

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

2.1.1. Perancangan

Perancangan merupakan sebuah proses untuk membuat dan menciptakan obyek baru (Srisulistiowati, 2017: 29).

Perancangan merupakan proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya (Dengen & Hatta, 2009: 48).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa perancangan adalah suatu proses untuk menciptakan objek baru dengan merealisasikan wujud fisiknya.

2.1.2. Sistem

Sistem merupakan sekelompok unsur yang saling berhubungan satu sama lain, untuk mencapai suatu tujuan tertentu (Zalina & Ahmadi, 2015: 12).

Sistem merupakan sekumpulan proses dan seperangkat elemen yang digabung serta dihimpun secara bersama, serta saling berintegrasi untuk mencapai suatu tujuan dari sistem organisasi (Warman & Saputra, 2012: 44).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa sistem adalah sekumpulan unsur berupa proses dan elemen yang saling berhubungan satu sama lain untuk mencapai tujuan.

2.1.3. Informasi

Informasi merupakan hasil pemrosesan data (fakta) menjadi sesuatu yang bermakna dan bernilai untuk pengambilan keputusan. Informasi tidak dapat terlepas dari aspek kehidupan manusia. Siapa, kapan, dan di manapun seseorang akan membutuhkan informasi (Zalina & Ahmadi, 2015: 12).

Informasi merupakan hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan (Husda, 2012: 117)

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa informasi adalah hasil pengolahan data yang menjadi suatu bentuk yang lebih berguna dan berarti bagi penggunanya untuk pengambilan keputusan.

2.1.4. Sistem Informasi

Sistem informasi merupakan kumpulan dari komponen-komponen dalam suatu perusahaan atau organisasi yang berhubungan dengan proses-proses penciptaan dan pengaliran informasi (Hapzi & Wangdra, 2010: 13).

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan-kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu (Sutabri, 2012: 46).

Sistem Informasi (SI) adalah kombinasi dari teknologi informasi dan aktivitas informasi untuk mendukung pengambilan keputusan dan kontrol dalam sebuah organisasi (Utomo, 2013: 18).

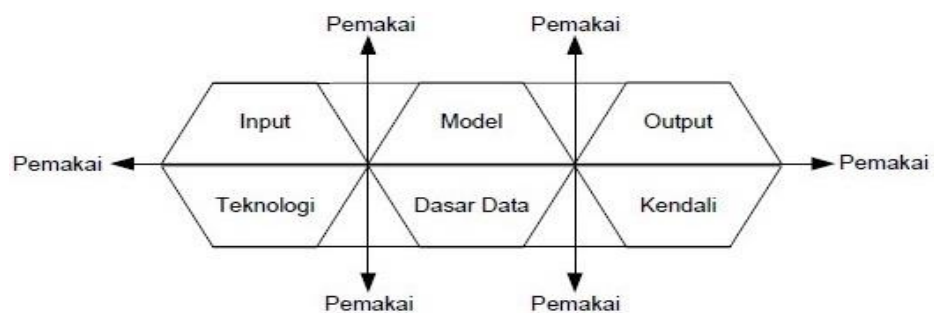
Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa sistem informasi adalah sekumpulan proses-proses dan komponen-komponen yang dapat mengelola dan menghasilkan suatu informasi dengan tujuan untuk mengambil keputusan dan kontrol dalam sebuah organisasi.

2.1.4.1. Komponen Sistem Informasi

Sistem informasi mempunyai enam buah komponen yang disebut dengan blok bangunan (*building block*). Keenam komponen ini harus membentuk satu kesatuan. Jika satu atau lebih komponen tersebut tidak ada, maka sistem informasi tidak akan berfungsi atau tidak dapat mencapai tujuannya, yaitu menghasilkan informasi yang akurat, tepat waktu dan relevan. Sistem informasi terdiri atas komponen-komponen antara lain (Husda, 2012: 120):

1. Blok Masukan (*Input Block*). Input merupakan data yang masuk ke dalam sistem informasi.

2. Blok Model (*Model Block*). Kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan dibasis data dengan untuk menghasilkan keluaran.
3. Blok Keluaran (*Output Block*). Keluran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem
4. Blok Teknologi (*Technology Block*). Teknologi merupakan alat (*tool box*) dalam sistem informasi yang digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran.
5. Blok Basis Data (*Database Block*). Basis data merupakan kumpulan dari data yang saling berhubungan satu sama lainnya yang tersimpan di perangkat keras komputer dan digunakan diperangkat lunak.
6. Blok Kendali (*Control Block*). Pengendalian dirancang secara khusus untuk menanggulangi gangguan-gangguan terhadap sistem.



Sumber: (Husda, 2012: 122)

Gambar 2.1. Komponen Sistem Informasi

2.1.5. SDLC (*System Development Life Cycle*)

SDLC atau *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (Rosa & Shalahuddin, 2011: 134). Berikut merupakan tahapan-tahapan dari SDLC (Rosa & Shalahuddin, 2011: 24):

1. Inisiasi (*initiation*)

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

2. Pengembangan konsep sistem (*system concept development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

3. Perencanaan (*planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.

4. Analisa kebutuhan (*requirements analysis*)

Menganalisa kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

5. Desain (*design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen *design* sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

6. Pengembangan (*development*)

Mengkonversi *design* ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan. Membuat basis data dan mempersiapkan prosedur kasus pengujian; mempersiapkan berkas atau *file* pengujian, pengodean, pengompilasian, memperbaiki program, peninjauan pengujian.

7. Integrasi dan pengujian (*intergration and test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang spesifikasi pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user*. Menghasilkan laporan analisis pengujian.

8. Implementasi (*implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

9. Operation dan pemeliharaan (*Operations and maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi, termasuk implementasi akhir dan masuk pada proses peninjauan.

10. Disposisi (*disposition*)

Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

SDLC memiliki beberapa model dalam penerapan tahapan prosesnya, berikut adalah model-model SDLC antara lain adalah (Rosa & Shalahuddin, 2011: 26):

1. Model *Waterfall*

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

2. Model *Prototype*

Model *prototype* dimulai dari pengumpulan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program *Prototype* agar pelanggan lebih terbayang dengan apa sebenarnya diinginkan. Program *Prototype* biasanya merupakan program *Prototype* yang belum jadi. Program ini biasanya merupakan program yang belum jadi.

3. Model *Rapid Application Development* (RAD)

Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari

model air terjun untuk mengembangkan setiap komponen perangkat lunak.

4. Model *Iteratif*

Model iteratif mengkombinasikan proses-proses pada model air terjun dan iteratif pada *prototype*. Model inkremental akan menghasilkan versi-versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya (inkremen/*increment*).

5. Model *Spiral*

Model spiral memasangkan iteratif dan model *prototype* dengan *control* dan aspek sistematis yang diambil dari model air terjun.

Dari kelima model SDLC yang sudah dijelaskan diatas maka penulis akan menggunakan model *waterfall* dikarenakan model yang digunakan mudah dan pengerjaannya masing-masing diselesaikan tahap demi tahap dengan urutan dari analisis, desain, *coding*, hingga *testing*.

2.1.6. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rosa & Shalahuddin, 2011: 118).

UML (*Unified Modeling Language*) adalah kumpulan notasi grafis yang didukung oleh sebuah meta model tunggal, yang membantu dalam menjelaskan dan merancang sistem perangkat lunak, khususnya sistem perangkat lunak dibangun menggunakan gaya berorientasi objek (Utomo, 2013: 19).

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak sebuah sistem (Anwar, *et al.*, 2016: 75).

Unified Modeling Language atau UML merupakan sebuah bahasa pemodelan yang telah distandarisasikan untuk merancang pemrograman berorientasi objek (Lee, 2012: 158).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa diagram UML (*Unified Modelling Language*) adalah bahasa visual untuk pemodelan dan komunikasi yang telah terstandarisasi yang digunakan untuk memberikan gambaran sistem berorientasi objek.

2.1.6.1. Bagian-Bagian UML

Terdapat beberapa bagian-bagian utama yang terdapat pada UML antara lain (Koespradono, *et al.*, 2013: 49):

1. *View*

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram.

2. *Use case view*

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa *user* atau sistem lainnya.

3. *Logical view*

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class*, *object*, dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu.

4. *Component view*

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi *administrative* lainnya.

5. *Concurrency view*

Membagi sistem ke dalam proses dan prosesor.

6. *Deployment view*

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya.

7. *Diagram*

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem.

2.1.6.2. **Diagram UML**

Penulis menggunakan empat jenis Diagram UML antara lain adalah *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram* untuk membantu dalam analisa dan desain, berikut penjelasannya:

1. Use Case Diagram

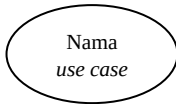
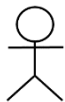
Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

Aktor: merupakan orang, proses, atau sistem lain berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

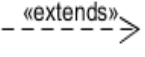
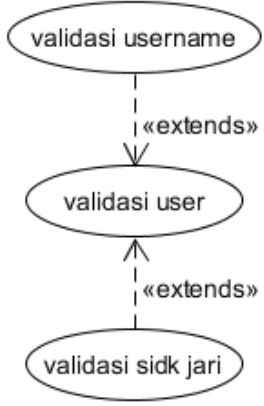
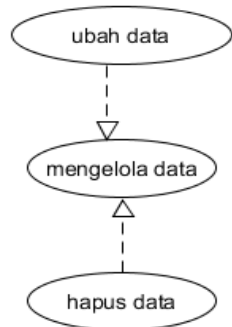
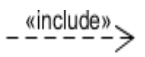
Use case: merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol dari *use case diagram*:

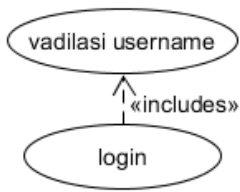
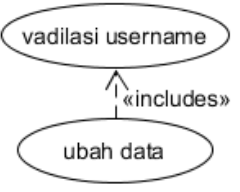
Tabel 2.1. Simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Use Case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i> .
	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
	Asosiasi/ <i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

Tabel Lanjutan 2.1. Simbol *use case diagram*

Simbol	Nama	Keterangan
	Ekstensi/ <i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan yaitu, mirip dengan prinsip <i>inheritance</i> pada pemrograman
		berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan sama dengan <i>use case</i> yang ditambahkan  Arah panah mengarah pada <i>use case</i> yang ditambahkan
	Generalisasi/ <i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :  Arah panah mengarah pada <i>use case</i> yang menjadi generalisasi (umum)
	Menggunakan/ <i>Include</i>	Relasi <i>use case</i> tambah 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.

Tabel Lanjutan 2.1. Simbol *use case diagram*

Simbol	Nama	Keterangan
		Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> yang ditambahkan dijalankan  <pre> graph BT login((login)) -- «includes» --> validasi_username((validasi username)) </pre> • <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan  <pre> graph BT ubah_data((ubah data)) -- «includes» --> validasi_username((validasi username)) </pre> Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan

Sumber: (Rosa & Shalahuddin, 2011: 131).

2. Activity Diagram

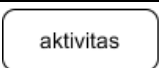
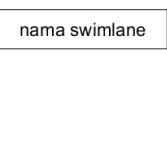
Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

2. Urutan atau pengelompokan tampilan dari *sistem/user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan khusus ujinya.

Berikut adalah simbol-simbol dari *Activity diagram*:

Tabel 2.2 Simbol *Activity Diagram*

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

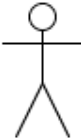
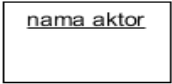
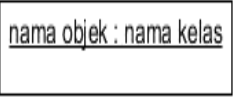
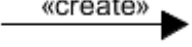
Sumber: (Rosa & Shalahuddin, 2011: 134).

3. *Sequence Diagram*

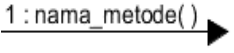
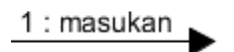
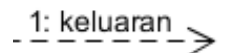
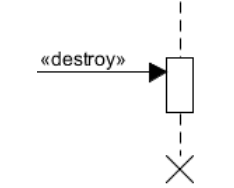
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case*

beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut adalah simbol-simbol dari *Sequence diagram*:

Tabel 2.3 Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
 Atau  Tanpa waktu aktif	Aktor	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; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
	Garis hidup/ <i>lifeline</i>	Menyatakan bahwa kehidupan suatu objek
	Objek	Menyatakan bahwa objek yang berinteraksi pesan
	Waktu Aktif	Menyatakan bahwa objek dalam keadaan aktif dan berinteraksi pesan
	Pesan tipe <i>create</i>	Menyatakan bahwa suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat

Tabel Lanjutan 2.3. Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Pesan tipe <i>call</i>	Menyatakan bahwa suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,
	Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
	Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
	Pesan tipe <i>destroy</i>	Menyatakan bahwa 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 & Shalahuddin, 2011: 138).

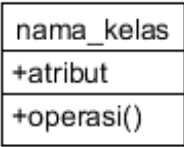
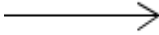
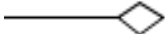
4. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas Memiliki atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain

sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.4 Simbol *Class Diagram*

Gambar	Nama	Keterangan
	Kelas	Kelas pada struktur sistem
	Antaramuka/ <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
	Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Asosiasi berarah/ <i>directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai <i>multiplicity</i>
	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
	Kebergantungan/ <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
	Agregasi/ <i>aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole part</i>)

Sumber: (Rosa & Shalahuddin, 2011: 123).

2.1.7. Basis Data (*Database*)

Basis Data (*Database*) adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan (Rosa & Shalahuddin, 2011: 44).

Database merupakan kumpulan data-data yang mempunyai fasilitas akses yang berhubungan satu sama lainnya sehingga dapat mengolah data-data tersebut sehingga membentuk suatu bangunan data (Warman & Saputra, 2012: 45).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa diagram *Database* atau Basis Data merupakan kumpulan data-data yang saling berhubungan satu sama lain dimana data tersebut diakses dikelola melalui melalui sistem komputer.

2.2. Tinjauan Teori Khusus

2.2.1. Website

Website adalah sebuah tempat di internet, yang menyajikan informasi seperti *text*, *image*, bahkan video dan dapat diakses menggunakan berbagai aplikasi *client* sehingga memungkinkan penyajian informasi yang lebih menarik dan dinamis dengan pengelolaan yang terorganisasi (Sumarlinda, 2015: 19).

Website adalah ruang informasi didalam ruang internet yang menggunakan teknologi *hypertext*. Informasinya dapat berupa teks, gambar, suara, animasi, dan data multimedia lainnya (Kiswanto, *et al.*, 2016: 69).

Website adalah kumpulan halaman-halaman yang di gunakan untuk mempublikasikan informasi berupa teks, gambar program multimedia lainnya berupa animasi gambar bergerak, tulisan bergerak), suara atau gabungan dari semua itu baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait antara satu *page* dengan *page* lain sering di sebut sebagai *hyperlink* (Rompis, *et al.*, 2015: 52).

Website merupakan kumpulan halaman yang menampilkan informasi data, teks data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*) (Pangerapan, 2013: 764).

Berdasarkan beberapa pendapat yang diatas, maka menurut peneliti bahwa *website* adalah sekumpulan halaman-halaman yang menyajikan informasi berupa teks, gambar, video, animasi, suara, dan lainnya yang dapat diakses diberbagai *client* yang menggunakan teknologi *hyperlink*.

2.2.1.1. Unsur-Unsur Website

Sebuah *website*, harus tersedia unsur-unsur penunjangnya, antara lain sebagai berikut (Pangerapan, 2013: 764):

1. Nama Domain (*Domain name/URL – Uniform Resource Locator*)

Pengertian Nama domain atau biasa disebut dengan *Domain Name* atau *URL* adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah *website*, atau dengan kata lain *domain name*

adalah alamat yang digunakan untuk menemukan sebuah *website* pada dunia internet. Contoh <http://www.detik.com>, www.unsrat.ac.id. Nama domain diperjualbelikan secara bebas di internet dengan status sewa tahunan.

2. Rumah Tempat *Website* (*Web hosting*)

Pengertian *Web Hosting* dapat diartikan sebagai ruangan yang terdapat dalam *harddisk* tempat menyimpan berbagai data, *file-file*, gambar dan lain sebagainya yang akan ditampilkan di *website*. Besarnya data yang bisa dimasukkan tergantung dari besarnya *web hosting* yang disewa atau dipunyai, semakin besar *web hosting* semakin besar pula data yang dapat ditampilkan pada *website*. *Web Hosting* juga diperoleh dengan menyewa. Besarnya *hosting* ditentukan ruangan *harddisk* dengan ukuran MB (*Mega Byte*) atau GB (*Giga Byte*). Lama penyewaan *web hosting* rata-rata dihitung per tahun. Penyewaan *hosting* dilakukan dari perusahaan-perusahaan penyewa *web hosting* yang banyak dijumpai baik di Indonesia maupun luar negeri.

3. Bahasa Program (*Scripts Program*)

Bahasa yang digunakan untuk menerjemahkan setiap perintah dalam *website*, pada saat diakses. Jenis bahasa program sangat menentukan statis, dinamis atau interaktifnya sebuah *website*. Semakin banyak ragam bahasa program yang digunakan maka akan terlihat *website* semakin dinamis dan interaktif. Bahasa program ini biasanya digunakan untuk membangun portal berita, artikel, forum diskusi, buku tamu, anggota

organisasi, *email*, *mailing list* dan lain sebagainya yang memerlukan *update* setiap saat.

4. Desain Website

Setelah melakukan penyewaan *domain name* dan *web hosting* serta penguasaan bahasa program (*scripts program*), unsur *website* yang penting dan utama adalah desain. Desain *website* menentukan kualitas dan keindahan sebuah *website*. *Website* juga butuh designer dan perlu diketahui bahwa kualitas situs sangat ditentukan oleh kualitas *designer*. Semakin banyak penguasaan *web designer* tentang beragam program atau *software* pendukung pembuatan situs maka akan dihasilkan situs yang semakin berkualitas, demikian pula sebaliknya. Jasa *web designer* ini yang umumnya memerlukan biaya yang tertinggi dari seluruh biaya pembangunan situs dan semuanya itu tergantung kualitas *designer*.

5. Publikasi Website

Keberadaan situs tidak ada gunanya dibangun tanpa dikunjungi atau dikenal oleh masyarakat atau pengunjung internet. Karena efektif tidaknya situs sangat tergantung dari besarnya pengunjung dan komentar yang masuk. Cara yang biasanya dilakukan dan paling efektif dengan tak terbatas ruang atau waktu adalah publikasi langsung di internet melalui *search engine* (mesin pencari, seperti: Yahoo, dan Google. Cara publikasi di *search engine* ada yang gratis dan ada pula yang membayar. Yang gratis biasanya terbatas dan cukup lama untuk bisa masuk dan dikenali di *search engine* terkenal seperti Yahoo atau Google. Cara

efektif publikasi adalah dengan membayar, walaupun harus sedikit mengeluarkan akan tetapi situs cepat masuk ke *search engine* dan dikenal oleh pengunjung.

6. Pemeliharaan *Website*

Tanpa pemeliharaan yang baik situs akan terkesan membosankan atau monoton juga akan segera ditinggal pengunjung. Pemeliharaan situs dapat dilakukan per periode tertentu seperti tiap hari, tiap minggu atau tiap bulan sekali secara rutin atau secara periodik saja tergantung kebutuhan (tidak rutin). Setiap pengeluaran internal pada pengembangan dan pengoperasian dari *website* perusahaan akan dicatat sesuai dengan PSAK 19 (revisi 2009): Aset Tidak Berwujud. Sifat dari masing-masing aktivitas atas pengeluaran yang terjadi (pelatihan dan pemeliharaan *website*) dan tahapan pengembangan atau pasca pengembangan *website* harus dievaluasi untuk menentukan perlakuan akuntansi yang sesuai.

2.2.1.2. Jenis-Jenis *Website*

Dalam pengelompokan jenis *web*, pengelompokan dibagi menjadi tiga jenis yaitu *website* berdasarkan sifat, fungsi dan bahasa pemrograman. Adapun jenis-jenis *web* antara lain (Hidayat, 2010: 3):

1. Jenis-jenis *website* berdasarkan sifat
 - b. *Website* dinamis, merupakan *website* yang menyediakan konten yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang

digunakan antara lain PHP, ASP, .NET dan memanfaatkan mySQL atau MS SQL.

- c. *Website statis*, merupakan *website* yang kontennya sangat jarang berubah. Bahasa pemrograman yang digunakan adalah HTML dan belum memanfaatkan *database*.

2. Jenis-jenis *website* berdasarkan fungsinya

- a. *Personal website*, *website* yang berisi informasi pribadi seseorang.
- b. *Commercial website*, *website* yang dimiliki oleh perusahaan yang bersifat bisnis.
- c. *Goverment website*, *website* yang dimiliki oleh instansi pemerintah, pendidikan, yang bertujuan untuk memberikan layanan kepada pengguna.
- d. *Non-profit organization website*, *website* yang dimiliki oleh organisasi yang bersifat non *profit* atau tidak bersifat bisnis.

3. Jenis-jenis *website* berdasarkan bahasa pemrograman

- a. *Server side*, merupakan *website* yang pemrograman yang tergantung kepada tersedianya *server*. Seperti PHP, ASP dan lain sebagainya. Jika tidak ada *server* maka *website* yang dibangun menggunakan bahasa pemrograman diatas tidak akan berfungsi sebagaimana mestinya.
- b. *Client side*, merupakan *website* yang tidak membutuhkan *server* dalam menjalankannya, cukup diakses melalui *browser* saja. Misalnya HTML.

2.2.1.3. Kriteria *Website* yang Baik

Situs *web* yang baik adalah situs *web* yang mampu mewujudkan tujuan kita semula, yaitu menjual. Lalu bagaimanakah kriteria situs *web* yang baik, yaitu antara lain (Dianawati, 2007: 38):

1. Pemilihan nama

Sebelumnya tentukan terlebih dahulu tipe situ yang ingin kita buat dan carikan nama yang sesuai. Sebaiknya nama tersebut harus mampu “menjual” mudah diingat dan relevan dengan isi situs. Situs *web* yang baik adalah situs *web* yang mempunyai nilai jual yang tinggi, artinya *web* tersebut mampu mewakili apa yang menjadi tujuan utama kita.

2. Beli nama *domain* dan *hosting*

Kita perlu membeli nama *domain* yang sesuai dengan nama situs. Bagian ini bisa membuat kita frustrasi karena biasanya nama yang kita pilih sudah digunakan oleh orang lain. Cari beberapa nama cadangan atau pindah ke *domain* lainnya, seperti .erg, .net, dan lain sebagainya sesuai dengan bisnis yang akan kita jalani. Meskipun mungkin ada yang lebih murah (bahkan gratis) untuk layanan *hosting*, sebaiknya belilah *hosting* dari perusahaan yang sama ketika membeli *domain*.

3. Isi

Keahlian menulis dibutuhkan untuk membuat isi. Jika diperlukan tambahan gambar-gambar pendukung yang relevan. Aturlah semenarik mungkin semua informasi yang ingin disampaikan kepada pengunjung. Apakah kita akan menampilkan *company profile*, berita, atau gambar?

4. Struktur menu

Perancangan menu-menu ini dilakukan setelah kita mengetahui apa yang akan ditampilkan. Menu-menu inilah yang menjadi *navigator* untuk memudahkan pengunjung menemukan informasi yang mereka perlukan.

5. Desain dan layout situs *web* semenarik mungkin

Meskipun konten/isi merupakan faktor terpenting dalam keberhasilan situs *web*, tapi kita harus mampu memikat pengunjung. Jika tampilan situs kita menarik, tentu saja akan membuat mereka betah berlama-lama disana. Pemilihan warna juga perlu diperhatikan, buatlah Pemilihan warna juga perlu diperhatikan, buatlah *eye catching* dimata pengunjung. Perhatikan juga jika ingin menggunakan flash, sebaiknya dijaga agar tidak terlalu ramai. Hal lainnya yang juga tak kalah penting adalah *layout*, dimana akan menentukan bagaimana konten/isi *web* tersebut dapat diakses keluar.

6. SEO

SEO (*Search Engine Optimization*) inilah yang membuat situs kita dikenal oleh mesin pencari. Tujuannya adalah agar situs *web* kita menduduki posisi teratas dihalaman pertama melalui pencariin kata kunci.

7. Test situs

Periksalah pesan-pesan *error* yang biasanya muncul karena kesalahan pemrograman (*bug*) disitus. Perbaikilah kesalahan-kesalahan itu agar situs kita tidak kehilangan kepercayaan dari awal.

8. Tambahan *traffic analysis*

Dengan penambahan ini, maka kita bisa mengetahui dari mana pengunjung berasal, dimana mereka tinggal, berapa lama mereka berada disitus kita, informasi apa saja yang mereka cari dan siapa yang berbicara dengan siapa (berkala atau tidak). Tentu saja informasi-informasi tadi sangat penting ketika akan memutuskan menjual suatu produk. Selain itu juga sebagai bukti ke pemasang iklan tentang *traffic* situs kita.

9. Daftarkan ke mesin pencari, iklan, *link exchange*

Mendaftarkan situs ke mesin pencari akan membantu pengunjung menemukan *web* kita. Iklankan situs ke mesin pencari seperti Google karena hal ini akan membantu kita secara mendapatkan *traffic*. Untuk memulainya, coba masuk ke www.google.com/ad-words. lakukan tukar-menukar *link* dengan situs lain, tapi sebaiknya hati-hati dengan situs-situs porno, judi, dan sejenisnya.

10. Jangan pernah berhenti

Ingatlah bahwa pembuatan situs tersebut tidak akan pernah selesai. Jadi jangan pernah berhenti atau bosan melakukan perbaikan, buatlah mereka (pengunjung) memiliki pengalaman yang unik disana. Perhatikan pula *feedback* yang kita terima dari mereka dan tambahkan isi yang relevan.

2.2.2. Tes

Tes merupakan suatu pengukuran yang objektif dan standar terhadap sampel perilaku (Fitrianingsih, *et al.*, 2015: 106).

Tes merupakan alat pengukur yang mempunyai standar objektif sehingga dapat dipergunakan untuk mengukur dan membandingkan keadaan psikis atau tingkah laku individu. (Matondang, 2009: 88).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa Tes adalah pengukuran dengan standar objektif yang digunakan untuk mengukur perilaku individu.

2.2.3. Potensi

Potensi adalah kemampuan dan kekuatan yang dimiliki oleh seseorang baik fisik maupun mental dan mempunyai kemungkinan untuk dikembangkan bila dilatih dan ditunjang dengan sarana yang baik (Aprillina, *et al.*, 2014: 44).

Potensi adalah kemampuan-kemampuan dan kualitas-kualitas yang dimiliki oleh seseorang namun belum dipergunakan secara maksimal (Sunarno, 2008: 8).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa potensi adalah kemampuan yang terdapat pada diri seseorang secara fisik maupun mental namun kemampuan tersebut belum dipergunakan sepenuhnya.

2.2.4. Akademik

Akademik merupakan kegiatan yang dilakukan didalam lingkungan dunia pendidikan yang berhubungan dengan proses belajar mengajar (Setiyawan, *et al.*, 2014: 2) .

Akademik merupakan proses belajar mengajar yang dilakukan di kelas atau dunia persekolahan. Kegiatan akademik meliputi tugas-tugas yang dinyatakan dalam program pembelajaran, diskusi, observasi, dan pengerjaan tugas (Membara, *et al.*, 2014: 73).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa Akademik adalah kegiatan proses belajar mengajar didalam lingkungan pendidikannya.

2.2.5. Tes Potensi Akademik

Tes Potensi Akademik adalah suatu tes yang diperuntukkan mengukur kemungkinan keberhasilan seseorang apabila yang bersangkutan melanjutkan ke dunia akademik atau memegang jabatan/golongan dimana jabatan/golongan tersebut membutuhkan kemampuan akademis (Kiswanto, *et al.*, 2016: 69).

Tes Potensi Akademik adalah sebuah tes yang bertujuan untuk mengukur kemampuan seseorang dibidang *akademik* umum yang diidentikkan sebagai tes kecerdasan seseorang (Sugiri & Ramdhani, 2015: 85).

Berdasarkan beberapa pendapat yang dikemukakan diatas, maka dapat disimpulkan bahwa Tes Potensi Akademik merupakan tes yang bertujuan untuk mengukur tingkat kemampuan seseorang secara akademik dimana penilaiannya bertujuan untuk melanjutkan pendidikan akademik atau memegang jabatan pada dunia karir.

2.2.5.1. Jenis Tes Potensi Akademik

Pada umumnya, Tes Potensi Akademik memiliki tiga jenis, yaitu tes *verbal* (bahasa), tes bilangan (angka) dan, tes gambar (logika) (Kiswanto, *et al.*, 2016: 68):

1. Tes Verbal (Bahasa)

Melalui tes verbal, kemampuan berbahasa anda akan diuji. Semakin tinggi kemampuan berbahasa anda, semakin mudah memahami objek yang dibaca. Kemudian kemampuan berbahasa dapat ditingkatkan dengan banyak latihan (Sulistiyowati, 2013: 1).

2. Tes Bilangan (Angka)

Tes Bilangan menyajikan serangkaian tes yang dapat digunakan untuk mengukur kemampuan seseorang dalam memecahkan persoalan serta kemampuan berpikir cepat, tepat, dan benar dalam menghadapi dan menyelesaikan persoalan, terutama dalam hal susunan huruf, angka, maupun gambar (Sulistiyowati, 2013: 31).

3. Tes Gambar (Logika)

Tes Gambar akan mengukur kemampuan berpikir logis, cepat, dan tepat dalam menghadapi permasalahan, khususnya yang berkaitan dengan bentuk gambar atau simbol. Soal-soalnya sangat beragam dan variatif berupa deretan gambar (Sulistiyowati, 2013: 87).

2.2.6. Penskoran Tes dan Konversi Skor

Penskoran tes berbentuk pilihan ganda ada tiga macam, yaitu: penskoran tanpa ada koreksi jawaban, penskoran ada koreksi jawaban, dan penskoran dengan

butir beda bobot (Arifin, 2012: 277). Namun penulis hanya menggunakan teknik penskoran tanpa koreksi jawaban untuk perancangan sistem Tes Potensi Akademik .

Teknik penskoran tanpa koreksi jawaban yaitu penskoran dengan cara setiap butir soal yang dijawab benar mendapat nilai satu (tergantung dari bobot butir soal) (Arifin, 2012: 277). Skor peserta didik diperoleh dengan cara menghitung banyaknya butir soal yang dijawab benar. Berikut merupakan rumus dari Teknik penskroan tanpa koreksi. Penskoran tanpa koreksi saat ini banyak digunakan dalam penilaian pembelajaran (Sumaryanta, 2015: 182). Berikut merupakan rumus dari teknik penskoran tanpa koreksi:

Rumus 2.1. Rumus Teknik Penskoran Tanpa Koreksi

$$S = \frac{B}{N} \times 100$$

Keterangan:

S = Hasil skor

B = Jumlah jawaban benar

N = Jumlah keseluruhan soal

Untuk konversi nilai, penulis memilih konversi dengan skala lima (huruf). Skala lima (huruf) merupakan skala penilaian hasil belajar yang dinyatakan dengan huruf/nilai lambang yang terdiri dari A (sangat baik), B (baik), C (cukup), D (kurang), E (buruk), yang masing-masing bernilai 4, 3, 2, 1 dan 0 dimana penilaian ini bersumber penjumlahan dari nilai tes kuis, nilai tugas, nilai kehadiran dan nilai ujian akhir semester sesuai pedoman (Nurbayani, 2012: 33).

Penulis memilih teknik penskoran tanpa koreksi jawaban dan teknik konversi nilai skala lima (huruf) pada sistem tes potensi akademik. dikarenakan kedua teknik tersebut sangat mudah untuk digunakan dan mudah dipahami oleh banyak orang sehingga pengguna sistem tidak perlu kesulitan untuk mengetahui hasil nilai tes potensi akademik yang keluar saat akhir sesi tes.

2.2.7. HTML (*Hypertext Markup Language*)

HTML (*Hypertext Markup language*) adalah bahasa yang menentukan bagaimana halaman yang mengandung teks, gambar, dan informasi lainnya diatur dengan baik dan berkaitan satu sama lain pada suatu *web* (Solusindo, 2008: 5).

HTML (*Hypertext Markup language*) adalah bahasa pemrograman yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *website* (Saputra, 2012: 1).

HTML adalah *script* dimana kita bisa menampilkan informasi dan daya kreasi kita lewat internet (Fitrianingsih, *et al.*, 2015: 106).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa HTML (*Hypertext Markup Language*) adalah bahasa pemrograman *web* dasar yang bersifat *client side* yang digunakan untuk menampilkan teks, gambar, dan informasi lainnya.

Penulis menggunakan bahasa *web* HTML dikarenakan HTML merupakan bahasa dasar *web* yang paling umum digunakan untuk merancang konten-konten yang ada pada *web* Tes Potensi Akademik. Selain merupakan bahasa yang

diutamakan, penulis memilih HTML dikarenakan bahasa HTML tergolong mudah untuk merancang konten-konten yang ada pada halaman *web*.

2.2.8. CSS (*Cascading Style Sheet*)

CSS (*Cascading Style Sheet*) merupakan bahasa pemrograman *web* yang didesain khusus untuk mengendalikan dan membangun berbagai komponen dalam *web* sehingga tampilan *web* lebih rapih, terstruktur dan seragam (Saputra, 2012: 27).

CSS merupakan bahasa pemrograman *web* yang digunakan untuk mengatur *style-style* yang ada di *tag-tag* HTML (Prayitno, *et al.*, 2015: 2).

Cascading Style Sheet (CSS) adalah salah satu bahasa untuk mengatur tampilan dalam *web* yang berfungsi memisahkan antara desain dengan *content* yang dapat ditulis untuk mendefinisikan ulang *tag-tag* HTML ataupun dengan menambah atribut *id* dan *class* pada *tag* HTML (Rahmad, *et al.*, 2013: 61).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa CSS (*Cascading Style Sheet*) adalah bahasa pemrograman *web* yang digunakan untuk mengatur desain tampilan antarmuka *web* yang ditulis dengan *tag-tag* HTML dengan atribut *id* dan *class* pada *tag* HTML.

Penulis menggunakan bahasa *web* CSS dikarenakan CSS merupakan bahasa dasar *web* yang sifatnya wajib untuk membuat desain antarmuka pada halaman *web* Tes Potensi Akademik.

2.2.9. PHP (*Personal Home Page*)

PHP (*Personal Home Page*) adalah bahasa pemrograman di sisi *server* yang lazim ditulis dalam konteks dokumen HTML (Solusindo, 2008: 7).

PHP (PHP, *Hypertext Processor*) adalah bahasa skrip yang dapat ditanamkan atau disisipkan kedalam HTML yang banyak dipakai untuk memprogram situs *web* dinamis (Anisya, 2013: 51).

PHP adalah sebuah bahasa pemrograman *web* berbasis *server* yang mampu memarsing kode PHP dari kode *web* dengan ekstensi .php, sehingga menghasilkan tampilan *website* yang dinamis disisi *client* (Kiswanto, *et al.*, 2016: 69).

PHP adalah singkatan dari *Personal Home Page* yang merupakan bahasa standar yang digunakan dalam dunia *website* yang berbentuk *script* yang diletakkan didalam *web server* (Fitrianingsih, *et al.*, 2015: 106).

PHP merupakan bahasa pemrograman pada sisi *server* yang memperbolehkan *programmer* menyisipkan perintah-perintah perangkat lunak *web server* (Apache, IIS, atau apapun) akan dieksekusi sebelum perintah itu dikirim oleh halaman ke *browser* yang me-request-nya (Februariyanti, *et al.*, 2012: 128).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa PHP (*Personal Home Page*) adalah bahasa pemrograman *web* yang bersifat *server side* dimana perintah pada bahasa PHP tersebut digunakan untuk menampilkan data yang terdapat pada perangkat lunak *web server*.

Dengan menggunakan bahasa *web* PHP, penulis dapat menghubungkan rancangan pada skrip pengkodean HTML ke tabel *database* sehingga data pada halaman *web* Tes Potensi Akademik dapat disimpan dan dikelola oleh *database*.

2.2.10. JavaScript

JavaScript merupakan *scripting language* yang terintegrasi dengan *web browser* untuk memberikan fleksibilitas tambahan lagi *programmer* untuk mengontrol elemen-elemen dalam halaman *web* (Fitrianingsih, *et al.*, 2015: 106).

JavaScript adalah bahasa *scripting* kecil, ringan, berorientasi objek yang ditempelkan pada kode HTML dan di proses di sisi *client*. JavaScript digunakan dalam pembuatan *website* agar lebih interaktif dengan memberikan kemampuan tambahan terhadap fitur HTML melalui eksekusi perintah di sisi *browser* (Yanti, 2014: 2).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa JavaScript adalah bahasa pemrograman *web* berorientasi objek sisi *client* sebagai tambahan dari HTML yang sehingga dapat menampilkan dokumen HTML dengan tampilan yang bersifat interaktif.

Penulis memilih JavaScript sebagai skrip tambahan dikarenakan keterbatasan fitur fungsi pada HTML. Sehingga penulis dapat menggunakan JavaScript sebagai skrip pelengkap pada sistem *web* yang dibuat oleh penulis.

2.2.11. AJAX (*Asynchronous JavaScript and XML*)

Asynchronous JavaScript and XML HTTP atau disingkat AJAX adalah suatu teknik pemrograman berbasis *web* interaktif yang tujuannya untuk pertukaran data dengan *server* sehingga halaman *web* tidak harus dibaca ulang secara keseluruhan setiap kali pengguna melakukan perubahan (Winarto, *et al.*, 2011: 147).

Asynchronous JavaScript And XML (AJAX) adalah teknik untuk membuat aplikasi *web* interaktif yang tujuannya adalah untuk memindahkan sebagian besar interaksi pada komputer dan melakukan pertukaran data dengan *server* di belakang layar, sehingga halaman *web* tidak harus dibaca ulang secara keseluruhan setiap kali seorang pengguna melakukan perubahan (Fitriya, *et al.*, 2015: 368).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa AJAX merupakan teknik pemrograman *web* dimana tujuannya adalah untuk melakukan pertukaran data antara halaman *web* dan *server* sehingga halaman *web* tidak harus dibaca ulang secara keseluruhan setiap kali pengguna melakukan perubahan data.

Penulis memilih AJAX dikarenakan AJAX merupakan bahasa dasar dan penting pada JavaScript yang digunakan untuk melakukan pertukaran data pada konten JavaScript ke *server database*.

2.2.12. jQuery

jQuery adalah *library* JavaScript yang memungkinkan kita untuk membuat program *web* pada suatu halaman *web*, tanpa harus secara eksplisit kita menambahkan *event* atau properti pada halaman *web* tersebut (Koespradono, *et al.*, 2013: 49).

jQuery adalah JavaScript *library* yang dirancang untuk meringkas kode-kode JavaScript, sehingga dapat menyederhanakan penulisan skrip program, sesuai dengan slogan “*write less, do more*” (Yanti, 2014: 2).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa jQuery merupakan *library* Javascript yang berupa hasil ringkasan dari kode JavaScript sehingga mempermudah dalam pembuatan *web* dengan JavaScript.

Penulis memilih jQuery dikarenakan jQuery dapat mempermudah penulis dalam menggunakan kode JavaScript dan AJAX untuk perancangan sistem tes potensi akademik berbasis *web* sehingga penulis tidak perlu mengkhawatirkan kesulitan saat pengkodean dengan JavaScript dan AJAX.

2.2.13. MySQL

MySQL merupakan suatu perangkat lunak basis data (*database*) relasi berupa RDBMS (*Relational Database management System*) seperti halnya ORACLE, POSTGRESQL, MSSQL, dan sebagainya (Anisya, 2013: 51).

MySQL merupakan salah satu *database* kelas dunia yang dipadukan pada pemrograman PHP yang bekerja menggunakan bahasa SQL (*Structure Query Language*) (Saputra, 2012: 77).

MySQL merupakan *software* yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *open source* (Kiswanto, *et al.*, 2016: 70).

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread* dan *multiuser* (Fitrianingsih, *et al.*, 2015: 106).

MySQL merupakan sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*) (Februariyanti & Zuliarso, 2012: 128).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa MySQL adalah sebuah *software* yang berupa RDBMS (*Relational Database Management System*) yang menggunakan bahasa SQL (*Structure Query Language*) dan bersifat *open source*.

Penulis menggunakan MySQL karena mudah untuk digunakan untuk merancang *database* pada *web*. Selain mudah MySQL bersifat *open source* sehingga penulis tidak perlu mengeluarkan biaya tambahan untuk perancangan sistem *web* Tes Potensi Akademik.

2.2.13.1. Perintah-Perintah MySQL

Query dikirimkan ke *database* dalam bentuk SQL *Query*. Beberapa perintah umum digunakan adalah sebagai berikut (Anisya, 2013: 51):

1. CREATE : Untuk membuat tabel baru.

Create Table <Nama Table> (<Nama Kolom> <Tipe>, <Nama Kolom>
<Tipe>, Primary Key (<Nama Kolom>) Foreign Key (<Nama
Kolom>) References <Nama_Table> (<Nama Kolom>)

2. SELECT : Untuk mengambil *record* dari *database* yang memenuhi kriteria tertentu.

Select <Nama Kolom>, <Nama Kolom>, . . . From <Nama Tabel>
Where <Kondisi>

3. INSERT : Untuk menambah *record* ke dalam suatu tabel .

Insert Into <Nama Tabel> (<Nama Kolom>, <Nama Kolom>) Values
(<Nilai Kolom>, <Nilai Kolom>, . . .)

4. UPDATE : Untuk merubah isi *record* tertentu pada suatu tabel.

Update <Nama Tabel> Set (<Nama Kolom> = <Nilai Kolom>, <Nama Kolom> = <Nilai Kolom>, . . .) Where <Kondisi>

5. DELETE : Untuk menghapus *record* pada suatu tabel.

Delete From <Nama Tabel> Where <Kondisi>

6. DROP : Untuk menghapus sebuah tabel.

Drop <Nama Tabel>

2.2.14. **Black Box Testing**

Pengujian *Black Box Testing* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Tujuan dari metode *Black Box Testing* adalah untuk menemukan kesalahan fungsionalitas pada aplikasi. Pengujian dinyatakan berhasil jika aplikasi mampu menjalankan fungsi-fungsi berdasarkan spesifikasi secara benar. Sebaliknya pengujian dinyatakan gagal jika terdapat fungsi dari perangkat lunak dalam spesifikasi yang tidak dapat dijalankan pada proses pengujian (Sumarlinda, 2015: 27).

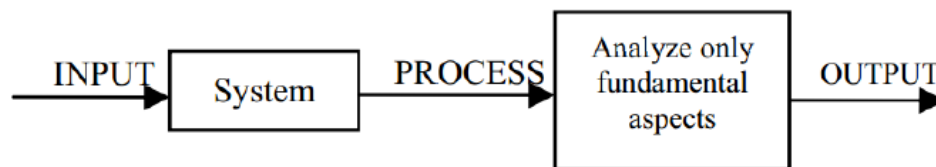
Black Box Testing merupakan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Rouf, 2012: 3).

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka menurut peneliti bahwa *Black Box Testing* merupakan teknik pengujian perangkat lunak yang bertujuan untuk menguji apakah fungsi pada perangkat lunak telah berfungsi semestinya sesuai dengan kebutuhan fungsional.

Black Box dirancang untuk dapat menemukan kesalahan pada perangkat lunak berupa (Rouf, 2012: 4):

Fungsi-fungsi yang tidak benar atau hilang

1. Kesalahan *interface*.
2. Kesalahan dalam struktur data atau akses basis data eksternal.
3. Inisialisasi dan kesalahan terminasi.
4. Validitas fungsional.
5. Kesensitifan sistem terhadap nilai input tertentu.
6. Batasan dari suatu data.



Sumber: (Rouf, 2012: 4)

Gambar 2.2. Sistem Kerja *Black Box Testing*

Penulis menggunakan pengujian dengan *black box testing* dikarenakan *black box testing* merupakan teknik pengujian sistem yang paling mudah untuk digunakan karena *black box testing* hanya fokus ke hasil eksekusi dan fungsionalitas sistemnya saja.