**BAB II**

**LANDASAN TEORI**

* 1. **Konsep Dasar**
     1. **Pengertian Data**

Informasi merupakan salah satu sumber daya penting dalam manajemen modern. Banyak keputusan strategis yang bergantung kepada informasi. Sebagaimana diketahui, sumber daya 4M+1I yang mencakup manusia (sumber daya manusia atau SDM), material (termasuk di dalamnya energi), mesin, modal, dan informasi merupakan sumber daya vital bagi kelangsungan organisasi bisnis.

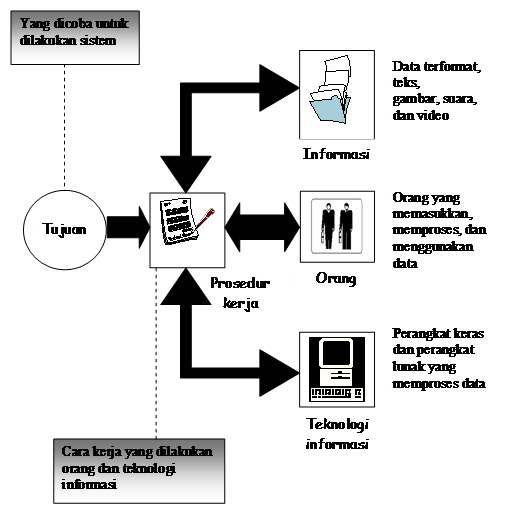
McFadden, dkk (1999) mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Shannon dan weaver, dua orang insinyur listrik, melakukan pendekatan secara matematis untuk mendefinisikan informasi (Kroenke, 1992). Menurut mereka, informasi adalah “jumlah ketidakpastian yang dikurangi ketika sebuah pesan diterima”. Artinya, dengan adanya informasi, tingkat kepastian menjadi meningkat. Menurut Davis (1999), informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.



**Gambar 2.1** Transformasi data menjadi informasi (Kadir, 2003)

* + 1. **Pengertian Informasi**

Sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan. Istilah sistem informasi juga sering dikacaukan dengan sistem informasi manajemen (SIM). Kedua hal ini sebenarnya tidak sama. Sistem informasi manajemen merupakan salah satu jenis sistem informasi, yang secara khusus ditujukan untuk menghasilkan informasi bagi pihak manajemen dan untuk pengambil keputusan.



**Gambar 2.2** Pengertian Sistem Informasi (Kadir, 2003).

* + 1. **Pengertian Pengetahuan (*Knowladge*)**

Menurut Al-Bahra Bin Ladjamudin, 2005 dalam bukunya yang berjudul Analisis & Desain Sistem Informasi menyebutkan bahwa : ”Perancangan adalah suatu kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik.”

Menurut My Earth, 2015 dalam bukunya yang berjudul Perancangan Sistem dan Analisis, menyebutkan bahwa ”Perancangan adalah suatu kegiatan membuat desain teknis berdasarkan evaluasi yang telah dilakukan pada kegiatan analisis”.

Berdasarkan definisi di atas, penulis menarik kesimpulan bahwa perancangan merupakan suatu pola atau kerangka yang dibuat untuk mengatasi masalah yang dihadapi perusahaan atau organisasi setelah melakukan analisis terlebih dahulu.

* + 1. **Aplikasi *Web***

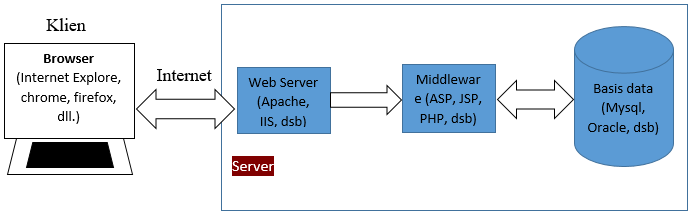
Pada awalnya aplikasi *web* dibangun hanya dengan menggunakan bahasa yang disebut HTML (*Hypertext Markup Language*). Pada perkembanganya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML. Pada saat ini, banyak skrip seperti itu, antara lain PHP dan ASP, sedangkan contoh yang berupa objek adalah applet.

