

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian terdahulu yang telah dilakukan antara lain:

1. (Umam, 2014), mengemukakan tentang pengembangan system informasi pengelolaan dan pelayanan kependudukan dan pencatatan sipil (studi di kota blitar jawa timur). Tujuan penelitian adalah untuk membantupetugas pelayanan maupun petugasoperator untukmerekap data pelayanan yangmasuk. Denganadanyasisteminformasipengelolaan data pelayanan kependudukan dan pencatatan sipil yang telah terintegrasidengan *database MySQL* akan mempermudah dalam pengelolaan data kependudukan yang dilakukan oleh pihak Dinas Kependudukan dan Pencatatan Sipil.
2. (Prihastini, 2014), mengemukakan tentang Sistem Informasi Pelayanan Kependudukan. Tujuan dari penelitian ini adalah merancang dan membangun suatu Sistem Informasi Pelayanan Kependudukan tingkat desa guna meningkatkan kualitas pelayanan bagi aparat desa dan penggunaan pelayanan bagi warga desa. Manfaat yang diperoleh dari penelitian ini adalah sistem ini dapat mempermudah aparat desa dalam pengumpulan data penduduk desa dan mempermudah melayani penduduk atau warga dalam permohonan surat - surat. Penelitian ini menggunakan model siklus hidup pengembangan sistem *waterfall* dan perancangan sistem informasi pelayanan kependudukan berbasis web. Metode pengumpulan data pada

penelitian ini menggunakan metode observasi, wawancara, dan angket, analisis datanya menggunakan teknik analisis deskriptif kualitatif.

3. (Eka Asyifa Hayat, 2014), mengemukakan tentang perancangan sistem informasi kependudukan berbasis. Tujuan dari penelitian adalah untuk merancang sistem informasi kependudukan menggunakan metode unified approach serta untuk pemodelan sistemnya menggunakan UML sedangkan bahasa pemrogramannya menggunakan PHP dan untuk *database* menggunakan xampp. Hasil penelitian ini menunjukkan bahwa penggunaan sistem informasi kependudukan ini dapat membantu bagian pendaftaran dan pendataan penduduk dalam mengelola data kelahiran, kematian dan pindah dating dan proses pembuatan laporan – laporan mengenai jumlah penduduk perbulan yang diharapkan.
4. (Triana, 2010), mengemukakan tentang analisa dan perancangan system informasi administrasi kependudukan berbasis desktop (studi kasus: kantor kelurahan parit lalang). Tujuan dari penelitian adalah penghematan waktu dalam mengolah data penduduk. Proses pengolahan data dapat dilakukan dalam waktu yang lebih cepat dan diusulkan dapat meningkatkan pelayanan, karena menghasilkan laporan yang lebih cepat dan lebih mudah digunakan serta meningkatkan kualitas informasi. Metode pengumpulan data pada penelitian ini menggunakan metode observasi, wawancara, kearsipan, kepustakaan. Metodologi penelitian yang akan digunakan adalah studi pustaka, wawancara, dan perancangan. Perancangan akan mencakup UML, ERD dan *UserInterface*. Perancangan UML terdiri dari empat

diagram yaitu *usecase* diagram, *activity* diagram, *sequence* diagram, dan *class* diagram.

2.2. Pengertian Perancangan Sistem

Menurut (Ladjamudin, 2005) perancangan sistem adalah:

“Kemampuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik.”

Menurut (Ladjamudin, 2005) perancangan sistem adalah :

“Pendefinisian kebutuhan-kebutuhan fungsional dengan mempersiapkan rancang bangun implementasi dengan menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi.”

Dari data diatas dapat disimpulkan bahwa perancangan sistem adalah suatu alternatif untuk memecahkan masalah dan yang telah dipilih selama tahap analisis dalam pemecahan masalah yang dihadapi perusahaan.

Tahapan perancangan memiliki tujuan untuk mendesain sistem baru untuk dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan sistem alternatif terbaik. Kegiatan yang dilakukan dalam perancangan ini meliputi perancangan *input*, *output* dan *file*.

