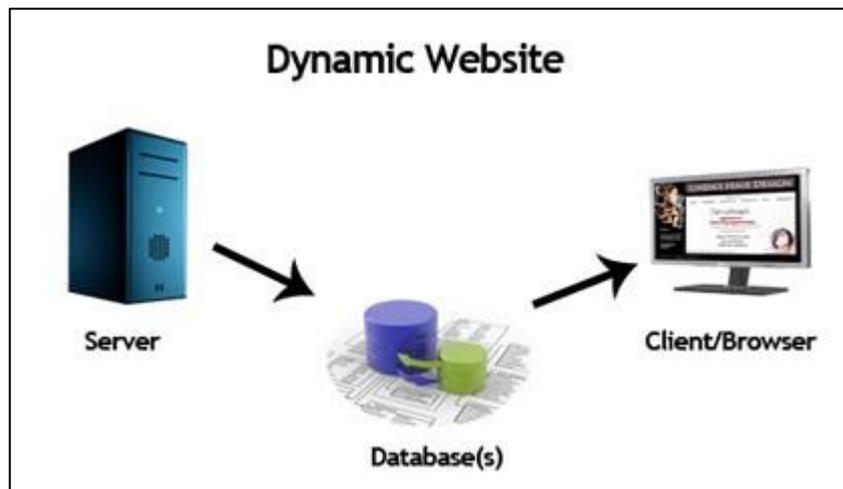


**Gambar 2.3 Website Statis**

*Website* statis merupakan situs jaringan yang pembuatannya dilakukan secara manual dengan cara menuliskan kode-kode html menjadi suatu halaman web. Konten dari *website* statis tidak dimaksudkan untuk diedit secara sewaktu-waktu, karena tentu sangat merepotkan jika mempunyai banyak halaman statis yang harus diupdate secara manual.

Web statis sangat cocok digunakan untuk demo contoh *website*, *website landing page* atau *mini website* untuk menampilkan promosi tertentu secara terpisah dari *website* utama. Beberapa pengembang menawarkan *website* murah, namun jika dicermati ternyata *website* yang mereka tawarkan adalah *website* statis yang perlu diedit secara manual dan harus ditulis ulang datanya jika ingin berganti tampilan.

## 2. Situs jaringan dinamis



**Gambar 2.4 Website Dinamis**

Pengertian dari *website* dinamis adalah sebuah situs jaringan yang bisa *update* secara mudah, sehingga isi dari situs tersebut bisa berubah sesuai keinginan pemilik *website* (dinamis). *Website* dinamis sendiri dibagi menjadi dua macam halaman:

- Halaman *Frontpage*: Halaman depan dari suatu situs. Merupakan halaman yang diakses oleh para pengunjung.
- Halaman *Backend*: Halaman situs yang hanya dapat diakses oleh pemilik dan pengembang *website*. Dari halaman inilah pengembang mengontrol seluruh isi dari *website*, termasuk jika ingin melakukan perubahan pada tampilan halaman depan.