Aplikasi *web* sendiri dapat dibagi menjadi:

1. *Web* Statis
2. *Web* Dinamis

*Web* statis dibentuk dengan menggunakan HTML saja. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perubahan yang terjadi. Kelemahan ini diatasi dengan model apikasi *web* dinamis.

Dengan memperluas kemampuan HTML, yakni dengan menggunakan perangkat lunak tambahan, perubahan informasi dalam halaman-halaman *web* dapat ditangani melalui perubahan data, bukan melalui perubahan program, aplikasi *Web* dapat dikoneksikan ke basis data. Dengan demikian perubahan informasi dapat dilakukan operator atau yang bertanggung jawab terhadap kemutakhiran data, dan tidak menjadi tanggung jawab pemrogram atau *Webmaster*.



**Gambar 2.3** Arsitektur aplikasi *web* (Kadir, 2003).

* 1. **Teori Program**
     1. **Basis Data**

Basis data atau *database* terdiri atas 2 kata, yaitu Basis dan Data. Basis diartikan sebagai markas atau Gudang, sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan peristiwa, konsep dan sebagainya yang akan menjadi sebuah entitas (Fathansyah, 2015).

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data atau arsip dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan data atau arsip. Perbedaannya hanya pada media penyimpanan yang digunakan. Jika arsip menggunakan sebuah lemari sebagai media penyimpanan, maka basis data menggunakan media elektronik seperti cakram magnetis.

Sedangkan menurut fathansyah di bukunya yang lain menyebutkan bahwa, Basis Data (*database*) dapat dibayangkan sebagai sebuah lemari arsip, jika kita memiliki sebuah lemari arsip dan berwenang/bertugas untuk mengelolanya maka kemungkinan besar kita akan melakukan hal – hal seperti memberi sampul, memberi map, pada kumpulan arsip yang akan disimpan untuk menentukan kelompok/jenis arsip dan memberi penmoran menggunakan pola tertentu bagaimana arsip – arsip tadi dapat tersusun. (Fathansyah, 2007).

Perbedaan media ini selanjutnya melahirkan yang selanjutnya melahirkan perbedaan-perbedaan lain yang menyangkut jumlah dan jenis metode yang dapat digunakan dalam upaya penyimpanan. (Fathansyah, 2015).

Berdasarkan ketiga sumber data di atas maka penulis dapat menyimpulkan bahwa basis data adalah sebuah tempat atau lemari sebagai media penyimpanan, untuk mengolah sebuah data yang tersusun yang tujuan utamanya adalah agar dapat mencari sebuah data dengan mudah dan cepat.

A close up of text on a white background

Description generated with high confidence

**Gambar 2.8** Lemari arsip dan basis data (Fathansyah, 2015).

* + 1. ***Database Management System* (MySQL)**

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL atau DBMS yang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi diseluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunanya tidak cocok dengan penggunaan GPL.

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, SQL (*Structure Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemelihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis kerangka pemikiran.

* + 1. **PHP (*PHP: Hypertext Preprocessor*)**

PHP *Hypertext Preprocessor* adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*.Karena sifatnya yang *Server Side* *Scripting*, maka untuk menjalankan PHP harus menggunakan *web server*.  *Server Side* *Scripting* pada PHP dapat bekerja jika ada tiga komponen berikut: PHP *Parse* (CGI atau *Server* Modul ), *web server* (contohnya *Apache* dalam XAMPP) dan *web browser*. Hasil *output* PHP yang melewati *web server* dapat dilihat pada *web browser*.

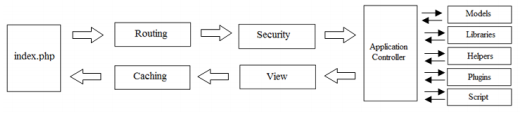
PHP juga dapat diintegrasikan dengan HTML, *JavaScript*, *JQuery* dan *Ajax*. Namun, pada umumnya PHP lebih banyak digunakan bersamaan dengan *file* bertipe HTML. Dengan menggunakan PHP kita bisa membuat *website powerful* yang dinamis dengan disertai manajemen *database*-nya.

