

BAB II

LANDASAN TEORI

2.1 SMS (*Short Message Service*)

Menurut Wahana Komputer (2005 : 12) mengartikan, SMS (Short Message Service) adalah merupakan salah satu layanan pesan teks yang dikembangkan dan distandardisasi oleh suatu badan bernama ETSI (European Telecommunication Standards Institute) sebagian dari pengembangan GSM (Global System for Mobile Communication) Phase 2, yang terdapat pada dokumentasi GSM 03.40 dan GSM 03.38. Fitur SMS ini memungkinkan perangkat Stasiun Seluler Digital (Digital Cellular Terminal, seperti Ponsel)

untuk dapat mengirim dan menerima pesan-pesan teks dengan panjang sampai dengan 160 karakter melalui jaringan GSM. 22 Karakter yang dimaksud adalah alphabet A sampai Z, angka 0 sampai 9 dan spasi. Untuk karakter non-Latin, seperti Arab, Kanji atau Mandarin dengan panjang sampai dengan 70 karakter. SMS dapat dikirimkan ke perangkat stasiun seluler digital lainnya hanya dalam beberapa detik selama berada pada jangkauan pelayanan GSM.

Lebih dari sekedar pengiriman pesan biasa, layanan SMS memberikan garansi SMS akan sampai pada tujuan meskipun perangkat yang dituju sedang tidak aktif yang dapat disebabkan karena sedang dalam kondisi mati atau berada di luar jangkauan layanan GSM. Dengan adanya feature seperti ini maka layanan SMS juga cocok untuk dikembangkan sebagai aplikasi-aplikasi seperti : pager, e-mail, dan notifikasi voice mail, serta layanan pesan banyak pemakai (multiple user). Namun pengembangan aplikasi tersebut masih bergantung pada tingkat layanan yang disediakan oleh operator jaringan.

2.2 Sistem

Tata Sutabri (2012,p.10) mendefinisikan Sistem adalah sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.

Menurut Tata Sutabri(2012,p.20) sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu :

1. Komponen sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar atau sering disebut “supra sistem”.

2. Batasan sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan luar sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak, maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan sistem (*Input*)

Energi yang dimasukkan kedalam sistem tersebut masukkan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Contoh, di dalam suatu unit sistem komputer, "Program" adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan "Data" adalah signal input untuk diolah menjadi informasi.

6. Keluaran sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

7. Pengolahan sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran dan tujuan yang telah direncanakan.

2.3 Konsep Dasar Informasi

2.3.1 Data

Menurut Edhy Sutanta(2004,p.4) data dapat didefinisikan sebagai bahan keterangan tentang kejadian-kejadian nyata atau fakta-fakta yang dirumuskan dalam kelompok lambang tertentu yang tidak acak yang menunjukkan jumlah, tindakan, atau hal.

2.3.2 Informasi

Menurut Tata Sutabri(2012,p.29)Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari tak berguna menjadi berguna bagi penerimanya.

Nilai informasi berhubungan dengan keputusan maka informasi menjadi tidak diperlukan keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks sebuah keputusan.

2.4 World Wide Web (WWW)

Kustiyahningsih & Anamisa (2011,p.4) mengatakan *World Wide Web* (WWW), lebih dikenal dengan web yang merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet dengan fasilitas *hypertext* untuk menampilkan data berupa *text*, gambar, suara, animasi dan data multimedia lainnya. Sehingga web pada awalnya adalah ruang informasi dalam internet, dengan menggunakan teknologi *hypertext*, pemakai dituntut menemukan informasi dengan mengikuti *link* yang disediakan dalam dokumen web yang ditampilkan dalam *web browser*. Situs atau web dapat dikategorikan menjadi dua yaitu :

1. Web Statis

Web statis adalah web yang menampilkan informasi-informasi yang sifatnya statis(tetap). Disebut statis karena pengguna tidak dapat berinteraksi dengan web tersebut.