1. Perancangan masukan (*input*)

Perancangan masukan bertujuan menentukan masukan data-data masuka yang akan digunakan untuk mengoperasikan sistem. Data-data masukan tersebut dapat berupa formulir-formulir, faktur dan lain-lain yang berfungsi

memberikan data masukan bagi pemroses sistem. Pada tahapan ini pula juga ditentukan format data masukan agar sesuai dengan kebutuhan sistem.

2. Perancangan keluaran (*output*)

Perancangan keluaran menentukan keluaran-keluaran yang akan digunakan oleh sistem. Keluaran tersebut berupa tampilan-tampilan layar dan juga format dan frekuensi laporan yang diperlukan.

3. Perancangan *file*

Perancangan file masukan dalam perancangan basis data yang diawali dengan merancang diagram hubungan antara entitas. Setelah itu melakukan uji normalisasi, setelah *file* yang telah lulus uji normalisasi yang harus dibutuhkan spesifikasi datanya.

Perancangan merupakan kegiatan untuk pengembangan sistem dan prosedur baru dalam kaitannya dengan sasaran-sasaran baru yang dikehendaki oleh pihak manajemen, untuk memperoleh suatu sistem informasi yang mampu dipakai untuk manajemen perusahaan secara lebih efektif dan efisien. Tahap ini memahami sistem ini sebelumnya dan kriteria-kriteria sistem yang akan dibangun, tim pembuat dapat membuat rancangan sistem informasi terlebih dahulu. Berdasarkan penjelasan di atas dapat disimpulkan bahwa perancangan kegiatan pokok yang logis untuk mendisain sistem baru yang dapat menyelesaikan masalah yang dihadapi perusahaan yang dapat diperoleh dari pemilihan alternatif sistem.

2.3. Konsep Dasar Sistem

2.3.1. Pengertian Sistem

Definisi sistem menurut beberapapandangan para ahli, yaitu:

Menurut (Jogiyanto, 2009) “Sistem dapat didefinisikan dengan pendekatan prosedur dan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu.”

Menurut (Yakub, 2012) "Sistem informasi adalah sebuah jaringan yang bekerja dari prosedur yang berhubungan, terkumpul bersama untuk melakukan suatu tujuan tertentu.”

Menurut (Jogiyanto, 2009)“Sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu.”

Jadi dapat disimpulkan bahwa sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

2.3.2. Karakteristik Sistem

Menurut (Jogiyanto, 2009) suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerja sama membentuk satu kesatuan. Suatu sistem mempunyai karakteristik atau sifat-sifat untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

a. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi atau subsistem-subsistem.

b. Batasan Sistem

Daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

c. Lingkungan Luar Sistem

Suatu sistem yang berasal di luar batas dari sistem yang mempengaruhi operasi sistem.

d. Penghubung Sistem

Sebagai media penghubung antara satu subsistem dengan subsistem yang lainnya.

e. Masukan Sistem

Merupakan suatu energi yang masuk dari lingkungan luar ke dalam sistem sehingga menyebabkan sistem bekerja.

f. Keluaran Sistem

Merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

g. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

h. Tujuan (Goal) Sistem

Merupakan suatu sistem pasti mempunyai tujuan atau sasaran. Jika suatu saat sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

2.3.3. Klasifikasi Sistem

Sistem dapat di klasifikasikan dari beberapa sudut pandang diantaranya adalah sebagai berikut menurut (Jogiyanto, 2009):

a. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan sistem fisik merupakan sistem yang ada secara fisik.

b. Sistem Alamiah dan Sistem Buatan Manusia.

Sistem alamiah adalah sistem yang terjadi karena proses alam tidak dibuat oleh manusia (ditentukan dan tunduk kepada kehendak sang pencipta alam), sedangkan sistem buatan manusia adalah sistem yang dirancang oleh manusia.

c. Sistem Tertentu (*deterministic system*) dan Sistem Tak Tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem tertentu relatif stabil atau konstan dalam jangka waktu yang lama, sedangkan sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

d. Sistem Tertutup dan Sistem Terbuka.

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya, sedangkan sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lain.

2.4. Konsep Dasar Informasi

2.4.1. Pengertian Informasi

Di dalam pengolahan sistem pada akhirnya menghasilkan suatu informasi, untuk itu pendefinisian informasi diperlukan untuk menunjang berhasilnya pengembangan sistem yang akan dirancang. Definisi umum untuk informasi dalam sistem informasi sebagai berikut:

Menurut (Jogiyanto, 2009) "Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerimanya."