* + 1. ***Framework Bootstrap***

*Bootstrap* adalah *open source front-end framework* untuk merancang tampilan *web*. *Bootstrap* terdiri dari kumpulan kode *Hypertext Markup Language* (HTML), *Cascading Style Sheet* (CSS) dan *JavaScript* “siap pakai” yang memudahkan kita dalam membuat desain *web*. B*ootstrap* juga sering disebut sebagai CSS *framework* karena berisi sebuah “kerangka kerja” (*framework*) bagaimana cara menyusun kode HTML dan CSS sesuai dengan aturan dari *Bootstrap*. Namun karena *bootstrap* juga menggunakan *JavaScript*, maka lebih pas disebut sebagai *front-end framework*. Dengan *bootstrap*, kita bisa membuat tampilan *web* menarik tanpa harus kode-kode rumit dibaliknya. Dasar yang diperlukan cukup basic HTML, CSS dan sedikit *JavaScript* jika ingin membuat efek-efek interaktif.

* + 1. ***Framework CodeIgniter* (CI)**

*CodeIgniter* (CI) adalah sebuah *framework* PHP untuk membuat sebuah aplikasi berbasis *web*, karena didalamnya terdapat fitur lengkap aplikasi *web* dimana fitur-fitur tersebut sudah dikemas menjadi satu. CI mempunyai file dokumentasi yang sangat memadai untuk menjelaskan setiap fungsi yang ada pada *library* dan *helper*. File dokumentasi ini disertakan secara langsung pada saat mengunduh paket *framework* CI. Untuk lebih memahami cara kerja *framework* CI, berikut susunan dan komponen CI pada Gambar 2.4.



**Gambar 2.4** Cara Kerja *Framework* CI (Riyanto, 2011).

Sumber: Buku Membuat Sendiri Aplikasi *E*-*Commerce* dengan PHP & MYSQL Menggunakan *CodeIgniter* & *JQuery*.2011

Berdasarkan Gambar 2.4 merupakan diagram cara kerja *framework codeIgniter* adalah sebagai berikut:

1. File index.php merupaka file yang bertindak sebagai kontrol halaman depan. File ini menginisialisai semua *resource* yang diperlukan untuk menjalankan *codeIgniter* atau halaman situs
2. Routing memeriksa setiap *request* HTTP (*hypertext transfer protocol*) dan melakukan apa yang perlu dilakukan terhadap *request* tersebut, tergantung *rule* yang ada.
   1. **Rekayasa Perangkat Lunak SDLC *(Systems Development Life Cycle)***

Siklus Hidup Pengembangan Sistem atau *System Development Life Cycle* (Siklus Hidup Sistem), dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut.

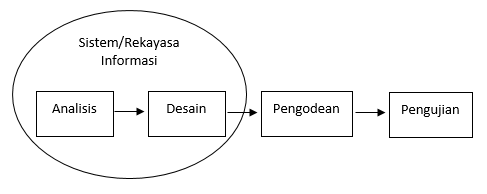


**Gambar 2.5** Siklus *System Development Life Cycle*(Sri Dharwiyanti, 2013)

* + 1. **Model pengembangan perangkat lunak air terjun(*Waterfall*)**

Model [*Waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*.Berikut ini ada dua gambaran dari waterfall model (Pressman, 2013).](https://www.blogger.com/null)

Metode SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis desain, pengodean, pengujian, dan tahap pendukung (*support*).



**Gambar 2.6** Ilustrasi Model *Waterfall* (M. Shalahudin, 2016).

Berikut penjelasan tahapan pengembangan sistem dengan metode *waterfall*:

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

1. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasikan kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

1. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

1. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai yang diinginkan.

1. Pemeliharaan (*Maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena ada kesalahan muncul dan tidak terdeteksi dengan lingkungan baru. Tahap pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

* 1. **Pemodelan dan UML (*Unified Modeling Language*)**

Salah satu pemodelan yang saat ini paling banyak digunakan adalah UML. *Unified Modeling 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.

* + 1. **Pemodelan**

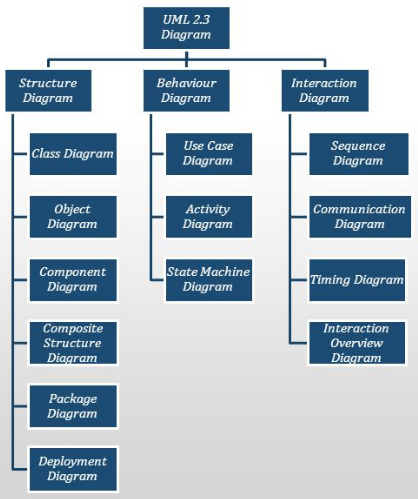
Pemodelan adalah gambaran dari realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pemodelan dapat menggunakan bentuk yang sama dengan realitas misalnya jika seorang arsitek ingin memodelkan sebuah gedung yang akan dibangun maka dia akan memodelkannya dengan membuat sebuah maket (tiruan) arsitektur gedung yang akan dibangun di mana maket itu akan dibuat semirip mungkin dengan desain gedung yang akan dibangun agar arsitektur gedung yang diinginkan dapat terlihat. Pemodelan juga banyak digunakan untuk merencanakan suatu hal agar kegagalan dan resiko yang mungkin terjadi dapat diminimalisasi.

Pada dunia pemrograman pembangunan perangkat lunak sistem informasi juga diperlukan pemodelan. Pemodelan perangkat lunak digunakan untuk mempermudah langkah berikutnya dari pengembangan sistem informasi sehingga lebih terencana.

* + 1. **Pengenalan UML**

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunakyang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumantasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

* + 1. ***Diagram* UML**



**Gambar 2.7** Diagram UML(M. Shalahudin, 2016).

Pada UML 2.3 terdiri dari 3 macam diagram yang dikelompokan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar di atas.

Berikut penjelasan singkat dari pembagian ketegori pada gambar 2.7 diagram UML.

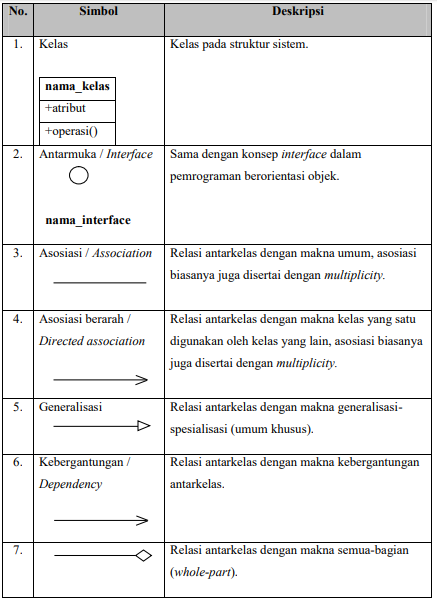
1. *Structure Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction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

Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi.

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem, sehingga pembuat perangkat lunak atau *programmer* dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas.

**Tabel 2.1** Simbol-simbol *Class Diagram* (M. Shalahudin, 2016)



* + 1. ***Use Case Diagram***

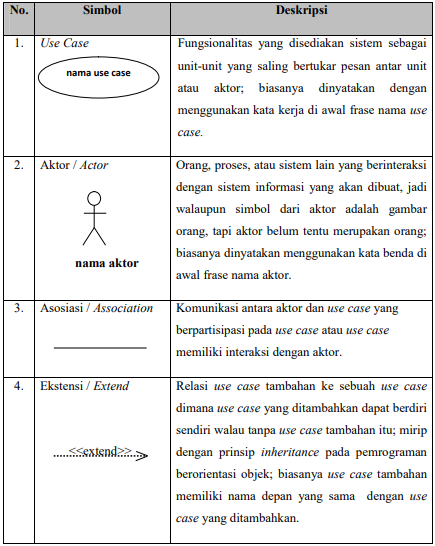
*Use Case Diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam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

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

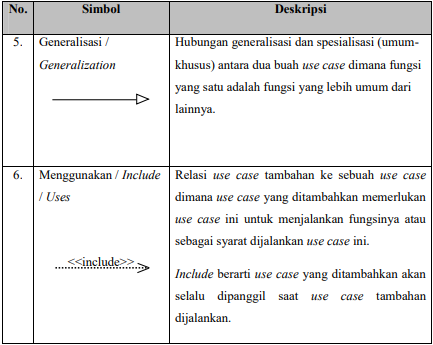
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

**Tabel 2.2** Simbol-simbol *Use Case Diagram* (M. Shalahudin, 2016)



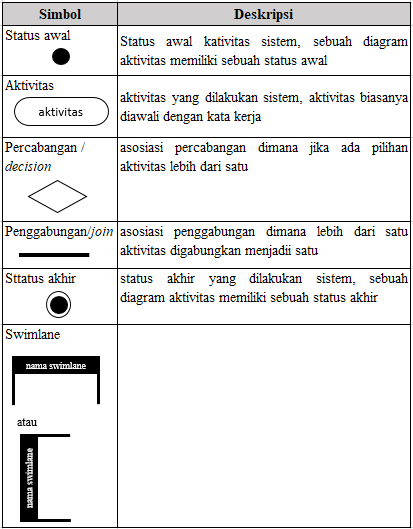
Lanjutan **Tabel 2.2** Simbol-simbol *Use Case Diagram* (M. Shalahudin, 2016)



* + 1. ***Activity Diagram***

Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

**Tabel 2.3** Simbol-simbol *Activity Diagram* (M. Shalahudin, 2016)

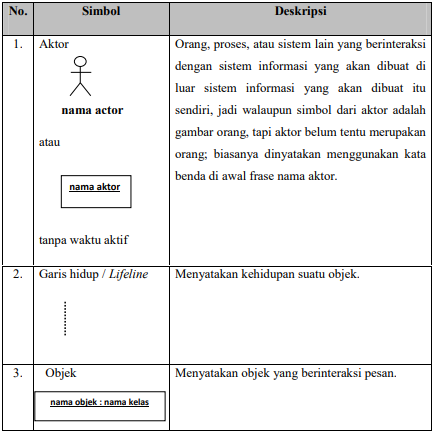


* + 1. ***Sequence Diagram***

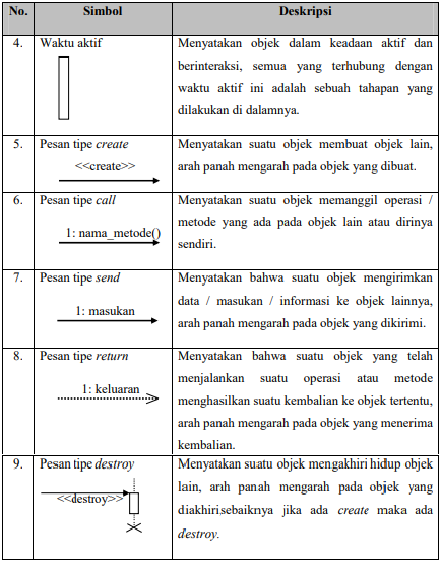
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Berikut simbol-simbol yang ada pada diagram sekuen:

**Tabel 2.4** Simbol-simbol *Sequence Diagram* (M. Shalahudin, 2016)

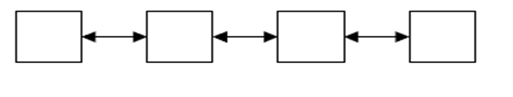


Lanjutan **Tabel 2.4** Simbol-simbol *Sequence Diagram* (M. Shalahudin, 2016)



* 1. **Navigasi Diagram**

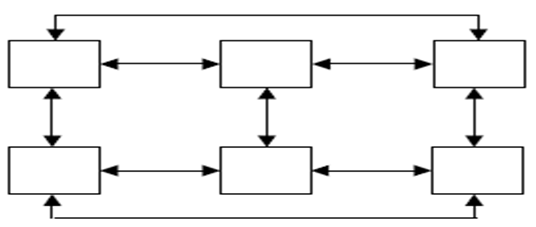
Struktur navigasi linier hanya mempunyai satu rangkaian cerita yang berurut, yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.



**Gambar 2.9** Struktur navigasi linier(**Prihatna, 2005**).

* + 1. **Struktur Navigasi Hirarki**

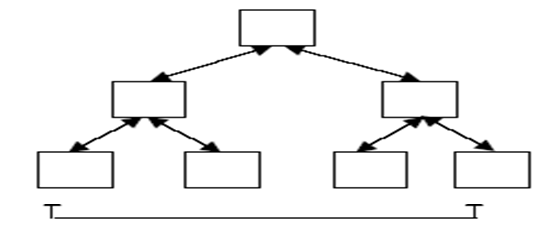
Struktur navigasi hirarki biasa disebut struktur bercabang, merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu satu akan disebut sebagai Master Page (halaman utama pertama), halaman utama ini mempunyai halaman percabangan yang disebut Slave Page (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama Master Page (halaman utama kedua) dan seterusnya. Pada navigasi ini tidak diperkenalkan adanya tampilan secara linier.