2. Web Dinamis

Web dinamis adalah web yang menampilkan informasi serta dapat berinteraksi dengan pengguna. Web yang dinamis memungkinkan pengguna untuk berinteraksi menggunakan form sehingga dapat mengolah informasi yang ditampilkan. Web dinamis bersifat interaktif, tidak kaku dan terlihat lebih indah.

2.5 URL

Menurut Kustiyahningsih & Anamisa (2011,p.7) mengatakan URL (Universal Resource Locator) adalah konsep nama file standar yang diperluas dengan jaringan untuk menentukan lokasi informasi pada *web browser*. Nama file ini tidak hanya menunjukkan direktori dan nama filenya, tetapi juga nama mesinnya dalam jaringan.

URL dapat disediakan (ada atau diakses) dengan berbagai metode, dan bisa jadi bukan sekedar file karena URL dapat menunjukkan query-query, dokumen yang disimpan dalam database, hasil dari suatu perintah *finger* atau *archie* atau apapun yang berkaitan dengan data hasil proses.

2.6 HTTP

Menurut Kustiyahningsih & Anamisa (2011,p.7) HTTP (*Hypertext Transfer Protocol*) adalah suatu protokol yang menentukan aturan yang perlu diikuti oleh *web browser* dalam meminta atau mengambil suatu dokumen dan menyediakan dokumen yang diminta oleh *browser*. Protokol ini merupakan protokol standar yang digunakan untuk mengakses halaman HTML. Server HTTP umumnya digunakan untuk melayani dokumen *hypertext*, karena HTTP adalah protokol dengan overhead yang sangat rendah, sehingga pada

kenyataannya navigasi informasi dapat ditambahkan langsung kedalam dokumen.

2.7 Web Browser

Menurut Kustiyahningsih & Anamisa (2011,p.8) *web browser* adalah software yang digunakan untuk menampilkan informasi dari server web. Software ini kini telah dikembangkan dengan menggunakan *user interface grafis*, sehingga pemakai dapat dengan melakukan '*point and click*' untuk pindah antar dokumen.

Dapat dikatakan saat ini hanya ada empat *web browser* GUI yang populer yaitu : Internet Explorer, Netscape Navigator, Opera dan Mozilla.

2.8 Perangkat Lunak yang Digunakan

Untuk membangun *website* diperlukan beberapa perangkat lunak yang digunakan dalam membangun *website* tersebut. Beberapa perangkat lunak yang digunakan adalah sebagai berikut :

2.8.1 Adobe Dreamweaver CC

Menurut Sigit (2010, p.1) *Adobe Dreamweaver CC* adalah sebuah HTML editor profesional untuk mendesain web secara visual dan mengelola situs atau halaman web. Saat ini terdapat *software* dari kelompok *Adobe* yang belakangan banyak digunakan untuk mendesain suatu *web*. Versi terbaru dari *Adobe Dreamweaver CC* memiliki beberapa kemampuan bukan hanya sebagai *software* untuk desain *web* saja, tetapi juga menyunting kode serta pembuatan aplikasi *web*. Antara lain: JSP, PHP, ASP, XML, dan *ColdFusion*.

Adobe Dreamweaver CC merupakan.

Software utama yang digunakan oleh *Web Designer* dan *Web Programmer* dalam mengembangkan suatu situs *web*. Hal ini disebabkan oleh ruang kerja, fasilitas, dan kemampuan *Dreamweaver*

yang mampu meningkatkan produktivitas dan efektifitas, baik dalam desain maupun membangun suatu situs *web*.

2.8.2 Notepad ++

Selain menggunakan aplikasi *Dreamweaver*, peneliti juga menggunakan aplikasi notepad ++ dikarenakan tampilan interface aplikasi yang lebih mudah dipahami dalam melihat source program. Notepad++ merupakan sebuah penyunting teks dan penyunting kode sumber yang tersedia dalam beberapa sistem operasi seperti Windows, Linux, dan Mac OS X. Notepad ++ menggunakan komponen Scintilla untuk dapat menampilkan dan menyunting teks dan berkas kode sumber berbagai bahasa pemrograman. Notepad ++ didistribusikan sebagai perangkat lunak bebas. Proyek ini dilayani oleh *Sourceforge.net* dengan telah diunduh lebih dari 27 juta kali dan dua kali memenangkan penghargaan *SorceForge Community Choice Award for Best Developer Tools*.