Menurut (Ladjamudin, 2005) "Informasi adalah sebagai data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun yang akan datang."

Jadi, dapat disimpulkan informasi adalah data yang diolah menjadi bentuk yang lebih berarti dan berguna bagi penerima sebagai pengambilan keputusan dimasa kini dan dimasa yang akan datang.

2.4.2. Kualitas Informasi

Kualitas dari informasi (*quality of information*) tergantung dari 3 hal, yaitu:

a. Akurat (*accurate*)

Informasi harus bebas dari kesalahan dan tidak bias atau menyesatkan.

b. Tepat pada waktunya (*timelines*)

Informasi yang sampai pada penerima tidak boleh terlambat.

c. Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakainya.

2.5. Pengertian Sistem Informasi

Sistem informasi diharapkan dapat membantu organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi yang dibutuhkan sehingga dapat digunakan dalam pengambilan keputusan atau untuk mengendalikan kegiatan organisasi.

Menurut Leitch A. Robert dan Davis Roscoe K : “Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan”. (Jogiyanto, 2009)

Menurut (Ladjamudin, 2005) “Sistem informasi adalah sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambilan keputusan dan atau untuk mengendalikan organisasi”.

Jadi sistem informasi merupakan suatu sistem yang mempertemukan antara kebutuhan dan transaksi didalam organisasi yang dapat memberikan suatu informasi yang berguna bagi organisasi tersebut. Sistem informasi dapat menjadi pengendali dan sarana untuk mengambil suatu keputusan.

Suatu sistem informasi terdapat lima komponen yang dapat diklasifikasikan sebagai berikut :

- a. Perangkat keras (*hardware*) dan perangkat lunak (*software*) yang berfungsi sebagai mesin.
- b. Orang (*people*) dan prosedur (*procedures*) yang merupakan manusia dan tatacara menggunakan mesin.
- c. Data merupakan jembatan penghubung antara manusia dan mesin agar terjadi suatu proses pengolahan data.

2.6. Pengertian Pelayanan

Konsep Pelayanan Istilah pelayanan berasal dari kata “layan” yang artinya menolong menyediakan segala apa yang diperlukan oleh orang lain untuk perbuatan melayani. Pada dasarnya setiap manusia membutuhkan pelayanan, bahkan secara ekstrim dapat dikatakan bahwa pelayanan tidak dapat dipisahkan dengan kehidupan manusia (Sinambela, 2010). Pelayanan adalah proses pemenuhan kebutuhan melalui aktivitas orang lain yang langsung (Moenir, 2006).

2.7. Pengertian Internet

Internet (*interconnected-networking*) adalah system global dari seluruh jaringan computer yang saling terhubung menggunakan standar *Internet Protocol Suite* (TCP/IP) untuk melayani miliaran pengguna diseluruh dunia, yang terhubung secara global dan menggunakan TCP/IP sebagai protocol pertukaran paket (*packet switching communication protocol*). Rangkaian internet yang terbesar dinamakan internet. Cara menghubungkan rangkaian dengan kaedah ini dinamakan *internet working*.

2.7.1. Web Server

Web server adalah *software* yang menjadi tulang belakang dari *world wide web* (www). Webserver menunggu permintaan dari *client* yang menggunakan *browser* seperti Internet Explorer, Mozilla, dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*. Data ini mempunyai format yang standar, disebut

dengan format SGML (*standargeneral markup language*). Data yang berupa format ini kemudian akan ditampilkan oleh *browser* sesuai dengan kemampuan *browser* tersebut. Contohnya, bila data yang dikirim berupa gambar, *browser* yang hanya mampu menampilkan teks (misalnya lynx) tidak akan mampu menampilkan gambar tersebut, dan jika ada akan menampilkan alternatifnya. *Web server*, untuk berkomunikasi dengan *client*-nya(*web browser*) mempunyai protokol sendiri, yaitu HTTP (*hypertext transfer protocol*) (Effendy, 2009).