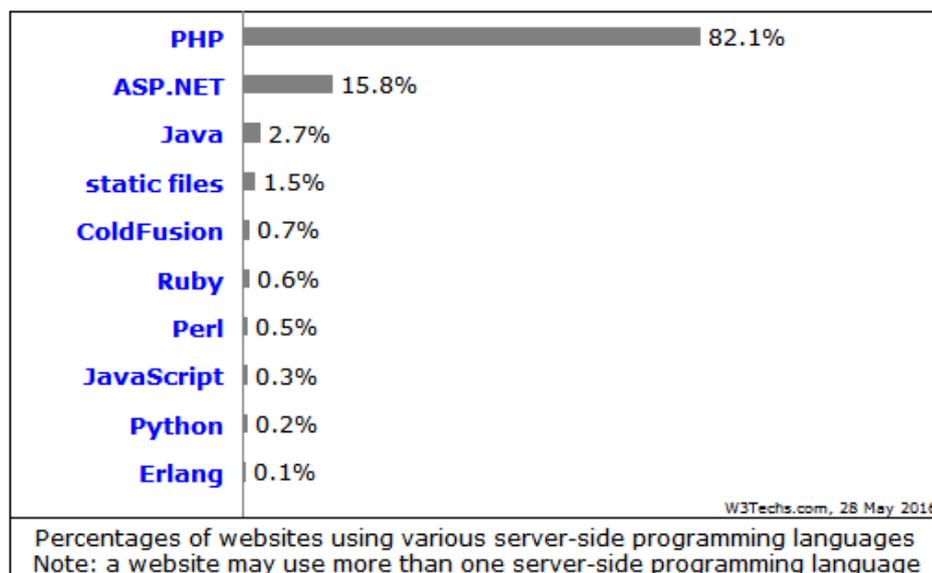
## 3. Situs jaringan interaktif

*Website* interaktif adalah pengembangan lebih lanjut dari situs jaringan dinamis, di mana dalam situs jaringan interaktif terjadi komunikasi dua arah antara pengunjung dan pengurus *website*, atau antara pengunjung dengan sesama pengunjung. Contohnya adalah seperti media sosial (Facebook, Twitter, dll) dan forum (Kaskus, CHIP, dll)

## 2.7 PHP

PHP yang merupakan kepanjangan "*PHP: Hypertext Preprocessor*", merupakan sebuah bahasa pemrograman yang pertama kali diciptakan pada tahun 1994 oleh Rasmus Lerdorf. PHP secara luas digunakan untuk mengembangkan dan membuat sebuah situs web dan biasanya bersamaan digunakan dengan bahasa dasar pemrograman web, yaitu HTML (*Hypertext Markup Language*).

Awalnya, PHP adalah kependekan dari *Personal Home Page*. Dan sesuai dengan namanya, PHP ini ditujukan untuk membuat sebuah *website* pribadi. Dan seiring dengan berjalannya waktu, PHP kini telah berubah menjadi sebuah bahasa pemrograman yang tidak hanya digunakan untuk membuat halaman web sederhana, namun juga *website* responsif yang dipakai oleh banyak orang dengan pengembangan beberapa *framework* seperti CodeIgniter, Bootstrap, Laravel, dan sebagainya. Kepopuleran dan kemudahan PHP kini ditetapkan menjadi standar bagi beberapa *web programmer* di seluruh dunia. Beberapa aplikasi CMS (*Content Management System*) yang banyak dikenal orang seperti WordPress, Joomla, dan Drupal, adalah berdasar dari pengembangan bahasa pemrograman PHP itu sendiri. Menurut W3Techs pada Mei 2016, sekitar 82.1% dari web server di dunia menggunakan PHP [10].



Gambar 2.5 Hasil Survey Penggunaan Bahasa Programming Website

## 2.8 SQL

SQL atau *Structured Query Language* adalah kumpulan-kumpulan instruksi yang dipakai untuk manajemen data pada basis data relasional, dengan mengikuti standard ANSI (*American Nasional Standard Institute*). Dengan menggunakan SQL, kita dapat menjalankan perintah atau *query* untuk mengakses, menambah, mengubah, dan menghapus data di dalam basis data. Kini hampir setiap server *database* yang ada mendukung SQL dalam melakukan pengolahan datanya. Ada 3 (tiga) jenis perintah dari SQL, yaitu Bahasa Pendefinisian Data, Bahasa Pemanipulasian Data, Bahasa Pengontrolan Data.

### 1. DDL atau *Data Definition Language*

Bahasa Pendefinisian Data adalah instruksi SQL yang menjabarkan mengenai pendefinisian suatu struktur *database*, dalam hal ini adalah basis data dan tabel. Perintah SQL yang termasuk dalam DDL antara lain:

- CREATE
- DROP
- ALTER

### 2. DML atau *Data Manipulation Language*

Bahasa Pemanipulasian Data adalah instruksi SQL yang menjabarkan mengenai pemanipulasian atau pengolahan suatu data dalam suatu tabel. Perintah SQL yang termasuk dalam DML antara lain:

- INSERT
- DELETE
- UPDATE
- SELECT

### 3. DCL atau *Data Control Language*

Bahasa Pengontrolan Data adalah instruksi SQL mengenai mengatur hak akses pengguna, baik terhadap *server*, *database*, *table* maupun *field*. Perintah SQL yang termasuk dalam DCL antara lain:

- REVOKE
- GRANT

## 2.9 Database

*Database* atau basis data merupakan sekumpulan data yang terorganisasi, sehingga dapat dengan mudah diakses, dikelola, diperbarui, dan diolah menjadi sebuah informasi yang dapat dimanfaatkan oleh penggunanya. Data-data tersebut disimpan di dalam perangkat seperti komputer secara sistematis, sehingga dapat diperiksa dengan beberapa perangkat lunak untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang dipakai untuk memanggil perintah (*query*) dan mengelola basis data tersebut disebut juga dengan sistem manajemen basis data (*Database management system*).