2.8.3 Framework Bootstrap

Menurut Husein (2013, p.4) *Bootstrap* merupakan *framework* ataupun *tools* untuk membuat aplikasi *website* ataupun situs *web responsive* secara cepat, mudah dan gratis, karena *website* yang dibangun oleh peneliti merupakan *website* yang dapat diakses dalam perangkat *mobile* ataupun *personal computer*. *Bootstrap* terdiri dari CSS dan HTML untuk menghasilkan *Grid, Layout, Typography, Table, Form, Navigation*, dan lain lain. Di dalam *bootstrap* juga sudah terdapat *jquery plugins* untuk menghasilkan komponen UI yang cantik seperti *Transitions, Modal, Dropdown, Scrollspy, Tooltip, Tab, Popever, Alert, Button, Carousel*, dan lain lain.

2.8.4 XAMPP

Library IBM (2004), XAMPP merupakan aplikasi server yang menggabungkan beberapa aplikasi *server* yang biasa digunakan di *web server*. Berikut beberapa komponen – komponen yang terdapat pada XAMPP, yaitu Apache (*web server*), MySQL (*database server*), Filezilla FTP *server*, Mercury Mail (*mail server*), phpMyAdmin (*web-based interface MySQL*)

2.8.5 GAMMU Library

Gammu merupakan salah satu tools untuk mengembangkan aplikasi SMS Gateway yang cukup mudah diimplementasikan dan pastinya gratis. Kelebihan Gammu dari tool sms gateway lainnya adalah :

- a. Gammu bisa di jalankan di Windows maupun Linux.
- b. Gammu menggunakan database MySQL, Bisa menggunakan interface web based.
- c. Baik kabel data USB maupun SERIAL, semuanya kompatibel di Gammu.
- d. Dapat membaca, menghapus dan mengirim SMS.
- e. Membaca menulis dan menghapus isi inbox.

2.9 Database

2.9.1 Definisi Database

Menurut Verdi Yasin (2012,p.274) mengatakan Basis data (*database*) adalah kumpulan informasi yang akan disimpan didalam komputer secara sistematis, sehingga dapat digunakan oleh suatu program komputer untuk memperoleh informasi dari basis data tersebut. Basis data adalah sekumpulan data yang terhubung satu sama lain secara logika dan suatu deskripsi data yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. *Database*

merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna atau user.

2.9.2 Tujuan Database

1. Kecepatan dan Kemudahan (*Speed*)

Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah.

2. Efisiensi Ruang Penyimpanan(*Space*)

Karena keterkaitan erat antara kelompok dalam basis data, maka redundansi data pasti selalu ada. Dengan basis data,efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan karena kita dapat melakukan penekanan jumlah redundansi data, baik menerapkan sejumlah pengkodean atau membuat relasi-relasi antar kelompok data yang saling berhubungan.

3. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan tipe data,domain data,keunikan data dan sebagainya yng secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan/penyimpanan data.

4. Ketersediaan (*Availability*)

Pertumbuhan data sejalan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data selalu kita gunakan/butuhkan. Karena itu kita dapat melakukan pemilihandata, sehingga data yang sudah jarang kita gunakan dapat kita pindahkan kedalam media penyimpanan offline.

5. Kelengkapan (*Completeness*)

Untuk mengkomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat menambah *record* data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam penambahan objek baru (tabel) atau dengan penambahan *field* baru pada suatu tabel.

6. Keamanan (*Security*)

Ada sejumlah sistem pengolahan basis data yang tidak menerapkan aspek keamanan dalam sebuah basis data.