2.7.2. Web Browser

Web browser adalah *software* yang digunakan pada komputer pemakai (*client*) yang meminta informasi dari *webserver* dan penerjemah dari kode-kode instruksi yang ada padahal aman *web* untuk ditampilkan pada monitor computer sesuai dengan yang diinginkan oleh *user*.

2.7.3. WEB dan HTML

Menurut (Sampurna, 1996) *web* adalah sekelompok kode berbasis teks yang sederhana dan universal, disebut *Hypertext Markup Language* (HTML). Karena berbasis teks, HTML dikenali dan diterjemahkan segala jenis computer dalam bentuk tampilan informasi yang sama. HTML adalah kreasi Tim BernersLee, ilmuwan *European Laboratory for Particle Physics* (CERN organisasi penelitian 18 negara Eropa) di Geneva Swiss. Maret 1989 Tim memperkenalkan WEB dan HTML sebagai standar antar muka distribusi informasi diinternet yang mampu menggabungkan teks, grafik dan multimedia dengan metode navigasi menu pada mesin UNIX.

HTML adalah bahasa kode pemrograman yang menjadi dasar bagi terwujudnya *web*. Dengan HTML seluruh sistem computer yang saling berbeda dapat mengenali format yang ditampilkan dalam situs internet tanpa perbedaan yang berarti, termasuk didalamnya penampilan multimedia (grafik, suara dan citravideo). HTML berbasis teks yang sangat sederhana dan praktis sehingga dapat dipahami oleh berbagai jenis computer dalam *platform* sistem yang berbeda.

Dokumen HTML mengandung perintah teks yang disebut *tag* untuk menampilkan tulisan, gambar, warna, suara, video, dan animasi. serta *link* yang menghubungkan berbagai topik. Dengan bentuk *hypertext* maka halaman internet (*website*– kios internet) akan dapat dibangun dengan cepat serta ditampilkan dengan cantik. Dalam konsep *hypertext* ini pembacaan suatu dokumen tidak harus berurutan namun bias melompat antar topic bahkan di *print* maupun *dicopy* ke media penyimpan local (*harddisk* misalnya). Format *hypertext* juga memungkinkan pemakai mesin yang hanya bisa mengenali teks untuk mengakses dokumen dalam bentuk teks. Perbedaannya hanya terletak pada tidak ditampilkannya gambar, grafik, animasi, video, suara dan warna.

2.8. Framework PHP

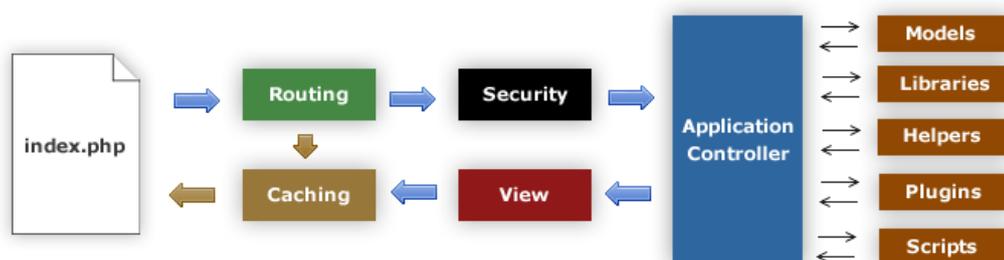
Framework secara sederhana dapat diartikan kumpulan dari fungsi- fungsi/ prosedur-prosedur dan kelas-kelas untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau kelas dari awal. (Septian, 2011)

2.8.1. Code Igniter

Code Igniter merupakan aplikasi *opensource* berupa *framework* PHP dengan model MVC (*Model, View, Controller*) untuk membangun aplikasi web dinamis dengan cepat dan mudah. *Code Igniter* memiliki desain dan struktur file yang sederhana, didukung dengan dokumentasi yang lengkap sehingga *framework* ini lebih mudah dipelajari.

Code Igniter ini memungkinkan para pengembang untuk menggunakan *framework* secara parsial atau secara keseluruhan. Artinya bahwa *Code Igniter* masih member kebebasan kepada para pengembang untuk menulis bagian-bagian kode tertentu didalam aplikasi menggunakan cara konvensional atau dengan *syntax* umum didalam PHP, tidak harus menggunakan aturan penulisan kode di *Code Igniter*. (Septian, 2011)

Adapun alur dari aplikasi yang ditulis menggunakan *Code Igniter* seperti pada gambar 2.1. :



Gambar 2.1. Alur Gambar *Framework Code Igniter*

File `index.php`, berfungsi sebagai *frontcontroller*, menginisialisasi *resource* utama yang dibutuhkan untuk menjalankan *Code Igniter*.