Sebuah basis data mempunyai penjelasan yang terorganisir dari jenis fakta yang tersimpan di dalamnya, yang biasa disebut dengan skema. Isi dari sebuah database adalah kumpulan dari tabel yang berisi kolom (*field*) dan baris (*record*) di mana tiap kolom berisikan atribut yang spesifik, dan tiap baris merepresentasikan nilai dari atribut yang terkait. Jumlah dari kolom dalam satu tabel bergantung dari seberapa banyak kategori yang berbeda dari informasi yang dibutuhkan untuk disimpan ke dalam database. Sedangkan jumlah dari baris didefinisikan sebagai kuantitas dari objek yang terdapat dari kategori tersebut.

Tujuan dari basis data sendiri yaitu sebagai berikut:

- Keakuratan data, demi menghindari inkonsistensi data,
- Ketersediaan data, dapat menyimpan dan *backup* data yang sudah hilang dari sisi pengguna,
- Kelengkapan data, dapat merubah struktur basis data baik itu menambah objek / tabel baru, maupun *field* yang ada pada tabel,
- Keamanan data, dapat menentukan hak akses pengguna yang dapat menggunakan basis data dan objek yang ada di dalamnya,
- Kebersamaan pengguna, dapat digunakan lebih dari satu pemakai,
- Kemudahan dan kecepatan dalam menyimpan, memanipulasi, serta menampilkan kembali data,
- Efisiensi ruang penyimpanan dengan menghindari redudansi data.

Sedangkan sistem basis data, yang merupakan perpaduan dari basis data dan sistem manajemen basis data, adalah sistem yang terdiri dari sekumpulan koleksi data yang saling berelasi. Tujuan dari sistem basis data itu sendiri adalah mengarahkan suatu lingkungan yang efisien dalam pengaksesan, penyimpanan, dan pengambilan informasi basis data, serta penyediaan antarmuka yang dapat lebih mudah digunakan oleh pengguna.

Komponen dari sistem basis data sendiri antara lain:

- Perangkat keras, seperti prosesor, memori, cakram keras, terminal.
- Perangkat lunak aplikasi, seperti VB, Delphi, Foxpro.
- Sistem operasi untuk menghubungkan kedua perangkat di atas. (Windows, Linux, Mac OS)
- Pengguna (*user*).
- Basis data dan perangkat lunak pengelola basis data, seperti SQL, Oracle, MS-Access. Perangkat inilah yang disebut dengan DBMS (sistem manajemen basis data), perangkat yang menentukan bagaimana data disimpan, dikelola, diakses kembali, serta penerapan mekanisme pengamanan dan penggunaan data secara bersamaan.

## 2.10 Diagram UML

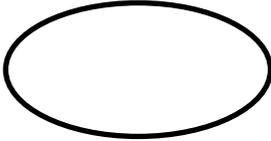
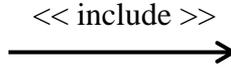
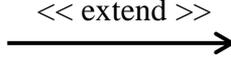
UML atau kependekan dari *Unified Modelling Language* merupakan bahasa yang dipakai dalam teknik perangkat lunak berorientasi objek, yang digunakan untuk mendeskripsikan dan menggambarkan rangkaian analisa, proses bisnis, serta memodelkan aplikasi yang terstruktur. Singkatnya, UML diagram biasa digunakan dalam memodelkan suatu sistem. [11]

Terdapat beberapa jenis-jenis diagram UML antara lain:

### 1. Use Case Diagram

*Use case* diagram merupakan salah satu yang paling dikenal dalam diagram UML. Secara dasar, diagram ini menjelaskan mengenai suatu pengguna (aktor) berinteraksi dengan sistem (*use case*) dengan berbagai fungsi yang berbeda.

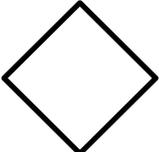
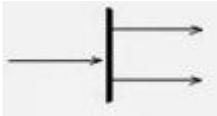
Tabel 2.3 Simbol Dalam Diagram *Use Case* [12]

No	Simbol	Keterangan
1.		Aktor: Yaitu adalah subjek atau orang yang berinteraksi dengan sistem.
2.		<i>Use case</i> : Suatu kegiatan atau aktivitas dalam suatu sistem.
3.		Asosiasi: Garis penghubung menunjukkan aktor melakukan suatu aktivitas.
4.		Generalisasi: Menunjukkan spesialisasi aktor dapat berpartisipasi dalam aktivitas tertentu pada <i>use case</i> .
5.		<i>Include</i> : Penghubung dimana <i>use case</i> yang akan dijalankan harus melewati proses <i>use case</i> sebelumnya.
6.		<i>Extend</i> : Penghubung dimana <i>use case</i> yang akan dijalankan dapat berjalan sendiri walau tanpa <i>use case</i> sebelumnya.

## 2. Activity Diagram

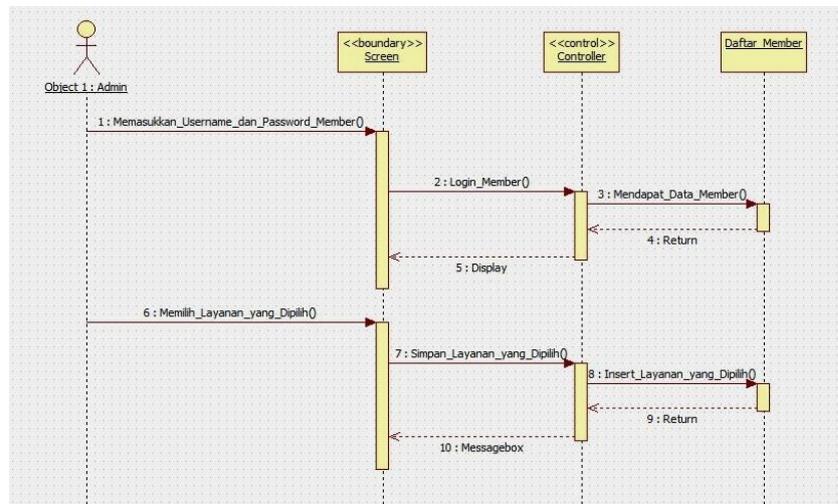
*Activity* diagram merepresentasikan alur kerja secara grafik. Diagram ini dapat digunakan untuk mendeskripsikan proses bisnis, maupun alur operasional tiap komponen dalam sebuah sistem. Pada dasarnya, *activity* diagram berfungsi untuk menggambarkan alur kerja yang berjalan pada sebuah sistem, menunjukkan bagaimana aktivitas berlangsung, dan bagaimana aktivitas tersebut selesai.

Tabel 2.4 Simbol Dalam *Activity Diagram* [12]

No	Simbol	Keterangan
1.		<i>Start point</i> : Status awal dari aktivitas sebuah sistem.
2.		<i>End point</i> : Status akhir dari aktivitas sebuah sistem.
3.		<i>Activity</i> : Proses yang dikerjakan suatu sistem.
4.		<i>Decision</i> : Percabangan suatu aktivitas dimana terdapat satu atau lebih aktivitas pilihan.
5.		<i>Fork / Percabangan</i> : Dipakai dalam menunjukkan kegiatan yang dikerjakan secara paralel.
6.		<i>Join / Penggabungan</i> : Dipakai dalam menunjukkan kegiatan yang digabungkan.

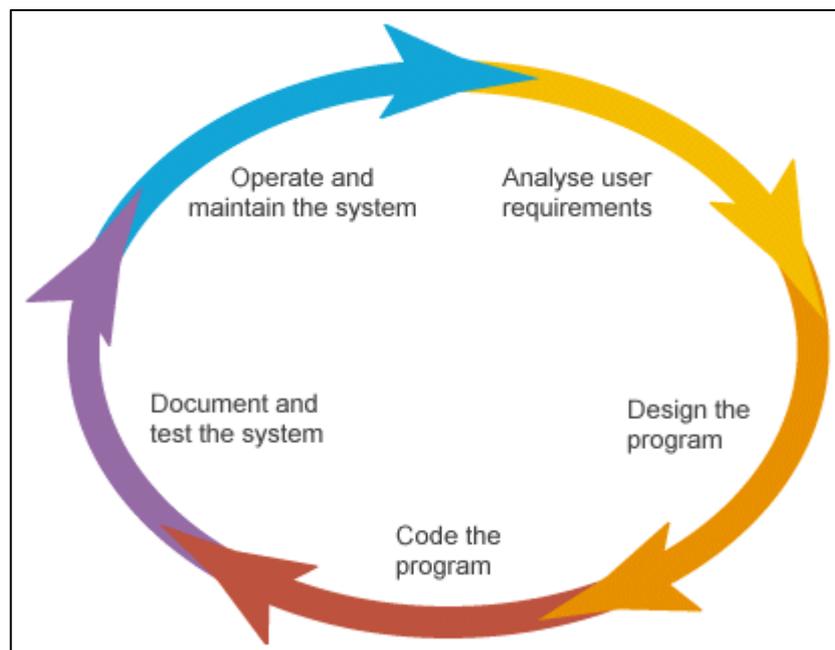
### 3. *Sequence Diagram*

*Diagram sequence* merupakan salah satu diagram interaksi yang menjelaskan bagaimana suatu operasi dilakukan. Digunakan untuk mendeskripsikan interaksi antar objek, dan disusun berdasarkan sesuai berjalannya waktu mengenai kapan harus berjalan, dan apa yang akan dilakukan. Objek yang berhubungan dengan proses operasional, berjalan urut dari kiri ke kanan. Singkatnya, diagram ini adalah visualisasi tiap tahap yang dilakukan sesuai dengan *use case* diagram untuk menghasilkan suatu proses.



**Gambar 2.6** Contoh *Sequence Diagram* [12]

## 2.11 System Development Life Cycle



**Gambar 2.7** *System Development Life Cycle*

*System Development Life Cycle* (SDLC) adalah metodologi klasik yang digunakan untuk mengembangkan, memelihara dan menggunakan sistem informasi. Siklus hidup sistem itu sendiri merupakan metodologi, tetapi polanya lebih dipengaruhi oleh kebutuhan untuk mengembangkan sistem yang lebih cepat.

Pengembangan sistem yang lebih cepat dapat dicapai dengan peningkatan siklus hidup dan penggunaan peralatan pengembangan berbasis komputer. Terdapat 5 fase dalam penerapan SDLC sendiri, dimulai dari menganalisa kebutuhan pengguna, hingga mengimplementasikan sistem itu sendiri.

### 1. Analisa

Tahap analisa adalah proses mengumpulkan data-data yang faktual, memahami proses yang sedang berjalan, mengidentifikasi masalah, dan merekomendasikan sugesti secara mendasar untuk meningkatkan fungsi sistem. Melibatkan pemahaman akan proses bisnis, mengumpulkan data operasional, mengerti akan alur informasi yang berjalan, mencari tau yang tidak sesuai dalam permasalahan, dan mengembangkan solusi akan kelemahan sistem yang sekarang, guna mencapai tujuan organisasi. Target utama dalam analisa sistem adalah menjawab pertanyaan-pertanyaan proses bisnis seperti:

- Apa yang harus dilakukan?
- Bagaimana cara melakukannya?
- Siapa yang melakukannya?
- Kapan dan mengapa harus dilakukan?
- Bagaimana agar dapat lebih berkembang?

Proses ini lebih memerlukan kemampuan berpikir dan kreativitas dari analis sistem. Proses analisa sistem adalah proses iteratif yang terus berjalan hingga mendapatkan solusi yang dapat menyelesaikan masalah.

### 2. Desain

Berdasarkan dari kebutuhan dan rincian dari analisa suatu sistem yang lama, perlu adanya desain sistem yang baru. Fase ini disebut dengan desain sistem (*system design*), yaitu fase paling krusial dalam pengembangan sistem. Proses desain sistem pada SDLC berlanjut dari pertanyaan “Apa yang harus dilakukan” menjadi “Bagaimana cara melakukannya”. Pemikiran desain logis pada tahap analisa diubah menjadi pemikiran desain fisik, yaitu deskripsi secara rinci dari apa yang dibutuhkan untuk menyelesaikan masalah utama, dari *input*, *output*, *database*, *form*, skema kerja, hingga spesifikasi proses

digambarkan secara rinci. Tahap yang dilakukan di sini adalah menentukan struktur data, proses kontrol, sumber peralatan, beban dan limitasi suatu sistem, desain antar muka, dokumentasi, prosedur yang akan digunakan sistem, dan kebutuhan akan pekerja.

Beberapa alat dan teknik yang dapat digunakan dalam tahap ini antara lain: *Flowchart*, DFD (*Data Flow Diagram*), kamus data, dan tabel-pohon keputusan.

### 3. Konstruksi

Konstruksi di sini merupakan proses *Coding*. Sebuah desain sistem tentunya harus diimplementasikan agar dapat bekerja, dalam hal ini diperlukanlah perubahan dari desain tadi menjadi suatu bahasa komputer, atau lebih dikenal dengan bahasa pemrograman. Tahap ini disebut juga dengan tahap *programming* atau tahap *coding*, dijalankan oleh seseorang yang disebut dengan *programmer*. Tahap ini merupakan tahapan yang penting di mana prosedur yang sudah didefinisikan tadi ditransformasikan menjadi spesifikasi kontrol, digunakan sebagai mengkoordinasi pergerakan data dan mengontrol seluruh proses dalam suatu sistem.

### 4. Testing

Setelah pengkodean selesai, sebelum mengimplementasikan dan mengoperasikan sistem yang baru suatu pengujian biasanya dilakukan terlebih dahulu untuk mencari *error* atau *bugs* pada suatu program. Tahap ini harus dilakukan, jika ingin memperoleh suatu sistem yang sukses. Fase testing biasanya dilakukan dalam dua tahap:

- *Program Test*: Ketika program telah selesai dikerjakan, fungsi dari tiap bagian dites dengan beberapa data *dummy*. Seluruh tahap harus dilakukan, dicek, diverifikasi dan divalidasi untuk menghindari hal-hal yang tidak diinginkan.
- *System Test*: Setelah dites tiap-tiap fungsinya, program dites secara utuh / menyeluruh. Pengujian sistem dilakukan menggunakan data yang aktual, apakah hasil yang akan ditampilkan sesuai dengan sistem yang sudah direncanakan.

## 5. Implementasi

Tahapan implementasi dapat disebut juga sebagai tahap akhir dalam pengembangan suatu sistem. Setelah semua proses analisa, desain, *coding*, dan pengujian selesai, maka fase menggantikan sistem yang lama dengan menerapkan sistem yang baru dimulai.

Pada tahap ini, semua program yang telah dikembangkan diterapkan ke komputer pengguna. Tidak hanya itu, dokumentasi juga perlu dilakukan untuk membantu pengguna sistem yang baru memahami bagaimana kerja dari program tersebut, yang dasarnya terdiri dari:

- Bagaimana menjalankannya?
- Bagaimana untuk memasukan data?
- Bagaimana memproses data yang ingin diolah?

### 2.12 Black Box Testing

Pengujian dengan menggunakan metode *black box* adalah pengujian kebutuhan dan hanya berfokuskan kepada fungsional dari sebuah perangkat lunak atau program. Pengujian ini dilakukan untuk meyakinkan semua *input* diterima dengan tepat, dan *output* yang dihasilkan juga tepat dan berjalan dengan baik. Dengan kata lain, metode pengujian *black box* adalah untuk mengetes hubungan antar program dalam sebuah sistem.

*Black box testing* tidak memerlukan pengetahuan akan alur internal, struktur atau implementasi dari pengujian perangkat lunak. Karena itu pengujian *black box* memungkinkan pengembang aplikasi untuk membuat himpunan kondisi input yang akan mengecek seluruh syarat-syarat fungsional suatu program.

Metode *black box* dapat digunakan untuk menemukan beberapa kesalahan-kesalahan suatu program pada beberapa kategori seperti:

- Kesalahan pada inisialisasi dan terminasi,
- Kesalahan pada struktur data,
- Kesalahan pada pengaksesan *database*,
- Kesalahan pada performa aplikasi,

- Kesalahan pada antar muka,
- Kesalahan pada fungsi yang tidak benar atau hilang.

Ciri-ciri dari *black box testing* antara lain:

- Lebih berfokuskan kepada kebutuhan fungsional berdasarkan spesifikasi kebutuhan suatu perangkat lunak.
- Melakukan pengetesan tanpa detil struktur internal dari suatu perangkat yang diuji.
- Merupakan pendekatan komplemen dalam mencakup *error* dengan kelas yang berbeda.

Sehingga dapat disimpulkan kelebihan dan kelemahan *black box testing* adalah sebagai berikut:

**Tabel 2.5 Kelebihan dan kelemahan *black box testing***

Kelebihan	Kelemahan
Perincian aplikasi dapat ditentukan di awal, dan pengujian dilakukan berdasarkan perincian spesifikasi tersebut.	Apabila keperluan perangkat lunak yang akan dikembangkan tidak begitu jelas, pembuatan dokumentasi yang tepat akan sedikit sulit.
Dapat dipakai untuk menilai konsistensi suatu aplikasi, dan tidak perlu melihat kode program secara detil.	Pengguna akan kurang merasa yakin dengan perangkat lunak yang diuji apakah lolos dalam standar pengujian.