7. Kebersamaan pemakain (*Sharebility*)

Pemakai basis data seringkali tidak terbatas pada satu pemakai saja atau di satu lokasi saja oleh satu sistem aplikasi.

2.10 Penelitian terdahulu

Tabel 2.1 berikut ini adalah beberapa penelitian terdahulu yang berkaitan dengan penerapan SMS Gateway.

Tabel 2.1 Penelitian Terdahulu

Nama	Judul	Terbit / Tahun	Keterangan
Apri Trisnawati	SISTEM LAYANAN BERITA BERBASIS SMS GATEWAY PADA LAMPUNG POST	SMS Gateway, SDLC, Layanan Berita 2014	Lampung Post merupakan perusahaan yang bergerak dalam bidang penerbitan surat kabar. Sehingga teknologi komputer sebagai alat bantu pemrosesan data menjadi sangat di perlukan bagi perusahaan. Dengan bantuan sistem komputer, maka kinerja perusahaan dapat di tingkatkan. Sehingga kualitas mutu pelayanan dan produk perusahaan menjadi lebih baik. SMS merupakan salah satu <i>feature</i> untuk sarana saling

		<p>berkirim pesan yang ada pada telpon genggam. Yang dapat dipadukan dengan sistem komputer dan dapat menghasilkan sistem sms yang otomatis, yang disebut sistem SMS Gateway.</p> <p>Metode Pengembangan sistemnya menggunakan metode Software Development Life Cycle (SDLC) dengan pendekatan model waterfall (pressman). Meliputi analisis, desain, kode, testing.</p> <p>Penelitian ini merancang sistem sms gateway yang bertujuan untuk meningkatkan jangkauan penjualan dan melihat pembacanya. Dan berimbas pada naiknya pelanggan dan juga pemasangan iklan. Sistem layanan berita ini digunakan sebagai media informasi yang bersifat topik singkat dan jelas. Layanan berita atau <i>Headline</i> yang akan diterbitkan, akan disampaikan kepada pelanggan lampung post yang telah menyetujui untuk menerima layanan berita dari pihak lampung post.</p>
--	--	--

Abdurrahman Masykur	Rancang Bangun sms gateway untuk mendukung layanan akademik pada pondok pesantren Al-Muhsin Metro	Gateway, Short Message Service, Data Flow Diagram, 2012	<p>Kemajuan dan keunggulan teknologi informasi yang semakin canggih mampu memenuhi kebutuhan manusia yang sulit dilakukan secara manual. SMS gateway digunakan untuk membantu mengelola penerimaan dan pengiriman informasi melalui SMS sehingga distribusi penyampaian informasi lebih mudah dan cepat. Sistem Informasi akademik yang dilengkapi SMS gateway dapat membantu pihak sekolah maupun wali murid dalam penyampaian informasi tentang murid, terutama informasi yang berkaitan dengan nilai dan absensi.</p> <p>Penguumpulan data dilakukan dengan metode : Observasi dan studi pustaka. Sistem dirancang dalam beberapa tahapan yaitu pembuatan DFD, rancangan basis data, relasi antar tabel, dan rancangan interface.</p>
---------------------	---	---	--

2.11 Metode String Matching

String matching adalah pencarian sebuah pattern pada sebuah teks (Cormen, T.H. et al. 1994). String matching digunakan untuk menemukan suatu string yang disebut dengan pattern dalam string yang disebut dengan teks (Charras, C. & Lecroq, T. 1997). Prinsip kerja algoritma string matching (Effendi, D. et al. 2013) adalah sebagai berikut: 1. Memindai teks dengan bantuan sebuah window yang ukurannya sama dengan panjang pattern. 2. Menempatkan window pada awal teks. 3. Membandingkan karakter pada window dengan karakter dari pattern. Setelah pencocokan (baik hasilnya cocok atau tidak cocok) dilakukan pergeseran ke kanan pada window. Prosedur ini dilakukan berulang-ulang sampai window berada pada akhir teks. Mekanisme ini disebut mekanisme sliding window. Algoritma string matching mempunyai tiga komponen utama (Effendi, D. et al. 2013), yaitu:

1. *Pattern*, yaitu deretan karakter yang akan dicocokkan dengan teks, dinyatakan dengan $x[0..m-1]$, panjang pattern dinyatakan dengan m .
2. Teks, yaitu tempat pencocokan pattern dilakukan. Dinyatakan dengan $y[0..n-1]$, panjang teks dinyatakan dengan n .
3. Alfabet, berisi semua simbol yang digunakan oleh bahasa pada teks dan pattern, dinyatakan dengan Σ dengan ukuran dinyatakan ASIZE.

