

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Kajian Pustaka**

Pembangunan sebuah aplikasi sistem akademik yang baik agar sesuai dengan kebutuhan, maka diperlukan referensi yang lain untuk membuat website seperti berikut:

Penelitian Ahmad Athoillah dan Marzuki (2016) membuat sistem informasi penerimaan siswa baru berbasis web pada SMK INFORMATIKA YASMA. Penerimaan siswa baru ini kelebihanannya dapat mengetahui secara jelas dan lengkap dan serta cepat tentang informasi yang dibutuhkan yang berkaitan tentang proses pendaftaran, mempermudah pencarian data calon siswa baru karena data yang tersimpan di dalam komputer sangat rapi dan mudah, dapat mengetahui secara jelas dan lengkap dan serta cepat tentang informasi yang dibutuhkan yang berkaitan tentang proses pendaftaran, dan pengumpulan laporan dan pengumpulan data calon siswa baru lebih cepat dan dapat diakses 24 jam selama koneksi internet tidak mengalami gangguan.

Penelitian Wirda Asrar (2013) membuat sistem informasi pendaftaran mahasiswa baru secara online pada yayasan sekolah tinggi ilmu kesehatan (STIKes) Harapan Bangsa Darussalam, Banda Aceh dengan menggunakan PHP dan Mysql. Sistem Informasi Pendaftaran Mahasiswa Baru Secara *Online* ini mampu membuat proses pendaftaran calon mahasiswa menjadi lebih mudah dan lebih cepat dan calon mahasiswa dapat mendaftar dimana saja dengan membuka *website* Yayasan Sekolah Tinggi Ilmu Kesehatan (STIKes) Harapan Bangsa Darussalam, Banda Aceh, dan dengan adanya Sistem Informasi Pendaftaran Mahasiswa Baru Secara *Online* memudahkan Yayasan Sekolah Tinggi Ilmu Kesehatan (STIKes) Harapan Bangsa Darussalam, Banda Aceh dalam mendapatkan data lengkap mengenai data calon mahasiswa yang mendaftar.

Penelitian Miftahus Sholihin dan Siti Mujilahwati (2014) membuat Sistem Penerimaan Peserta Didik Baru Berbasis Web (*Online*) Di SMK Muhammadiyah 7, Kedungpring, Lamongan. Sistem Penerimaan Peserta Didik Baru Berbasis Web

(Online) di SMK Muhammadiyah 7, Kedungpring, Lamongan ini mampu menangani kegiatan pendaftaran calon siswa, keluaran utama sistem ini yaitu menyajikan hasil pendaftar calon peserta didik baru melalui media internet atau *online* dan kemudahan dalam pendaftaran tanpa harus datang langsung, dan adanya sistem yang dapat melakukan pemasukan data, penyaringan data pendaftar sehingga dapat menyajikan informasi Sistem Penerimaan Peserta Didik Baru di SMK Muhammadiyah 7, Kedungpring, Lamongan.

## 2.2 Kerangka Berfikir

Kerangka pemikiran dari tugas akhir ini dapat dijelaskan atau didefinisikan sebagai berikut:

- 1) Latar belakang masalah  
Pokok permasalahan yang mendasari perlunya dibangun sistem pendaftaran siswa baru di SDN Pucangan 6 Kartasura.
- 2) Perumusan masalah  
Perumusan masalah merupakan inti permasalahan dan jalan keluar menyelesaikan permasalahan.
- 3) Pengumpulan data tertulis dan tidak tertulis  
Pada penelitian dilakukan pengumpulan data secara tertulis dan tidak tertulis pada SDN Pucangan 6 Kartasura. Pengumpulan data penelitian ini menggunakan metode observasi, dokumentasi, dan wawancara.
- 4) Penguasaan dasar  
Pada penelitian dilakukan percobaan membuat sistem agar lebih menguasai.
- 5) Observasi sistem  
Pada penelitian dilakukan pengamatan pada sistem yang sebelumnya digunakan agar dapat menjadi referensi dalam membangun aplikasi ini.
- 6) Analisis dan perancangan sistem  
Pada penelitian dilakukan menganalisa dan merancang bagaimana sistem nantinya akan dibuat untuk membantu memecahkan permasalahan yang ada.

7) Implementasi sistem

Pada penelitian dilakukan implementasi apa yang sudah dirancang untuk membuat sistem pendaftaran siswa baru di SDN Pucangan 6 Kartasura.

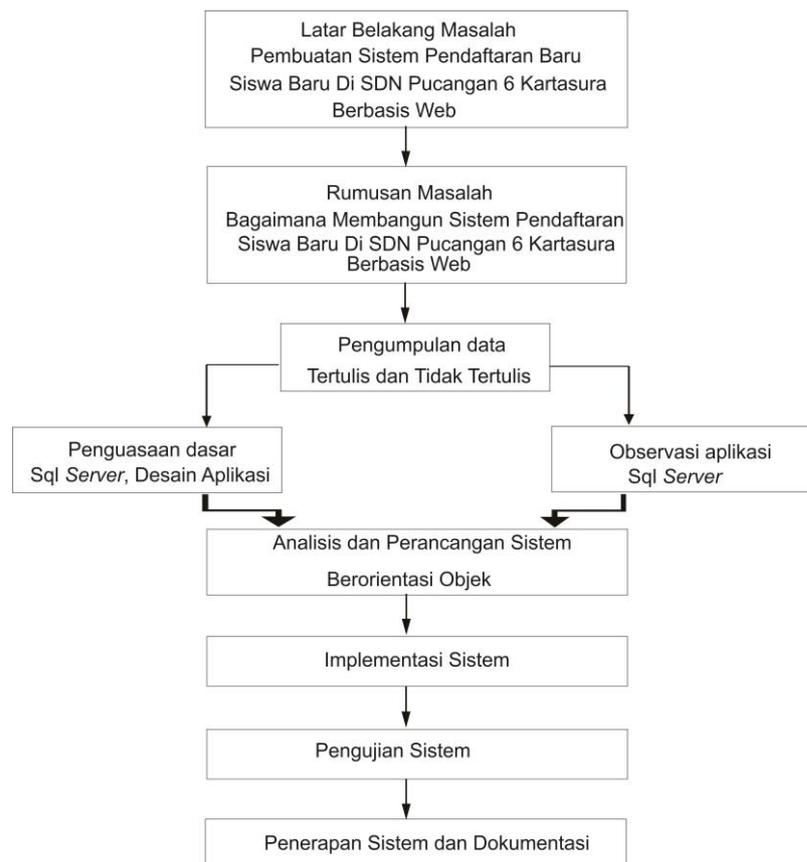
8) Pengujian sistem

Pada penelitian dilakukan uji coba aplikasi apakah masih terjadi kesalahan ataupun kekurangan pada sistem.

9) Penerapan sistem dan Dokumentasi

Sistem yang sudah diimplementasikan dan diuji coba kemudian diterapkan pada SDN Pucangan 6 Kartasura dan setelah itu dibuatnya dokumentasi dari keseluruhan kegiatan penyusunan Tugas Akhir.

Kerangka pemikiran dari penelitian tugas akhir ini digambarkan pada Gambar2.1



Gambar 2.1 Kerangka Pemikiran

## **2.3 Teori Pendukung**

### **2.3.1 Pengertian Sistem**

Menurut Abdul Kadir (2014), sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

Menurut Subhan (2012), sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang mengolah data menggunakan perangkat memegang peranan yang penting dalam sistem informasi. Data akan dimasukkan dalam sebuah sistem informasi dapat berupa formulir-formulir, prosedur-prosedur dan bentuk data lainnya.

### **2.3.2 Pengertian Penerimaan Siswa Baru**

Penerimaan merupakan penyambutan, proses, perbuatan atau sikap terhadap seseorang. Kata siswa merupakan pelajar pada akademi atau perguruan tinggi dan kata baru merupakan suatu hal belum ada sebelumnya (Desi Anwar, 2003).

### **2.3.3 Metode *Project Web***

Pada tahap ini metode yang digunakan adalah metode *Project Web* menurut Richar Jain, 2014 adalah proses pembuatan *web* dengan tahap-tahap utama dari model ini memetakan kegiatan-kegiatan pengembangan dasar yaitu:

#### 1) Definisikan Tujuan Proyek

Ini merupakan langkah awal yang harus di lakukan bersama dengan *web developer*. Walaupun banyak usaha yang sudah memiliki gagasan yang jelas mengenai apa yang mereka inginkan, masih banyak bisnis baru atau kecil yang masih belum tahu apa yang ingin di lakukan. Dalam tahap awal ini harus melalui fase sebagai berikut :

##### a) Tujuan Usaha dan *Website*

Menjelaskan mengenai usaha, nilai dan visi misi kepada *web developer* dan menyampaikan pula mengenai target yang ingin dicapai dan

tujuan yang diinginkan melalui *website*. Rancangan dibuat agar *website* tidak hanya terlihat indah, namun juga bisa melibatkan pelanggan atau pengunjung yang berujung membantu memperoleh pendapatan.

b) *Fitur dan Spesifikasi*

Pada tahap ini akan menentukan aspek teknis yang akan dibutuhkan. Apakah akan membutuhkan *website* yang dioptimasi untuk *desktop* atau perangkat *mobile*? Jika ingin *website* statis atau *website* berbentuk *blog/podcast*? Lalu apakah ingin membuat komunitas yang aktif pada *website*, atau malah memilih untuk mengaktifkannya melalui media sosial?

Banyak aspek teknis yang harus dipertimbangkan untuk diterapkan pada *website* yang akan dibuat. Pilihan-pilihan itu akan memandu untuk memilih mana *fitur* dan *spesifikasi* yang paling cocok.

c) *Sumber Konten*

Jika sudah memiliki merek yang pasti, harus memberitahukan hal itu kepada *web developer*. Jika tidak, maka kemungkinan *web developer* akan memberikan sendiri logo dan merek untuk usaha. Sampaikan dengan jelas konten seperti apa yang ingin ditampilkan pada *website*, mulai dari informasi perusahaan, gambar, video dan lainnya.

d) *Keperluan vs Layanan Web Developer*

Beberapa *web developer* hanya menyediakan desain atau *wireframes*, beberapa lainnya hanya membuat *website statis*, *web developer* lainnya ada yang akan mengintegrasikan forum atau *e-commerce*. Kemudian *web developer* yang lainnya akan menjalankan *blog* atau media sosial sebagai bagian dari paket yang ditawarkan.

Namun harus diingat bahwa tidak semua *web developer* menawarkan hal-hal tersebut, karena itu harus dengan jelas menyampaikan apa-apa saja yang harus dikerjakan oleh *web developer*. Jelaskan juga seperti apa perpindahan pengelolaan *website* dari *web developer* ke pemilik setelah *website* telah selesai.

Pada tahap ini, pastikan komunikasi pemilik dengan *web developer* berjalan dengan baik dan jelas untuk menghindari kekurangan dan kesalahan pada pembuatan *website*. Hal ini juga akan membantu *web developer* dalam memberikan perkiraan mengenai waktu dan anggaran yang dibutuhkan.

## 2) Penelitian dan Perencanaan

Ketika selesai dengan tahap definisi proyek, maka *web developer* dapat melanjutkan pekerjaan ke tahap perencanaan. Tahap ini membutuhkan penelitian dan diskusi untuk menguraikan ide-ide dan konsep yang akan diterapkan pada *website*. Pemilik bersama dengan *web developer* dapat saling bertukar pikiran mengenai ide-ide tertentu. Ada beberapa fase yang harus dilakukan pada tahap ini :

### a) Profil Pengunjung *Website*

Pemilik memberikan penjelasan mengenai target pelanggan, kemudian *web developer* akan melakukan penelitian. *Web developer* akan mencari tahu beberapa hal seperti misalnya gaya atau kebiasaan target pelanggan atau pengunjung, desain apa yang cocok, serta tema dan konten apa yang sesuai.

### b) Arsitektur

Hal ini merupakan definisi dasar dari struktur konten dan bagaimana pelanggan akan mengaksesnya. Arsitektur meliputi beberapa rincian seperti jumlah halaman yang diperlukan *website*, bagaimana pelanggan atau pengunjung akan ditempatkan dan bagaimana cara pelanggan atau pengunjung bernavigasi melaluinya.

### c) *Page Layout*

Pada tahap ini, *web developer* akan membantu Pemilik untuk memilih tipe *layout* yang sesuai dengan setiap halaman pada *website*, termasuk jenis menu apa yang diinginkan. Pemilik juga bisa memilih berbagai posisi untuk teks, media, gambar, dan sebagainya.

d) *Mockup/Wireframe*

Setelah struktur dasar dan *layout* sudah ditetapkan, maka hal selanjutnya yang harus diperhatikan adalah *mockup* atau *wireframe*. Hal ini akan membuat pemilik melihat dan merasakan bagaimana komponen-komponen *website* yang akan distrukturkan. Hal ini juga membantu menyampaikan visi yang dimiliki oleh bagian desainer secara jelas.

e) *Infrastruktur*

Keputusan mengatur infrastruktur termasuk sejumlah hal seperti menentukan nama *domain*, mendapatkan *hosting*, memilih CMS yang digunakan, menentukan bahasa pemrograman dan lainnya.

3) *Merancang Desain*

Setelah perencanaan dasar selesai dilakukan, desainer akan melihat data-data tersebut untuk mengembangkan tampilan dan nuansa *website*. Proses desain meliputi langkah-langkah berikut :

a) *Skema Warna*

Skema warna menetapkan ‘nada’ untuk pengalaman pengunjung *website*. Warna cerah dan *multi-chrome* misalnya, cocok untuk pelanggan ataupun pengunjung yang muda. Sedangkan warna ‘bersih’ dan ‘halus’ lebih cocok untuk usaha-usaha seperti keuangan, legal, hukum dan lainnya.

b) *Elemen Visual*

*Elemen visual* pada *website* diantaranya adalah tombol, menu, tipografi dan lainnya. Semua hal itu harus bisa merepresentasikan merek.

c) *Media*

Media pada *website* adalah gambar, video dan konten audio. Seperti apa jenis gambar yang digunakan, bagaimana merepresentasikan pelanggan, bagaimana penempatan dan pergeseran pelanggan, dan lainnya harus Pemilik putuskan.

#### 4) Pengembangan

Ini saatnya *web developer* bekerja melakukan *coding* untuk ‘menghidupkan’ semua perencanaan dan perancangan yang telah dilakukan sebelumnya. Hal ini biasanya dilakukan pada *server internal* sehingga seluruh dunia belum bisa mengakses *website*.

##### a) *Template/CMS*

Pemilik bisa memilih untuk menggunakan kerangka kerja, template yang sudah ada sebelumnya, sistem manajemen konten standar seperti *WordPress* atau hanya mengolah *website* dengan *coding manual*.

##### b) Standar Web dan Bahasa Pemrograman

Berdasarkan kerangka dan jenis fungsional yang diperlukan, *web developer* akan mengumpulkan alat-alat dan bahasa pemrograman yang cocok. *Web developer* juga harus memperhatikan standar web untuk HTML, CSS dan *JavaScript*.

##### c) Database

Halaman, *template*, *layout* gambar dan kode sebuah *website* disimpan dalam *database*. Pemilik juga bisa memilih *database SQL* atau *database* lainnya yang perlu dikonfigurasi dan dikelola dengan tepat.

##### d) Fitur

Setelah memiliki kerangka dan *database*, maka bisa dilanjutkan ke proses *coding* fitur. Setiap *web developer* memiliki gayanya yang berbeda untuk melakukan *coding*.

##### e) Keamanan

Salah satu aspek kunci *website* adalah keamanan. Penting untuk menjaga *website* beserta konten dan juga keamanan data pengguna seperti alamat *email*, *password* atau informasi kartu kredit pelanggan.

##### f) Kinerja

Penambahan banyak fitur atau gambar resolusi tinggi pada *website* bisa memperlambat kinerja *website*. Waktu membuka halaman *website* adalah salah satu aspek paling krusial yang mempengaruhi retensi

pelanggan atau pengguna. Optimasi *website* agar bisa loading dengan cepat, lakukan hal ini pada tahap awal pengembangan untuk menghindari masalah di masa depan.

## 5) *Launching*

Dalam *launching* ini ada beberapa hal yang harus dilewati:

### a) Pengujian Beta

Ini adalah tahap akhir pengujian dimana *web developer* akan memeriksa fungsionalitas, kinerja dan keamanan *website* Anda. Umumnya *web developer* akan melibatkan Anda untuk memastikan semua hal yang ada sudah cocok dengan kebutuhan dan pengalaman pengguna yang Anda inginkan.

### b) Pengiriman

Setelah *website* telah diverifikasi, maka selanjutnya *web developer* akan mengirimkan kode, *database* dan aset lainnya untuk nama domain dan *hosting* Anda. *Web developer* yang baik akan menyertakan pengaktifan berbagai layanan seperti *web analytic* yang kemungkinan akan digunakan oleh Anda. Pada tahap ini, *web developer* juga mengirimkan kontrol ke Anda, termasuk *hosting*, panel kontrol dan administrasi lainnya.

### c) Pelatihan dan Dokumentasi

Terdapat *web developer* yang menyediakan jasa untuk dokumentasi dan memberikan pelatihan untuk Anda agar bisa mengelola aspek dasar pemeliharaan *website*.

## 6) Pemeliharaan

Beberapa *web developer* dan kliennya mungkin bisa tergoda untuk yakin bahwa tidak akan terjadi sesuatu yang perlu dikhawatirkan setelah *website* selesai diluncurkan. Namun, *website* sebenarnya memerlukan perawatan berkala. Akan lebih baik, jika meminta *web developer* untuk memberikan pelayanan dalam merawat *website*.

Hal tersebut mencakup :

a) *Support dan Troubleshoot*

Aspek ini termasuk keseluruhan area luas yang mencakup segala jenis masalah yang ditemukan pengguna pada *website*, masalah *hosting*, keamanan dan lainnya.

b) *Perpanjangan Periodik*

Pembuat mungkin saja perlu untuk memperbaharui nama domain (tergantung pada pembelian awal), *hosting* dan izin lainnya secara berkala. Dianjurkan untuk memperbarui konten *website* pemilik seperti informasi perusahaan atau produk dan perubahan lainnya dari waktu ke waktu.

c) *Blog/Live Content*

Jika ingin terlibat dengan pelanggan dan ingin pelanggan kembali ke *website* pembuat, maka disarankan untuk memiliki *blog* atau konten baru lainnya seperti *blog*, *podcast* atau video yang di posting secara teratur.

d) *Optimasi*

Untuk membuat *website* 'terlihat', maka harus menggunakan beberapa teknis *Search Engine Optimizations* dan metode lainnya untuk menaikkan *traffic*.

### 2.3.4 Internet

Menurut Ahmadi dan Hermawan (2013), Internet adalah komunikasi jaringan komunikasi global yang menghubungkan seluruh komputer di dunia meskipun berbeda sistem operasi dan mesin, *interconnection network (internet)* adalah sistem global dari seluruh jaringan komputer yang saling terhubung. Internet berasal dari bahasa latin "inter" yang berarti "antara". Internet merupakan jaringan yang terdiri dari milyaran komputer yang ada di seluruh dunia. Internet melibatkan berbagai jenis komputer serta topologi jaringan yang berbeda. Aplikasi-aplikasi dasar yang paling sering dimanfaatkan oleh pengguna internet, antara lain:

### 1) WWW (*World Wide Web*)

WWW atau yang sering disebut sebagai “*web*” saja merupakan aplikasi internet yang paling populer. Demikian populernya hingga banyak orang yang keliru mengidentikkan *web* dengan internet. Secara teknis, *web* adalah sebuah sistem dimana informasi dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah internet *web server* dipresentasikan dalam bentuk hypertext. Informasi di *web* dalam bentuk teks umumnya ditulis dalam format HTML (*Hipertext Markuo Language*). Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG), suara (dalam format AU, WAV), dan objek multimedia lainnya (seperti MIDI, *Shockwave*, *Quicktime Movie*, *3D World*). *Web* dapat di akses oleh perangkat lunak *web client* yang secara populer disebut sebagai *browser*. *Browser* membaca halaman - halaman *web* yang tersimpan dalam *webserver* melalui protokol yang disebut HTTP (*Hypertext Transfer Protocol*). Dewasa ini, tersedia beragam perangkat lunak *browser*. Beberapa diantaranya cukup populer dan digunakan secara meluas, contohnya seperti *Microsoft Internet Explorer*, *Netscape Navigator*, maupun *Opera*. Namun ada juga beberapa produk *browser* yang kurang dikenal dan hanya digunakan di lingkungan yang terbatas (Lenawati, 2007).

### 2) *EMail (Electronic Mail)*

*Email* atau dalam istilah Indonesia, “surat elektronik” adalah aplikasi yang memungkinkan para pengguna internet untuk salingberkirim pesan melalui alamat elektronik di internet untuk saling berkirim pesan melalui alamat elektronik di internet. Para pengguna *email* memiliki sebuah *mailbox* (kotak surat) elektronik yang tersimpan dalam suatu *mailserver*. Suatu *Mailbox* memiliki sebuah alamat sebagai pengenalan agar dapat berhubungan dengan *mailbox* lainnya, baik dalam bentuk penerimaan maupun pengiriman pesan. Pesan yang diterima akan ditampung dalam *mailbox*, selanjutnya pemilik *mailbox* sewaktu-waktu dapat mengecek isinya, menjawab pesan, menghapus, atau menyunting dan mengirimkan pesan *email* (Kurniawan, 2004).

### 3) FTP (*File Transfer Protocol*)

FTP memungkinkan para pengguna internet untuk melakukan pengiriman (*upload*) atau menyalin (*download*) sebuah *file* antara komputer lokal dengan komputer lain yang terhubung dalam jaringan internet. FTP umumnya dimanfaatkan sebagai sarana pendukung untuk kepentingan pertukaran maupun penyebarluasan sebuah file melalui jaringan internet. FTP juga dimanfaatkan untuk melakukan proses *upload* suatu halaman *web* ke *webserver* agar dapat diakses oleh pengguna internet lainnya (Rahmat Rafiudin, 2013).

#### 2.3.5 Basis Data

*Database* adalah kumpulan data (elementer) yang secara logik berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu. Basisdata adalah kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi.

Basis data mendeskripsikan *state* organisasi, perusahaan, sistem. Saat satu kejadian muncul di dunia nyata mengubah *state* organisasi, perusahaan atau sistem maka satu perubahan pun harus dilakukan terhadap data yang disimpan di basisdata. Basis data merupakan komponen utama sistem informasi karena semua informasi untuk pengambilan keputusan berasal dari data di basis data. Pengelolaan basisdata yang buruk dapat mengakibatkan ketidakterediaan data penting yang digunakan untuk menghasilkan informasi yang diperlukan dalam pengambilan keputusan (Hariyanto, 2004).

#### 2.3.6 PHP 5

PHP kependekan dari *Hypertext Preprocessor* yang dibangun yang dibangun oleh Rasmus Lerdorf pada awal pengembangan PHP disebut sebagai *Personal Home Page*. PHP merupakan produk *open source* sehingga kita dapat mengakses *source code*, menggunakan dan mengubahnya tanpa harus membayar sepeserpun.

PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dilajankan pada *server side*. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada server sedangkan yang dikirim ke *browser* hanya hasil saja. PHP mampu berjalan melalui *platform* seperti *Windows*, *Unix* serta varian *Linux*. Beberapa kelebihan PHP antara lain dapat membuat situs yang interaktif dengan forum diskusi, *guestbook* dan sebagainya, koneksitas yang baik dengan bermacam-macam *database* seperti *Oracle*, *PostgreSQL*, dan lain-lain (Nugraha, 2010).

PHP 5 saat ini memiliki fitur yang cukup banyak. Menjadikan bahas pemrograman PHP mendapatkan penghargaan dari masyarakat dunia, melalui *Zend Technologies*(*Zend Engine*) sebagai pengembangnya. Keluarga PHP 5.x mendukung sepenuhnya teknik *Object Oriented Programming* atau OOP, menjadikan bahasa ini sebagai salah satu yang terbaik didalam mengembangkan aplikasi *web* yang besar. Sekalipun demikian penggunaan teknik modulasi atau fungsi masih tetap digunakan sampai pada dekade terakhir ini karena merupakan unsur utama dalam penggunaan teknik OOP menurut Sakur (2010).

Pada Juni 2004, Zend merilis kembali versi PHP 5.0 perubahan sangat besar sekali terjadi karena paradigmanya telah berubah menjadi suatu bahasa pemrograman yang berorientasi pada objek (OOP) (Saputra, 2013).

### 2.3.7 HTML 5

Menurut Iqbal Dkk (2012), HTML5 (*Hypertext Markup Language version 5*) adalah sebuah bahasa markah yang menstrukturkan isi dari *World Wide Web*, sebuah teknologi utama pada internet. Standar HTML5 menyempurnakan elemen-elemen lama yang terdapat pada standar sebelumnya, menambahkan elemen-elemen yang lebih semantik dan menambahkan fitur-fitur baru untuk mendukung pembuatan aplikasi *web* yang lebih kompleks. Berikut fitur-fitur baru HTML5 antara lain :

#### 1) *Canvas*

*Canvas* merupakan elemen yang digunakan untuk menggambar grafik menggunakan *scripting*. Untuk menggunakan elemen *canvas* harus dengan

melakukan *scripting*, *canvas* mempunyai beberapa metode untuk menggambar suatu kotak, lingkaran, karakter, atau memasukan suatu gambar.

#### 2) *Scalable Vector Graphics*

*Scalable Vector Graphics* (SVG) digunakan untuk menentukan suatu grafik berbasis *vector* untuk *web*, grafik ini menggunakan format XML, yang berarti tidak akan kehilangan fokus atau tidak akan terjadi *blur* ketika dilakukan proses *zoom*.

#### 3) *Drag/Drop*

*Drag and drop* merupakan fitur yang sudah sangat dikenal bagi pengguna computer, tetapi di dalam HTML5 ketika di lakukan suatu “*drag*” objek atau melakukan penarikan objek yang di dapat menaruhnya di lokasi yang berbeda.

#### 4) *Geolocation*

HTML5 *geolocation* API digunakan untuk mengetahui lokasi penggunanya, tetapi karena ini menyangkut kerahasiaan pengguna, hal ini hanya dapat dilakukan ketika *user* yang dituju menyetujui untuk melakukannya.

#### 5) *Video*

HTML5 menyediakan elemen baru untuk menampilkan sebuah video didalam *website*, elemen ini juga dapat mengatur tinggi dan lebar atau disebut resolusi dari video tersebut.

### 2.3.8 MySQL

MySQL *Improved* atau MySQLi, merupakan peningkatan pengaksesan terhadap database MySQL, yang merupakan perkembangan dari PHP 5.x atau yang terbaru. Dengan menggunakan MySQLi, maka dapat di akses seluruh fungsi-fungsi dari MySQL versi 4.1 ke atas, termasuk di dalamnya dapat mengakses *Stored Procedure*, *Stored Function* atau *Prepare Statement*. *Stored Prosedure* dan *Function* serta *Trigger* merupakan komponen dasar dari SQL yang sudah didukung Oleh MySQL pada versi 5.0.2 atau yang terbaru.

Penggunaan MySQLi, merupakan opsional bagi pemrograman PHP dalam arti bahwa setiap pemrogram dapat saja tidak menggunakan APIs ini, akan tetapi tidak akan dapat memanfaatkan kemampuan MySQL yang lebih baru.

MySQLi dikembangkan dengan tujuan utama agar PHP dapat mengakses seluruh fasilitas-fasilitas terbaru yang ada pada MySQL 4.1 atau lebih baru. Sehingga secara tidak langsung fitur APIs ini menjadi kewajiban bagi pemrogram PHP ketika menggunakan database MySQL 5.0 atau lebih tinggi. MySQLi *extension* memiliki sejumlah manfaat, yang merupakan peningkatan dari MySQL *extension*, diantaranya adalah :

- 1) Menggunakan *Interface Object Oriented*
- 2) Mendukung penggunaan *Prepared Statement*
- 3) Mendukung penggunaan *Multiple Statement*
- 4) Mendukung penggunaan *Transaction*
- 5) Peningkatan terhadap kemampuan pencarian kesalahan program (*Debugging*)
- 6) Mendukung penggabungan dengan server.

Sekalipun menggunakan interface *object-Oriented*, MySQLi *extension* juga menyediakan pengaksesan dengan cara prosedural menurut Sakur (2010).

### 2.3.9 Apache

*Apache* adalah server web yang handal dan paling banyak digunakan oleh para *administrator* yang menggunakan sistem operasi *Unix*. Walaupun banyak digunakan pada sistem operasi *unix*, *apache* ini juga dapat digunakan pada operasi *Windows NT/9x*, *Windows 2000*, *Netware 5.x*, dan *OS/2*. Selain handal, *Apache* adalah server web yang fleksibel dan mengimplementasikan protokol-protokol web terbaru seperti HTTP/1.1 (RFC 2616). Salah satu sebab kenapa *Apache* banyak digunakan karena sifat dari software *Apache* sendiri yang *opensource* dan tidak menggunakan lisensi dalam pemakaian *software* tersebut (Taufan dan Riza, 2002).

### 2.3.10 Model WebQual

*WebQual* yang merupakan salah satu teknik pengukuran kualitas *website* berdasarkan persepsi pengguna. Barnes dan Vidgen (2002, 2003, 2005) sebagai pengembang metode *WebQual* telah melakukan beberapa penelitian menggunakan metode *WebQual* 4.0 untuk mengevaluasi beberapa *website* baik *website* non pemerintah (*ecommerce*) maupun *website* pemerintahan (*e-government*) yang mengacu pada tiga dimensi kualitas, yaitu *usability quality*, *information quality*, dan *service interaction*.

Beberapa penelitian selanjutnya telah memperluas dan mengembangkan metode *WebQual* 4.0 menjadi beberapa variabel *WebQual* untuk menilai kualitas sebuah *website*. Nasution dan Mudjahidin (2013) melakukan penelitian untuk mengukur kualitas *website* pemerintahan dari sisi kepuasan pengguna akhir dan keinginan pengunjung situs untuk menggunakan kembali layanan *website*, yang menitikberatkan pada sejauh mana persepsi tentang mutu layanan *website* yang dirasakan (*actual*) dengan tingkat harapan (*ideal*). Penelitian Nasution dan Mudjahidin (2013) menggunakan tiga dimensi utama *WebQual* 4.0 (*usability quality*, *information quality*, dan *service interaction*) ditambahkan dengan dimensi keempat yaitu *design*.

Loiacono, dkk. (2002) mengembangkan metode *WebQual* dan memperkenalkan *WebQual*<sup>TM</sup>, sebuah ukuran kualitas *website* dengan 12 dimensi didasarkan pada tinjauan literatur yang luas dan wawancara dengan desainer dan pengunjung *website*. Lima kategori umum kualitas *website* yang didapatkan dari kajian literatur dan eksplorasi penelitian Loiacono, dkk. (2002), yaitu: *ease of use*, *usefulness*, *entertainment*, *complementary relationship*, dan *customer service*.

Khawaja dan Bokhari (2010) mengidentifikasi sembilan dimensi *University WebQual* (*reliability*, *navigability*, *responsiveness*, *efficiency*, *functionality*, *usefulness*, *ease of use*, *accuracy*, dan *web appearance*) untuk mengukur layanan kualitas *website* universitas ditambah efeknya pada kepuasan siswa. Dalam penelitian ini, metode *WebQual* yang digunakan adalah *WebQual* versi 4.0 yang telah dimodifikasi dengan menambahkan dimensi kualitas antarmuka pengguna (*user*

*interface quality*) dengan tiga dimensi utama *WebQual* 4.0, yaitu kualitas kegunaan (*usability quality*), kualitas informasi (*information quality*) dan layanan interaksi (*service interaction*). Penambahan dimensi kualitas antarmuka pengguna (*user interface quality*) dalam penelitian ini adalah untuk mengukur kualitas antarmuka *website* yang berkaitan dengan daya tarik visual *website* karena pengguna *website* akan melihat tampilan *website* sebagai interaksi pertama.

## 2.4 Pengembangan Sistem Beorientasi Objek

Menurut Jogiyanto (2005) analisis sistem adalah penguraian dari suatu sistem informasi yang utuh kedalam bagian bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

Dalam analisis dan perancangan sistem penulis menggunakan metode UML. Menurut Hend (2006) “*Unified Modeling Language (UML)* adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak”.

Menurut Adi Nugroho (2005). “Bangunan dasar metodologi *Unified Modeling Language (UML)* menggunakan tiga bangunan dasar untuk mendeskripsikan sistem/perangkat lunak yang akan dikembangkan yaitu :

### 1) Sesuatu (*things*)

Ada 4 (empat) *things* dalam *Unified Modeling Language (UML)*, yaitu:

#### a) *Structural things*

Merupakan bagian yang relative statis dalam model *Unified Modeling Language (UML)*. Bagian yang relative statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.

#### b) *Behavioral things*

Merupakan bagian yang dinamis pada model *Unified Modeling Language (UML)*, biasanya merupakan kata kerjadari model *Unified*

*Modeling Language* (UML), yang mencerminkan perilaku sepanjang ruang dan waktu.

c) *Grouping things*

Merupakan bagian pengorganisasi dalam *Unified Modeling Language* (UML). Dalam penggambaran model yang rumit kadang *diperlukan* penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokkan sesuatu, misalnya model-model dan subsistem-subsistem.

d) *Annotational things*

Merupakan bagian yang memperjelas model *Unified Modeling Language* (UML) dan dapat berupa komentar-komentar yang menjelaskan fungsi serta ciri-ciri setiap elemen dalam model *Unified Modeling Language* (UML).

2) Relasi (*Relationship*)

Ada 4 (empat) macam *relationship* dalam *Unified Modeling Language* (UML), yaitu :

a) Kebergantungan

Merupakan hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).

b) Asosiasi

Merupakan apa yang menghubungkan antara objek satu dengan objek lainnya, bagaimana hubungan suatu objek dengan objek lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.

c) Generalisasi

Merupakan hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). Arah dari atas kebawah dari objek induk ke objek anak dinamakan spesialisasi,

sedangkan arah berlawanan sebaliknya dari arah bawah keatas dinamakan generalisasi.

d) Realisasi

Merupakan operasi yang benar-benar dilakukan oleh suatu objek.

3) Diagram

Menurut Rosa dan Shalahuddin (2013) pada perkembangan teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan mendokumentasi dari sistem perangkat lunak.

Menurut Sukamto dan Shalahuddin (2013) dijelaskan *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. Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an. Pada 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 *Unified Modeling Language* (UML) diakomodasi oleh *Object Management Group* (OMG) sehingga sampai saat ini(UML) telah memberikan kontribusinya yang cukup besar dalam metodologi berorientasi objek.

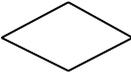
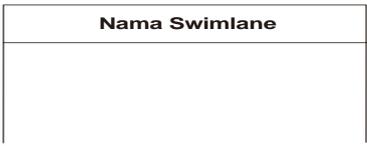
Ada 5 (empat) macam diagram dalam *Unified Modeling Language* (UML), yaitu :

a) *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem proses bisnis atau menu yang ada pada perangkat lunak, yang perlu diperhatikan di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang

dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol – simbol *Activity Diagram* ditunjukkan pada Tabel 2.1

Tabel 2.1 Simbol – simbol *Activity Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1.		Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.
3.		Percabangan / <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan / <i>Join</i>	Asosiasi penggabungan lebih dari satu aktivitas digabungkan menjadi satu.
5.		Status Akhir	Status akhir dilakukan sebuah sistem.
6.		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Rosa dan Shalahuddin (2013)

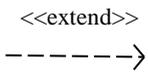
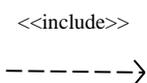
#### b) *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk kelakuan (*behaviour*) 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 didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sederhana 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 diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

2. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.. Simbol – simbol *Use Case Diagram* ditunjukkan pada Tabel 2.2.

Tabel 2.2 Simbol – simbol *Use Case Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor.
2.		Aktor / <i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
3.		Asosiasi / <i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> memiliki interaksi dengan aktor.
4.		Ekstensi / <i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5.		Generalisasi / <i>Generalization</i>	Hubungan generalisasi dan spesifikasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.		Menggunakan / <i>Include / uses</i>	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.

Sumber : Rosa dan Shalahuddin (2013)

c) *Class Diagram*

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

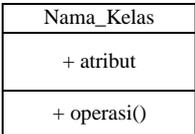
1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Kelas main, kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem (*view*), kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
3. Kelas yang diambil dari pendefinisian *use case (controller)*, kelas yang menangani fungsi-fungsi yang harus ada di ambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.
4. Kelas yang diambil dari pendefinisian data, kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Semua tabel yang dibuat di basis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut *multivalued* pada ERD dapat dijadikan kelas tersendiri dapat juga tidak, asalkan aksesannya dapat dipertanggungjawabkan atau tetap ada di dalam perancangan kelas.

Simbol – simbol *Class Diagram* Ditunjukkan pada Tabel 2.3.

Tabel 2.3 Simbol – simbol *Class Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		Kelas	Kelas pada struktur sistem.
2.		Antarmuka / <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.		Asosiasi / <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga di sertai dengan <i>multiplicity</i> .
4.		Asosiasi Berarah / <i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain. asosiasi biasanya juga di sertai dengan <i>multiplicity</i> .
5.		Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6.		Kebergantungan / <i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.		Agregasi / <i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

Sumber : Rosa dan Shalahuddin (2013)

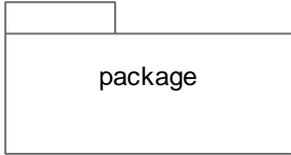
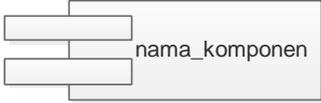
#### d) *Component Diagram*

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

Hubungan antara *component* dan *class*, *Component* adalah implementasi *software* dan sebuah *class*. *Class* mewakili abstraksi dari

serangkaian *attribute* dan *operation*. Hal terpenting yang perlu diingat tentang *class* dan *component* adalah sebuah *component* bisa jadi merupakan implementasi dari lebih dari sebuah *class*. Simbol *Component Diagram* disajikan pada Tabel 2.4.

Tabel 2.4 Simbol *Component Diagram*.

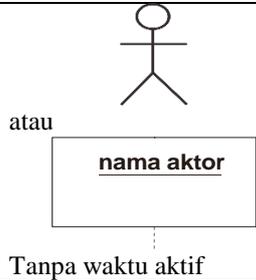
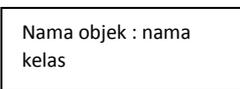
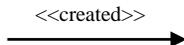
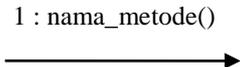
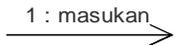
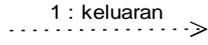
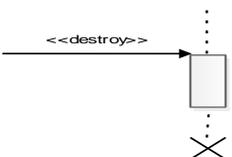
NO.	SIMBOL	NAMA	KETERANGAN
1.	 package	<i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen.
2.	 nama_komponen	Komponen	Komponen sistem.
3.		Kebergantungan / <i>Depedency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang di pakai.
4.	 nama_interface	Antarmuka / <i>Interface</i>	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen.
5.		<i>Link</i>	Relasi antar komponen.

Sumber : Rosa dan Shalahuddin (2013)

e) *Sequence Digram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan di terima antar objek, oleh karena itu untuk menggambar diagram sekuen maka harus di ketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek. Simbol *Sequence Diagram* ditunjukkan pada Tabel 2.5.

Tabel 2.5 Simbol *Sequence Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.	 <p>atau</p> <p>Tanpa waktu aktif</p>	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat di luar sistem informasi yang akan di buat sendiri.
2.		Garis Hidup	Menyatakan kehidupan suatu objek.
3.		Objek	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang di buat.
4.		Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
5.		Pesan Tipe <i>Create</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.
6.		Pesan Tipe <i>Call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
7.		Pesan Tipe <i>Send</i>	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.		Pesan Tipe <i>Return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarahkan pada objek yang menerima kembalian.
9.		Pesan Tipe <i>Destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber : Rosa dan Shalahuddin (2013)

f) *Deployment Diagram*

Diagram komponen di buat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut :

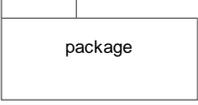
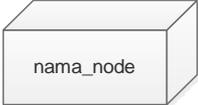
1. *Source code* program perangkat lunak.
2. Komponen *executable* yang di lepas ke *user*.
3. Basis data secara fisik.
4. Sistem yang harus beradaptasi dengan sistem lain.
5. *Framework* sistem, *framework* pada perangkat lunak merupakan kerangka kerja yg di buat untuk memudahkan pengembangan dan pemeliharaan aplikasi.

Komponen dasar yang biasanya ada dalam suatu sistem adalah sebagai berikut :

1. Komponen *user interface* yang menangani tampilan.
2. Komponen *bussiness procesiing* yang menangani fungsi-fungsi proses bisnis .
3. Komponen data yang menangani manipulasi data.
4. Komponen *security* yang menangani keamanan sistem.

Komponen lebih berfokus pada penggolongan secara umum fungsi-fungsi yang diperlukan. Simbol – simbol *Deployment Diagram* ditunjukkan pada Tabel 2.6

Tabel 2.6 Simbol – simbol *Deployment Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih <i>node</i> .
2.		<i>Node</i>	Mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak yang tidak di buat sendiri ( <i>software</i> ) jika di dalam <i>node</i> disertakan komponen untuk mengkonsistensikan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefenisikan sebelumnya pada diagram komponen.
3.		Kebergantungan / <i>Depedency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang di pakai.
4.		<i>Link</i>	Relasi antar <i>node</i> .

Sumber : Rosa dan Shalahuddin (2013)