**Gambar 2.10** Struktur navigasi hirarki(**Prihatna, 2005**).

* + 1. **Struktur Navigasi Non-Linear**

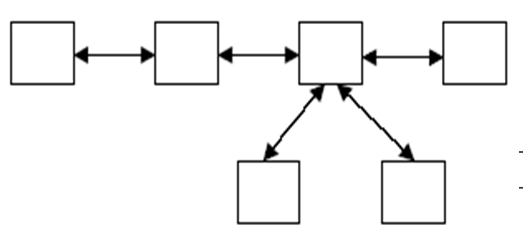
Struktur navigasi Non-linier merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan pada struktur non linier ini berbeda dengan percabangan pada struktur hirarki. Karena pada percabangan ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada Master Page dan Slave Page.



**Gambar 2.11** Struktur navigasi non-linear(**Prihatna, 2005**).

* + 1. **Struktur Navigasi *Composite* (Campuran)**

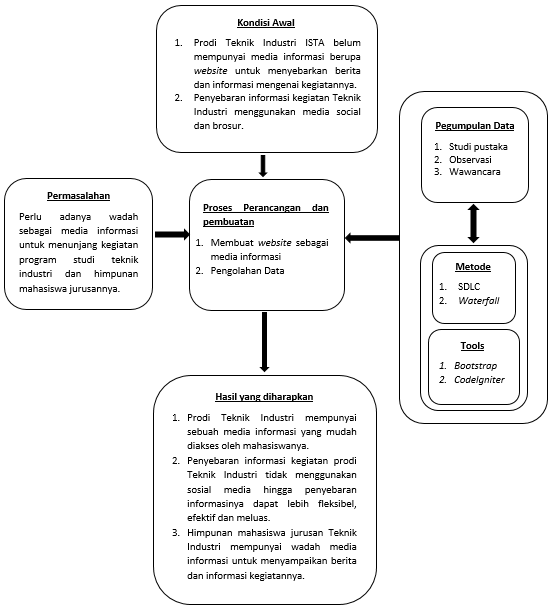
Struktur navigasi composite (campuran) disebut juga struktur navigasi bebas yang merupakan gabungan dari ketiga struktur yang ada. Struktur navigasi ini biasa digunakan dalam pembuatan multimedia karena dapat memberikan keinteraksian yang lebih tinggi.



**Gambar 2.12** Struktur navigasi campuran(**Prihatna, 2005**).

* 1. **Pola Pemikiran Pemecahan Masalah**

Berdasarkan hasil pengamatan sementara dan kajian teori yang telah disusun oleh peneliti, maka selanjutnya diperoleh pola pemikiran pemecahan masalah tentang pengembangan dan perancangan media informasi *website* sebagai wadah dan informasi kegiatan yang ada di prodi teknik industri.



**Gambar 2.13** Pola Pemikiran Pemecahan Masalah

Kerangka Pola Pemikiran Pemecahan Masalah pada gambar 2.13 menggambarkan tahapan proses pemecahan masalah dalam penelitian sebagai berikut

1. Kondisi Awal

Kondisi berjalan saat ini dimana program studi teknik industri institut sains dan teknologi al-kamal belum mempunyai sebuah wadah media informasi yang dapat di akses mahasiswanya dalam menunjang kegiatan program studi teknik industri. Sedangkan penyebaran informasi saat ini menggunakan brosur dan *media social* sehingga kurang efisien.

1. Permasalahan

Dibutuhkanya suatu wadah sebagai media informasi untuk menunjang kegiatan program studi teknik indsutri dan himpunan mahasiswa jurusanya

1. Metode Pengumpulan Data

Pada tahap ini dilakukan langkah-langkah seperti pengumpulan data, melakukan survei dan menganalisa permasalahan yang sedang terjadi, membuat perancangan media informasi berbasis website yang dapat menunjang kegiatan dari program studi teknik informatika dan himpunan mahasiswanya agar pesan yang akan disampaikan dapat menyeluruh dan mudah di akses.

1. Konsep yang digunakan dalam perancangan media informasi adalah Proses pembuatan media informasi berbasis website adalah *System development life cycle (SDLC)* dan menggunakan metode *waterfall* keseluruhan dalam pembangunan sistem melalui beberapa langkah yang akan digunakan nanti pada saat pembangunan system
2. Hasil yang diharapkan

Hasil dari pembuatan media informasi berbasis *website* ini antara lain, program studi teknik industri mempunyai sebuah wadah media informasi yang dapat diakses oleh dosen, mahasiswanya, dan Himpunan mahasiswa jurusan (HMJ) dimana dengan adanya *website* ini dapat memudahkan HMJ dalam menyampaikan informasi tentang kegiatannya.

* 1. **Manajemen Proyek Perangkat Lunak**
     1. **Pengertian Manajemen proyek Perangkat Lunak**

Proyek adalah urutan kegiatan yang unik, kompleks, dan saling terkait, memiliki satu tujuan harus diselesaikan dalam waktu tertentu, sesuai anggaran, dan memenuhi spesifikasi. Manajemen/pengelolaan proyek perangkat lunak bertujuan agar perangkat lunak yang dibuat sampai ke tangan pelanggan (*customer*) tepat waktu dan sesuai dengan harapan pelanggan.

* + 1. **Perencanaan Proyek**

Hal-hal yang harus masuk dalam perencanaan proyek adalah sebagai berikut:

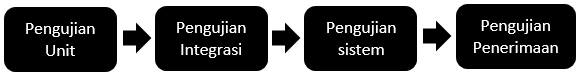
1. Pernyataan kerja yang merepresentasikan produk seperti spesifikasi perangkat lunak, rencana pengujian perangkat lunak, perencanaan kode program perangkat lunak, perencanaan laporan kesalahan/kecacatan (*defect*), dan semua perencanaan pekerjaan selama protek berjalan.
2. Daftar semua sumber daya (*resources*) yang dibutuhkan untuk pada rekayasa oerangkat lunak beserta ketersediaanya.
3. Rincian struktur pekerjaan dan kumpulan perkiraan pekerjaan yang dilakukan selama proses rekayasa perangkat lunak.
4. Perencanaan dan jadwal proyek

Hal ini harus didasari dengan perencanaan yang sangat detail dan estimasi yang tepat. Jika perencanaan jadwal tidak tepat maka bisa jadi jadwal yang telah ditetapkan menjadi molor.

* + 1. **Pengujian Perangkat Lunak**

Pengujian perangkat lunak adalah sebuah elemen sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi dan validasi atau (V&V). Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak mengimplementasikan dengan benar sebuah fungsi yang spesifik. Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun sesuai dengan kebutuhan *user*. Dapat juga dikatakan sebagai berikut:

1. **Verifikasi:** “Apakah perangkat lunak dibangun dengan benar?” (lebih ke arah apakah proses pengembangan perangkat lunak sudah benar dan telah berhasil mengimplementasikan fungsi yang benar)
2. **Validasi:** “Apakah sudah membangun perangkat lunak yang benar?” (lebih ke arah hasil perangkat lunak sudah sesuai dengan yang diinginkan)



**Gambar 2.14** Pengujian perangkat lunak (M. Shalahudin, 2016).

Pengujian integrasi sebaiknya dilakukan secara bertahap, untuk menghindari kesulitan penelusuran jika terjadi kesalahan (*error*). Pengujian integrasi lebih pada pengujian penggabungan dari dua atau lebih unit pada perangkat lunak. Setelah pengujian integrasi maka dilakukan pengujian system dimana unit-unit proses yang sudah diintegrasikan diuji dengan antarmuka yang sudah dibuat sehingga pengujian ini dimaksudkan untuk menguji perangkat lunak secara keseluruhan dan diuji secara satu sistem.

Pengujian untuk validasi memiliki beberapa pendekatan sebagai berikut:

1. ***Black-box Testing*** (Pengujian Kotak Hitam)

Yaitu menguji 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.

Misalkan untuk kasus proses *login* maka kasus uji yang dibuat adalah:

1. Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalkan nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.
3. ***White-box Testing*** (Pengujian Kotak Putih)

Yaitu menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pengujian kotak putih dilakukan dengan memeriksa lojik dari kode program. Contoh dari pengujian kotak putih misalkan menguji alur (dengan menelusuri) pengulangan (*looping*) pada logika pemrograman.