2.11.1 Algoritma Horspool

Algoritma Horspool adalah penyederhanaan dari algoritma *Boyer-Moore* yang dibuat oleh R. Nigel Horspool. Menurut Horspool, R.N. (1980), masalah dalam pencarian teks ini adalah mencari dalam teks yang besar untuk menemukan pattern pertama. Karena teks yang dicari bisa sangat besar (memungkinkan ratusan ribu karakter) maka penting untuk menggunakan teknik yang lebih efisien. Algoritma *Horspool* bekerja dengan metode yang hampir sama dengan algoritma *Boyer-Moore* namun tidak melakukan lompatan berdasarkan karakter pada pattern yang ditemukan tidak cocok pada teks.

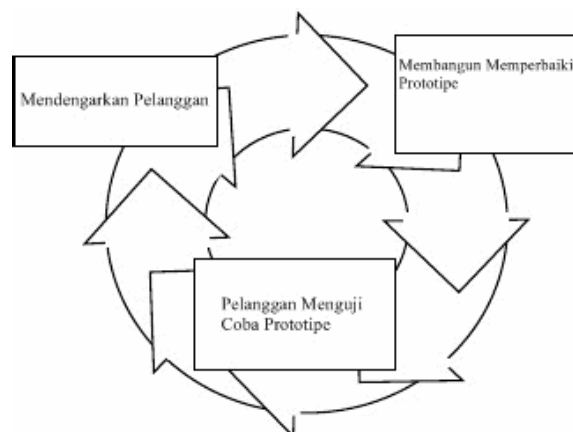
Algoritma Horspool mempunyai nilai pergeseran karakter yang paling kanan dari *windows*. Pada tahap observasi awal (preprocessing), nilai shift akan dihitung untuk semua karakter. Pada tahap ini, dibandingkan pattern dari kanan ke kiri hingga kecocokan atau ketidakcocokan pattern terjadi. Karakter yang paling kanan pada *windows* digunakan sebagai indeks dalam melakukan nilai shift. Dalam kasus ketidakcocokan (karakter tidak terdapat pada *pattern*) terjadi, *windows* digeser oleh panjang dari sebuah pattern. Jika tidak, *windows* digeser menurut karakter yang paling kanan pada pattern (Baeza-Yates, R.A. & Regnier, M. 1992).

2.12 Metode Pengembangan Perangkat Lunak

2.12.1 Metode *Prototype*

Mohammad Dastbaz (2012, p.118) menyatakan bahwa *Prototyping paradigma* dimulai dengan pengumpulan kebutuhan, pengembang bertemu dengan pengguna dan mengidentifikasi objektif keseluruhan dari perangkat lunak, selanjutnya mengidentifikasi segala kebutuhan yang diketahui secara garis besar di mana definisi-definisi lebih jauh merupakan keharusan, kemudian dilakukan perancangan kilat, lalu diakhiri dengan evaluasi *prototyping*.

Tahap-tahap rekayasa *software* dalam *prototype model* adalah seperti pada gambar 2.1 berikut.



Gambar 2.1 Model Prototip

1. Pengumpulan kebutuhan

Developer dan klien bertemu untuk menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan.

Selanjutnya peneliti akan melakukan analisis terhadap data apa saja yang dibutuhkan, seperti analisis terhadap sistem yang berjalan, analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis kebutuhan materi pembelajaran.

2. Perancangan

Perancangan dilakukan dengan cepat dan rancangan mewakili semua aspek software yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*. Dalam tahap ini peneliti akan membangun sebuah versi *prototype* yang dirancang kembali dimana masalah-masalah tersebut diselesaikan

3. Evaluasi *prototype*

Pada tahap ini, calon pengguna mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*. *Software* yang sudah jadi dijalankan dan akan dilakukan perbaikan apabila kurang memuaskan. Perbaikan termasuk dalam memperbaiki kesalahan atau kerusakan yang tidak ditemukan pada langkah sebelumnya.

Kelebihan *Prototype Model* adalah:

1. End user dapat berpartisipasi aktif.
2. Penentuan kebutuhan lebih mudah diwujudkan.
3. Mempersingkat waktu pengembangan *software*.

Kekurangan *Prototype Model* adalah:

1. Proses analisis dan perancangan terlalu singkat.
2. Mengesampingkan alternatif pemecahan masalah.
3. Biasanya kurang fleksibel dalam menghadapi perubahan.
4. *Prototype* yang dihasilkan tidak selamanya mudah dirubah.
5. *Prototype* terlalu cepat selesai.

2.12.2 Unified Modeling Language (UML)

Verdi Yasin (2012,p.194) mendefinisikan *Unified Modelling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak, UML menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan Penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

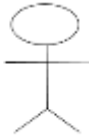


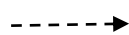
Menurut Verdi Yasin (2012,p.268) tipe-tipe diagram UML adalah sebagai berikut :

1) *Use Case Diagram*

Use case diagram adalah gambar dari beberapa atau seluruh aktor dan *use case* dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dan sistem.

Dalam *use case diagram* terdapat istilah seperti aktor, *use case* dan *case relationship*. Penjelasan simbol pada tabel 2.2




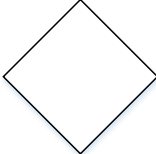

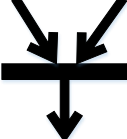
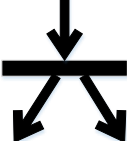
Tabel 2.2 Simbol Use Case

Simbol	Keterangan
	Aktor : Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.
	<i>Use case</i> : perangkat tertinggi dari fungsionalitas yang dimiliki sistem.
	<i>Association</i> : adalah relasi antara actor dan <i>use case</i> .
	<i>Generalisasi</i> : untuk memperlihatkan struktur pewaris yang terjadi.

2) *Activity Diagram*

Activity diagram menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi. *Activity Diagram* berupa flow chart yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity diagram* adalah sebagai berikut :

Tabel 2.2 Simbol *Activity Diagram*

Simbol	Keterangan
	<i>Activity</i> : Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Initial Node</i> : Bagaimana objek dibentuk atau diawali
	<i>Activity Final Node</i> : Bagaimana objek dibentuk dan diakhiri.
	<i>Decision</i> : Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
	<i>Swimlane</i> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.
	<i>Join</i> : Digunakan untuk menunjukkan kegiatan yang digabungkan.
	<i>Fork</i> : Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel

3) *Sequence Diagram*

Sequence diagram menggambarkan kolaborasi dinamis antara sejumlah dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang

terjadi pada titik tertentu dalam eksekusi sistem. *Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu.

Dalam *sequence* diagram terdapat 2 simbol yaitu :

- a. Actor, untuk menggambarkan pengguna sistem.
- b. *Lifeline*, untuk menggambarkan kelas dan objek.

4) *Class Diagram*

Class diagram menggambarkan dtruktur data dan desripsi *class*, *package*, dan objek beserta hubungan satu sama lain. *Class* diagram berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. *Class* memiliki 3 area pokok yaitu nama, atribut dan metode.

2.12.3 *Black Box Testing*

Menurut Rosa & Shalahuddin (2013,p.275) *black box testing* adalah pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian *black box* dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian *black box* harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah :

1. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) benar.
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tetapi kata sandi salah, atau sebaliknya, atau keduanya salah.