

BAB III

LANDASAN TEORI

3.1. Perangkat Lunak

Menurut Roger S Pressman di dalam bukunya mengartikan perangkat lunak sebagai sekumpulan instruksi yang jika dieksekusi menghasilkan *output* yang diharapkan [7]. Perangkat lunak juga dapat diartikan sebagai sekumpulan struktur data yang dapat memanipulasi informasi. Roger S Pressman membagi jenis aplikasi menjadi setidaknya tujuh kategori:

1. *System Software*

Sekumpulan program yang dibuat untuk menunjang fungsi sistem yang lain. Contohnya seperti *compiler*, *operating system* dll.

2. *Application Software*

Merupakan sebuah perangkat lunak yang berfungsi untuk mengerjakan tugas umum secara spesifik. Contohnya seperti *Point of Sales*, *Manufacturing Control System* dll.

3. *Engineering Software*

Jika aplikasi dapat mengolah suatu formula atau algoritma tertentu dan biasa digunakan pada bidang sains, bias dikatakan aplikasi tersebut tergolong pada jenis *engineering software*. Beberapa contohnya seperti Matlab, AutoCAD dll.

4. *Embedded Software*

Sebuah perangkat lunak yang ditulis lalu ditanam pada perangkat *internet of things*.

5. *Product Line Software*

Aplikasi yang didesain untuk memenuhi kebutuhan spesifik banyak pengguna. Seperti aplikasi mengolah kata dan lain-lain.

6. *Web App*

Perangkat lunak *web-app* berjalan pada *browser*. