

BAB II

LANDASAN TEORI

Bab ini akan membahas mengenai tinjauan pustaka yang berisi pustaka dan hasil penelitian yang pernah dilakukan, yang mana isi pustaka berhubungan dengan penelitian ini, kerangka pemikiran, serta landasan teori yang membahas teori-teori dasar pendukung untuk penelitian ini.

2.1. Tinjauan Pustaka

Penelitian mengenai Pembangunan *Website* Sekolah SMK Islam SUDIRMAN KADUNGGATI (Studi kasus Sekolah SMK Islam Sudirman Kedungjati) yang dilakukan oleh Muhammad Sam Soleh (2013) dimana dalam Tugas Akhir tersebut membahas tentang pembuatan *website* sekolah sebagai media informasi yang menyediakan artikel maupun informasi data siswa dengan lebih cepat. **Kelebihan** : sebagai media informasi dan menyediakan data siswa dengan lebih cepat. **Kekurangan** : tidak ada proses yang berhubungan dengan pendaftaran.

Penelitian tentang Pembuatan *Website* SMA Negeri 1 Pracimantoro Menggunakan *PHP & MySQL* yang dilakukan oleh Yoga Pujantoko (2011) dimana dalam Tugas Akhir tersebut membahas tentang pembuatan *website* SMA Negeri 1 Pracimantoro sebagai media informasi. **Kelebihan** : sebagai media informasi dan menyediakan data siswa dengan lebih cepat. **Kekurangan** : tidak ada proses yang berhubungan dengan pendaftaran.

Penelitian tentang Aplikasi Berbasis *Web* Tentang Profil SMAN 8 Yogyakarta yang dilakukan oleh Farid Prasetiawan (2011) dalam penelitian ini membahas tentang pembuatan aplikasi berbasis *web* sebagai media informasi bagi sekolah SMAN 8 Yogyakarta. **Kelebihan** : sebagai media informasi dan menyediakan data siswa dengan lebih cepat. **Kekurangan** : tidak ada proses yang berhubungan dengan pendaftaran.

Penelitian dengan judul Sistem Informasi Manajemen Pendaftaran *Online* Siswa Baru “SMP MUHAMMADIYAH 1 DEPOK” dengan PHP Dan MYSQL Berbasis *Website* yang dilakukan oleh Agung Nugroho dan Erna Cipta Ningsih (2010) dalam penelitian ini membahas tentang pembuatan *website* sebagai sebuah sarana pendaftaran siswa baru. **Kelebihan:** sebagai sarana pendaftaran bagi para calon siswa baru agar tidak perlu datang ke sekolah. **Kekurangan:** kurang interaktifnya *website* sebagai media informasi bagi masyarakat dan calon siswa.

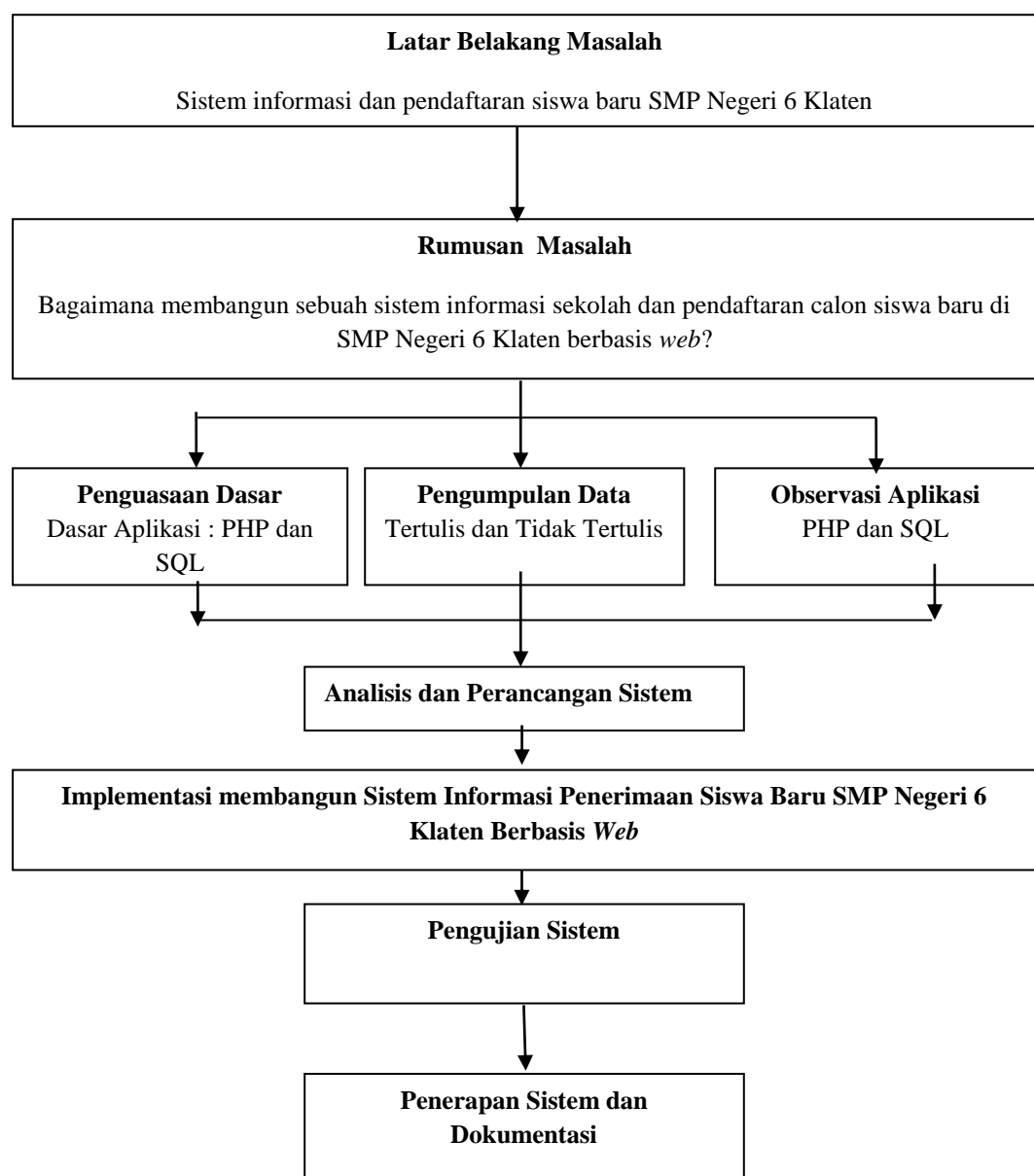
Penelitian dengan judul pengembangan sistem informasi pendaftaran siswa baru secara online berbasis *web* (Studi kasus SMK Pelopor Nasional Ciputat Tangerang Selatan) yang dilakukan oleh Catur Sudrajat (2011) dalam penelitian ini membahas tentang sebuah sistem informasi pendaftaran siswa baru yang dapat mempermudah proses pendaftaran dan juga membantu guru dalam melaksanakan proses penerimaan siswa baru. **Kelebihan:** sebagai sarana pendaftaran bagi para calon siswa baru agar tidak perlu datang ke sekolah dan mempermudah petugas pendaftaran. **Kekurangan:** kurang interaktifnya *website* sebagai media informasi bagi masyarakat dan calon siswa.

Penelitian dengan judul Sistem Informasi Penerimaan Siswa Baru Berbasis *Web* pada SMA NU Al Ma’ruf Kudus yang dilakukan oleh Adi Kurniawan (2014) dalam laporan penelitian ini membahas tentang merancang dan membangun suatu aplikasi Penerimaan Siswa Baru Berbasis *Web* pada SMA NU Al Ma’ruf Kudus dengan tujuan agar dapat memudahkan pihak panitia penerimaan siswa baru untuk melaksanakan pendaftaran calon siswa dan memudahkan calon siswa untuk mendaftar pada sekolah tersebut. **Kelebihan:** sebagai sarana pendaftaran bagi para calon siswa baru agar tidak perlu datang ke sekolah dan mempermudah petugas pendaftaran. **Kekurangan:** kurang interaktifnya *website* sebagai media informasi bagi masyarakat dan calon siswa.

Perbedaan penelitian yang penulis angkat dengan tinjauan pustaka yang sudah ada adalah menggunakan *web* sebagai media informasi sekolah dan sebagai proses pendaftaran bagi para calon siswa baru.

2.2. Kerangka Pemikiran

Penelitian ini menggunakan SMP Negeri 6 Klaten sebagai objek penelitian. SMP Negeri 6 Klaten merupakan salah satu SMP Negeri yang ada di daerah Klaten Jawa Tengah. Kerangka pemikiran dalam penelitian ini ditunjukkan dalam Gambar 2.1.



Gambar 2.1 Kerangka Berfikir

Gambar 2.1. merupakan diagram kerangka pemikiran yang dijalankan dalam penelitian ini. Keterangan dari Gambar 2.1 :

1. Latar belakang masalah

Tahapan paling awal, yakni menelusuri latar belakang kenapa masalah yang akan diangkat menjadi penting untuk dipilih.

2. Rumusan masalah

Penyimpulan latar belakang masalah menjadi suatu rumusan masalah yang akan diangkat untuk menjadi bahan penelitian.

3. Pengumpulan data tertulis dan tidak tertulis

Pengumpulan data dilakukan baik dengan tanya – jawab (*interview*), observasi, maupun studi *literatur* di perpustakaan dan internet.

4. Penguasaan dasar *PHP* dan *SQL*

Tahap untuk melakukan beberapa percobaan untuk membuat tampilan *database* sederhana untuk yang akan digunakan untuk membangun aplikasi.

5. Observasi aplikasi

Merupakan tahap pengamatan sampel – sampel aplikasi yang telah ada, jurnal, buku, maupun karya ilmiah untuk kajian yang dapat dijadikan referensi untuk pembangunan aplikasi.

6. Analisis dan perancangan aplikasi berbasis objek

7. Implementasi aplikasi Sistem Informasi Pendaftaran Siswa Baru SMP Negeri 6 Berbasis *Web*

8. Pengujian aplikasi

Pengujian sistem akan dilakukan untuk mengecek jika ada kesalahan dan kekurangan pada aplikasi.

9. Penerapan sistem dan Dokumentasi

Tahapan terakhir, yakni tahap pendokumentasian seluruh poses penyusunan tugas akhir ke dalam laporan

2.3. Landasan Teori Pendukung

2.3.1. Pengertian Sistem Informasi

Sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses atau pekerjaan tertentu (Fatansyah, 2001).

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran yang tertentu (Jogiyanto, 2001). Sistem adalah kumpulan komponen-komponen yang berhubungan dalam mencapai suatu tujuan. Informasi ialah hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagaimana dasar alam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang (Abdul Kadir, 2004). Informasi ialah rangkaian data yang mempunyai sifat sementara tergantung dengan waktu, mampu memberi kejutan pada yang menerimanya (Witarto, 2004).

Dari penjelasan di atas dapat disimpulkan bahwa informasi adalah kumpulan data yang telah di olah sehingga menjadi suatu bentuk yang berguna bagi penerimanya. Sistem informasi adalah suatu sistem didalam suatu organisasi yang memperlakukan 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, 2001).

Sistem informasi merupakan sistem, yang berisi jaringan sistem pengolahan data yang dilengkapi dengan kanal-kanal komunikasi yang digunakan dalam sistem organisasi data (Witarto, 2004).

2.3.2. Pengertian Basis data

Basis data atau *database* adalah kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan *hardware* komputer dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu (Sutanta, 2004).

Menurut Fathansyah (2007), basis data atau *database* terdiri atas dua kata, yaitu Basis dan Data. Basis dikatan sebagai markas atau gedung, tempat bersarang atau

berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek yang direkam dalam bentuk angka, simbol, huruf, teks, gambar, bunyi dan kombinasinya. Sebagai Basis Data dapat didefinisikan dalam sejumlah sudut pandang sebagai berikut:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa perulangan yang tidak perlu, untuk memenuhi berbagai kebutuhan
3. Kumpulan file, table, atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik


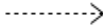

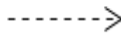


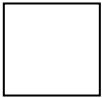



2.3.3. Analisis dan perancangan berorientasi objek

Pemrograman berorientasi objek adalah suatu cara baru dalam berpikir serta berlogika dalam menghadapi masalah-masalah yang akan dicoba diatasi dengan bantuan komputer. Tidak seperti pendahulunya (pemrograman terstruktur), pemrograman berorientasi objek mencoba melihat permasalahan lewat pengamatan dunia nyata dimana setiap objek adalah entitas tunggal yang memiliki kombinasi struktur data dan fungsi tertentu (Adi Nugroho, 2002:1). Dalam analisis dan perancangan metodologi berorientasi objek menggunakan diagram *UML (Unified Modeling Language)*. *UML* adalah keluarga notasi grafis yang didukung oleh meta model tunggal yang membantu pendeskripsian dan desain sistem yang dibangun menggunakan pemrograman berorientasi objek (Prabowo Pudjo Widodo dan Herlawati, 2011).



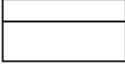




Simbol *UML* secara lengkap disajikan dalam Tabel 2.1

Tabel 2.1 Simbol *Use Case Diagram*

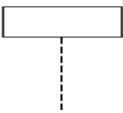
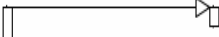
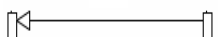
NO	GAMBAR	NAMA	KETERANGAN
----	--------	------	------------

1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi







Tabel 2.2 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2.3 Simbol *Sequence Diagram*






NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Tabel 2.4 Simbol *StateChart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2.5 Simbol *Actifty Diagram*

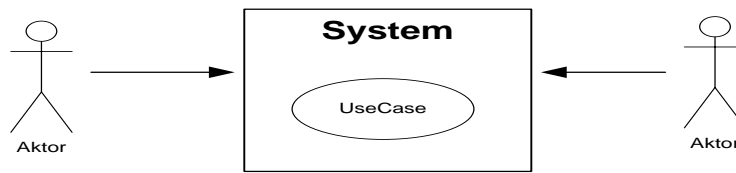
NO	GAMBAR	NAMA	KETERANGAN
----	--------	------	------------

1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

2.3.3.1. Use Case Diagram

Use case diagram adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan (Prabowo Pudjo Widodo dan Herlawati, 2011). *Use Case Diagram* dibuat untuk memvisualisasikan atau menggambarkan hubungan antara *Actor* dan *Use Case*. *Actor* mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberikan informasi pada sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki control atas *use case*. *Use case* adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

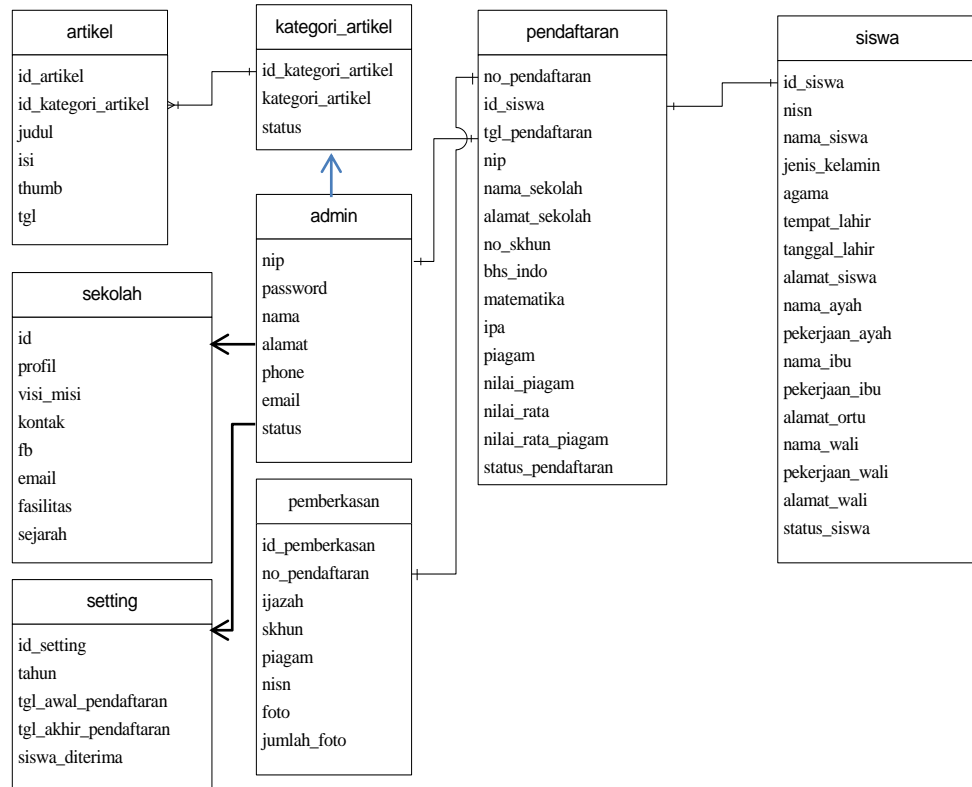
Use Case Model ditunjukkan Gambar 2.2



Gambar 2.2 Use Case Model

2.3.3.2. Class Diagram

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan property dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut (Prabowo Pudjo Widodo dan Herlawati. 2011). Class Diagram ditunjukkan Gambar 2.3

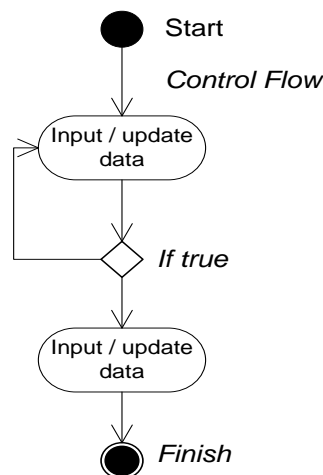


Gambar 2.3 Class Diagram

2.3.3.3. Activity Diagram

Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram ini

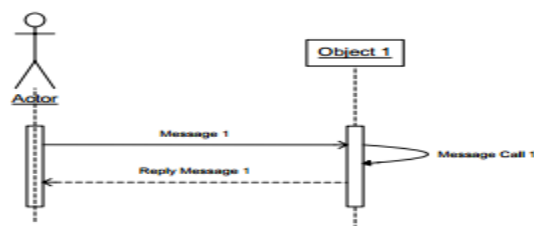
memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dan notasi diagram alir adalah diagram ini mendukung *behavior parallel* (Prabowo Pudjo Widodo dan Herlawati. 2011). *Activity Diagram* ditunjukkan Gambar 2.4



Gambar 2.4 *Activity Diagram*

2.3.3.4. *Statechart Diagram*

Statechart diagram adalah teknik umum yang digunakan untuk menggambarkan behavior sebuah sistem (Prabowo Pudjo Widodo dan Herlawati. 2011). *Statechart diagram* menggambarkan semua *state* atau kondisi yang dimiliki suatu object dari suatu class dan kejadian yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. *Statechart Diagram* ditunjukkan Gambar 2.5

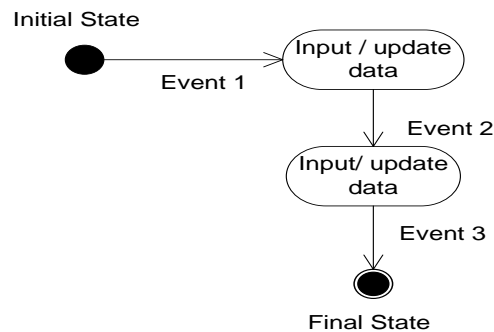


Gambar 2.5 *Statechart Diagram*

2.3.3.5. *Sequence Diagram*

Sequence diagram adalah penjabaran *behavior* sebuah skenario tunggal. *Sequence diagram* menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini didalam *use case* (Prabowo Pudjo Widodo dan

Herlawati. 2011). *Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence diagram* ditunjukkan Gambar 2.6



Gambar 2.6 *Sequence diagram*

2.3.3.6. *Component Diagram*

Component diagram merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada di analisis. Component terhubung melalui antarmuka yang digunakan dan dibutuhkan (Prabowo Pudjo Widodo dan Herlawati. 2011). *Component* dapat berupa *source code*, komponen biner, atau *executable component*. Sebuah component berisi informasi tentang *logic class* atau class yg diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*. Sehingga *component diagram* merepresentasikan dunia riil yaitu *component software* yang mengandung component, *interface*, dan *relationship*.

2.3.3.7. *Deployment Diagram*

Deployment diagram menunjukkan susunan fisik sebuah sistem, menunjukkan bagaimana perangkat lunak mana yang berjalan pada perangkat keras mana (Prabowo Pudjo Widodo dan Herlawati. 2011). *Deployment diagram* juga menggambarkan tata letak sebuah sistem secara fisik, menampilkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya

2.3.4. Perangkat Lunak (*Software*)

1. *Adobe Dreamweaver*

Adobe dreamweaver adalah program untuk mengedit HTML, secara visual dan mengelola halaman sebuah situs. Dreamweaver menyertakan banyak perangkat yang berkaitan dengan pengkodean seperti *HTML*, *CSS*, *Javascript*, *PHP*, *ASP*, *Cloudfusion*, dan *XML* (Ptihatna, 2005)

2. *Browser*

Browser merupakan program yang digunakan untuk menjalankan program yang telah jadi. Browser yang digunakan antara lain:

- a. Google chrome.
- b. Mozilla Firefox

3. *Apache Server*

Server HTTP Apache atau *Server Web/WWW Apache* dapat dijalankan di banyak sistem operasi (*Unix*, *BSD*, *Linux*, *Microsoft Windows* dan *Novell Netware* serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan *HTTP*. Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (*GUI*) yang memungkinkan penanganan server menjadi mudah. Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

Pada awal mulanya, Apache merupakan perangkat lunak sumber terbuka yang menjadi alternatif dari *server web Netscape* (sekarang dikenal sebagai *Sun Java System Web Server*). Sejak April 1996 Apache menjadi server web terpopuler di internet. Pada Mei 1999, Apache digunakan di 57% dari semua *web server* di dunia. Pada November 2005 persentase ini naik menjadi 71%. (*Netcraft Web Server Survey*, November 2005).

2.3.5. Bahasa Pemrograman

1. **PHP (Hypertext Preprocessor)**

PHP (Hypertext Preprocessor) merupakan bahasa pemrograman yang sebagian besar digunakan pada sisi server, yang dapat digunakan untuk menghasilkan *Hypertext Markup Language (HTML)* informasi secara dinamis. *PHP* terhubung ke *server web*, umumnya *Apache* atau *Internet Information Server (IIS)*, dan setelah selesai menghasilkan *HTML* yang tepat, ia akan mengirimkan penciptaan kembali ke server web untuk pengiriman ke klien meminta. (MacIntyre, 2012)

2. **MySQL**

MySQL merupakan *RDBMS (Relational Data Base Managemen System)*. *MySQL* didistribusikan secara *open source* dan gratis mulai tahun 1996, tetapi mempunyai sejarah pengembangan sejak tahun 1979.

SQL adalah bahasa yang fleksibel yang dapat Anda gunakan dalam berbagai cara. Ini alat yang paling banyak digunakan untuk berkomunikasi dengan *database* relasional. (Taylor, 2010). *Database MySQL* adalah *database* yang sangat powerfull, stabil, mudah. *MySQL* sangat banyak dipakai dalam sistem *database web* dengan menggunakan *PHP*. Seperti sistem *database SQL (Structured Query Language)* yang lain, *MySQL* juga dilengkapi dengan perintah-perintah dan sintaks-sintaks *SQL*, dengan keunggulan sebagai berikut:

- a. Konsep *database MySQL* kecepatan tinggi tentang sistem penyajian data
- b. Harga relatif murah, karena ada yang dapat diperoleh secara gratis
- c. Sintaks bahasanya menggunakan perintah sederhana
- d. Dapat bekerja dalam beberapa system operasi
- e. Dukungan penggunaan banyak tersedia (Swastika, 2006)

3. **HTML (HyperText Markup Language)**

HTML (HyperText Markup Language) adalah sekumpulan symbol-symbol atau *tag-tag* yang dituliskan dalam sebuah file yang dimaksudkan untuk menampilkan halaman pada *web browser*. *Tag-tag* tadi memberitahukan *browser* bagaimana menampilkan halaman *web* dengan lengkap kepada pengguna (Astamal, 2006). *HTML* diperlukan untuk menjelaskan struktur dari sebuah halaman *web*. Halaman ini terdiri dari banyak kata-kata. Pada halaman *web*, itu

adalah tugas *HTML* untuk menjelaskan struktur dari kata-mana kata membentuk pos, di mana paragraf awal dan akhir, dan mana teks harus memiliki peluru poin. Bahasa ini juga menentukan hal-hal seperti hubungan antara halaman web yang berbeda, di mana gambar akan muncul, di mana video akan muncul, dan bentuk untuk memasukkan teks (Larsen, 2013).

4. *CSS*

CSS digunakan untuk mengontrol bagaimana halaman Anda terlihat. Misalnya, Anda dapat menggunakan *CSS* untuk menentukan bahwa jenis huruf harus besar, tebal *arial typeface* atau latar belakang halaman harus hijau muda. *CSS* dapat digunakan untuk mengontrol mana item yang berbeda muncul pada halaman, seperti sebagai menempatkan tiga kolom teks samping satu sama lain (Larsen, 2013).

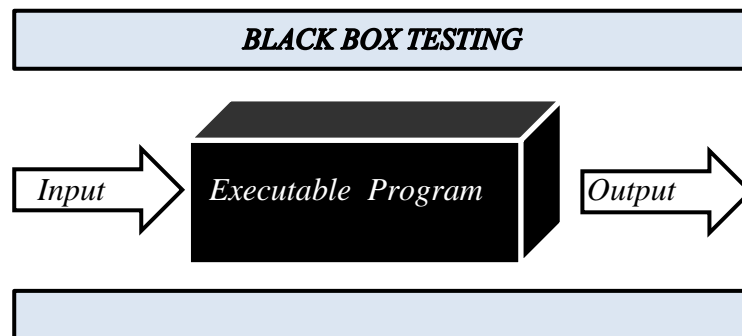
5. *Java Script*

JavaScript adalah bahasa penting karena merupakan bahasa *web browser*. *Java Script* dapat berasosiasi dengan browser dan menjadi salah satu bahasa pemrograman yang paling populer di dunia (Crockford, 2008).

2.3.6. Teori Pengujian sistem

Black-box testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program (Pressman, 2007).

Prosedur *Black Box Testing* ditunjukkan Gambar 2.7



Gambar 2.7 *Black Box Testing*

Kelebihan *Black Box Testing*

1. Dapat memilih *subset* test secara *efektif* dan *efisien*
2. Dapat menemukan cacat
3. Memaksimalkan testing *investmen*

Ciri-Ciri *Black Box Testing*

1. *Black box* testing berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
2. *Black box* testing bukan teknik alternatif daripada *white box* testing. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box* testing.
3. *Black box* testing melakukan pengujian tanpa pengetahuan *detil* struktur internal dari sistem atau komponen yang dites. juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*