- a. *Router*, memeriksa HTTP *request* untuk menentukan apa yang harus dilakukan.
- b. Jika file cache ada, dikirim langsung ke *browser*, melewati eksekusi sistem normal.
- c. Keamanan, sebelum *controller* aplikasi dimuat, HTTP *request* dan setiap data pengguna yang *submit* disaring terlebih dahulu untuk keamanan.
- d. *Controller*, memuat model, *library* utama, *helper*, dan setiap *resource* lainnya yang diperlukan untuk memproses permintaan khusus.
- e. *View*, proses *render* kemudian dikirim ke web *browser* agar dapat dilihat. Jika *caching* diaktifkan, *view* *cache* terlebih dahulu sehingga pada permintaan berikutnya dapat dilayani.

2.9. Metode Pengembangan Sistem

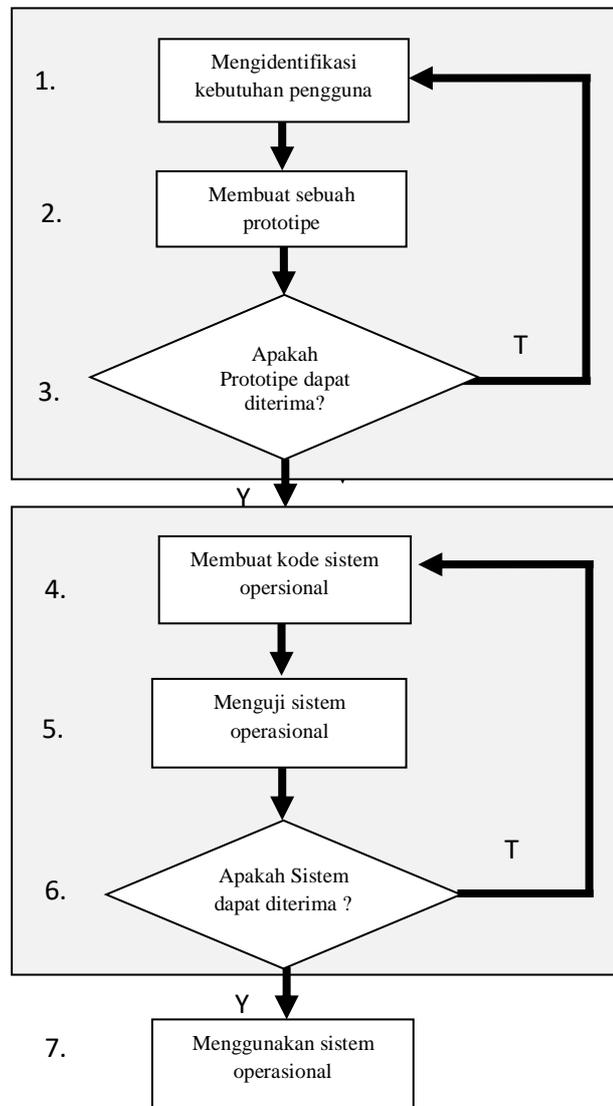
Terdapat beberapa pendekatan sistem dalam pengembangan sistem, salah satu diantaranya adalah Prototipe.

Menurut (Raymond, 2007):

”*Prototype* adalah satu versi dari sistem potensial yang memberikan ide bagi para pengembang dan calon pengguna, bagaimana sistem akan berfungsi dalam bentuk yang telah selesai. Proses pembuatan *prototype* ini disebut *Prototyping*.”

Dua jenis pengembangan sistem prototipe yaitu evolusioner dan persyaratan. Prototipe evolusioner (*Evolutionary Prototype*) terus menerus disempurnakan sampai memiliki seluruh fungsionalitas yang dibutuhkan pengguna dari sistem yang baru. Prototipe ini kemudian diproduksi. Jadi, satu prototipe evolusioner akan menjadi sistem aktual. Akan tetapi, prototipe persyaratan (*requirements prototype*) dikembangkan sebagai satu cara untuk mendefinisikan persyaratan-persyaratan fungsional dari sistem baru ketika pengguna tidak mampu mengungkapkan dengan jelas apa yang mereka inginkan. Sesuai dengan fungsi di atas maka penulis menggunakan jenis prototipe persyaratan sesuai dengan fungsinya.

Tujuh tahap pengembangan sistem menggunakan prototipe menurut (Raymond, 2007) dalam dilihat pada gambar 2.2. sebagai berikut:



Gambar 2.2. Tahap Pembangunan Prototipe

Ada tujuh tahapan dalam pengembangan sistem model prototipe *requirement*, yaitu:

1. Mengidentifikasi Kebutuhan Pemakai

Pengembang mengidentifikasi terhadap pemakai untuk memperoleh suatu gagasan mengenai apa yang dibutuhkan dari sistem yang akan digunakanyaitu dengan mewawancarai bagian administrasi, untuk mendapatkan ide mengenai apa yang diminta dari sistem.

2. Membuat Satu Prototipe

Pengembang menggunakan satu alat *prototyping* atau lebih untuk membuat prototipe. Contoh dari alat-alat *prototyping* adalah generator aplikasi terintegrasi dan *toolkit prototyping*. Generator aplikasi terintegrasi (*integrated application generator*) adalah sistem piranti lunak siap pakai yang mampu membuat seluruh fitur yang diinginkan dari sistem baru seperti menu, laporan, tampilan, basis data dan seterusnya.

3. Menentukan Apakah Prototipe Dapat Diterima

Pengembang mendemonstrasikan prototipe kepada pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan. Jika ya, langkah 4 akan diambil, jika tidak prototipe akan direvisi dengan mengulang kembali langkah 1, 2 dan 3 dengan pemahaman lebih baik tentang kabutuhan pengguna.

4. Membuat Kode Sistem Baru

Pengembang menggunakan prototipe sebagai dasar untuk pengkodean sistem yang baru yang bertujuan mengkonversi rancangan program ke program yang terdokumentasi dengan baik.

5. Menguji Sistem Baru

Pengembang menguji sistem baru menggunakan *black box testing*.

a. Black Box Testing

Pengujian *black box* berfokus pada pengujian persyaratan fungsional perangkat lunak, untuk mendapatkan serangkaian kondisi *input* yang sesuai dengan persyaratan fungsional suatu program.

Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada, tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut.

6. Menentukan apakah sistem yang baru dapat diterima

Pengguna memberitahukan kepada pengembang apakah sistem dapat diterima. Jika ya, maka langkah 7 diambil, jika tidak, langkah 4 dan 5 diulang kembali.

7. Menggunakan sistem yang baru

Pada tahap ini, sistem akan dinyatakan lulus dapat mulai digunakan untuk menangani prosedur yang sesungguhnya. Selama sistem digunakan, tim teknis harus memperhatikan masalah pemeliharaan sistem.

2.10. Pengembangan Berorientasi Objek

Pengembangan berorientasi objek merupakan cara berfikir mengenai perangkat lunak yang berdasarkan pada abstraksi yang terdapat pada dunia nyata.

2.10.1. Metodologi Berorientasi Objek

Metodologi berorientasi objek terbagi menjadi 3 yaitu:

a. Object Oriented Analysis (OOA)

Object Oriented Analysis (OOA) dimulai dengan menyatakan suatu masalah dengan cara membuat analisis situasi dari dunia nyata.

b. Object Oriented Design (OOD)

Object Oriented Design (OOD) merupakan tahap lanjutan setelah analisis berorientasi objek. Di dalam OOD menentukan karakteristik penampilan secara optimal, dan menentukan strategi memecahkan masalah.

2.10.2. Karakteristik Berorientasi Objek

Menurut (Ariesto A. S., 2002), metodologi pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama, yaitu:

a. Encapsulation

Encapsulation (pengkapsulan) merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses.

b. Inheritance

Inheritance (pewarisan) merupakan properti yang muncul ketika tipe entitas atau *object class* disusun secara hierarki dan setiap tipe entitas

atau *objectclass* menerima atau mewarisi atribut dan metode dari pendahulunya.

c. Polymorphism

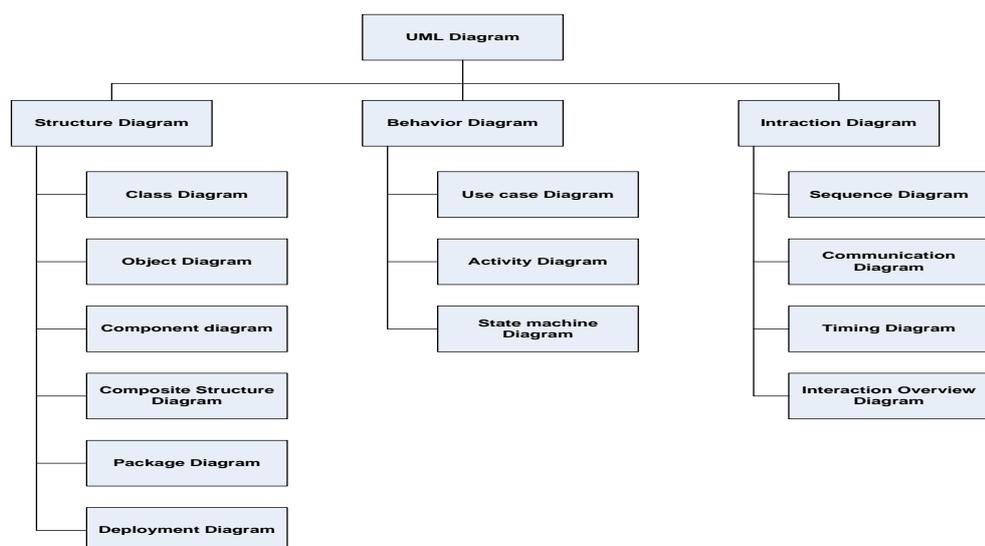
Polymorphism yaitu konsep yang menyatakan bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda.

2.11. Perancangan Sistem *Unified Modelling Language* (UML)

Menurut (Shalahudin, 2014) :

“*Unified Modelling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”

Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam tiga kategori (Shalahudin, 2014). Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



Gambar 2.3.Diagram *Unified Modelling Language* (UML)

Berikut ini merupakan penjelasan singkat dari pembagian kategori tersebut:

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsystem pada suatu sistem.

Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *Unified Modelling Language (UML)*:

1. *Class Diagram*

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut maupun metode atau operasi. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Berikut adalah simbol-simbol yang ada pada diagram kelas (Shalahudin, 2014):

Tabel 2.1. Simbol-simbol *Class Diagram*

| Simbol | Deskripsi | | | |
|---|-------------------|----------|------------|----------------------------|
| Kelas <table border="1" style="margin-left: 20px;"> <tr> <td>nama_kelas</td> </tr> <tr> <td>+atribut</td> </tr> <tr> <td>+operasi()</td> </tr> </table> | nama_kelas | +atribut | +operasi() | Kelas pada struktur sistem |
| nama_kelas | | | | |
| +atribut | | | | |
| +operasi() | | | | |

| | |
|---|---|
| Antarmuka/ <i>Interface</i>  nama_interface | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek |
| Asosiasi/ <i>asociation</i>  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> |
| Asosiasi berarah/ <i>directed association</i>  | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> . |
| Generalisasi  | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) |
| Kebergantungan/ <i>dependecy</i>  | Relasi antar kelas dengan makna kebergantungan antar kelas |
| Agregasi/ <i>agregation</i>  | Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>) |

2. *Object Diagram*

Object Diagram menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak maka pendefinisian kelas itu tidak dapat dipertanggung jawabkan.

3. *Component Diagram*

Component Diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem.

4. *Composite Structure Diagram*

Composite Structure Diagram baru mulai ada pada UML versi 2.0, pada versi 1.x diagram ini belum muncul. Diagram ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan dari *instance* yang saling terhubung.

5. *Package diagram*

Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. Hampir semua diagram dalam UML dapat dikelompokkan menggunakan *Package diagram*.

6. *Deployment Diagram*

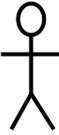
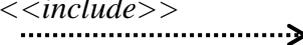
Deployment Diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.

7. *Use CaseDiagram*

Use CaseDiagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada diagram *use case* (Shalahudin, 2014):

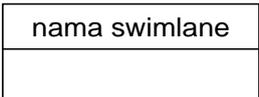
Tabel 2.2. Simbol-Simbol Use Case Diagram

| Simbol | Deskripsi |
|---|---|
| <p><i>Use Case</i></p>  | <p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p> |
| <p>Aktor/<i>actor</i></p>  | <p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang</p> |
| <p>Asosiasi/<i>asociation</i></p>  | <p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p> |
| <p>Ekstensi/<i>extend</i></p>  | <p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan</p> |
| <p>Generalisasi/<i>generalization</i></p>  | <p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya</p> |
| <p>Menggunakan/<i>Include/uses</i></p>  | <p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> |

8. Activity Diagram

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas(Shalahudin, 2014):

Tabel 2.3. Simbol-simbol Activity Diagram

| Simbol | Deskripsi |
|---|--|
| Status awal  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal |
| Aktivitas  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja |
| Percabangan/ <i>decision</i>  | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
| Penggabungan/ <i>join</i>  | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |
| Status akhir  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| <i>Swimlane</i>  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

9. *State Machine Diagram*

State Machine Diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem. Perubahan tersebut digambarkan dalam suatu graf berarah. Diagram ini cocok digunakan untuk menggambarkan alur interaksi pengguna dengan sistem.

10. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Berikut adalah simbol-simbol yang ada pada diagram sekuen:

11. *Communication Diagram*

Communication Diagram pada UML versi 2.x menyederhanakan dari diagram kolaborasi pada UML 1.x. jadi diagram kolaborasi tidak muncul lagi pada UML versi 2.x. diagram komunikasi menggambarkan interaksi antar objek dalam bentuk urutan pengiriman pesan. Diagram komunikasi merepresentasikan informasi yang diperoleh dari diagram kelas, diagram sekuen, dan diagram use case untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem.

12. *Timing Diagram*

Timing Diagram merupakan diagram yang fokus pada penggambaran terkait batasan waktu. Diagram ini digunakan untuk menggambarkan tingkah laku sistem dalam periode waktu tertentu. *Timing diagram* biasanya digunakan

untuk mendeskripsikan operasi dari alat digital karena penggambaran secara visual akan lebih mudah dipahami dari pada kata-kata.

13. *Interaction Overview Diagram*

Interaction Overview Diagram mirip dengan diagram aktivitas yang berfungsi untuk menggambarkan sekumpulan urutan aktivitas. *Interaction Overview Diagram* adalah bentuk aktivitas diagram yang setiap titik merepresentasikan diagram interaksi.

2.12. Pengujian Sistem

Pengujian menyajikan anomaly yang menarik bagi perancang perangkat lunak. Pada proses perangkat lunak, perancang pertama-tama berusaha membangun perangkat lunak dari konsep abstrak keimplementasi yang dapat dilihat, baru kemudian dilakukan pengujian.

2.12.1. Sasaran-Sasaran Pengujian

Terdapat sejumlah aturan yang berfungsi sebagai sasaran pengujian:

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
2. Test case yang baik adalah test case yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

2.12.2. Prinsip Pengujian

Sebelum mengaplikasikan metode untuk mendesain test case yang efektif, perencana perangkat lunak harus memahami prinsip dasar yang menuntun pengujian perangkat lunak. Serangkaian prinsip pengujian akan dijelaskan berikut ini:

1. Semua pengujian harus dapat ditelusuri sampai ke persyaratan pelanggan.
2. Pengujian harus direncanakan lama sebelum pengujian itu dimulai.
3. Prinsip pareto berlaku untuk pengujian perangkat lunak.
4. Pengujian harus mulai dari yang kecil dan berkembang kepengujian yang besar.
5. Pengujian yang mendalam tidak mungkin.
6. Untuk menjadi paling efektif, pengujian harus dilakukan oleh pihak ketiga yang independent.

2.12.3. Pengujian Black – Box

Pengujian black – box berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian black - box memungkinkan perencana perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Pengujian black – box berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi – fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses eksternal

4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi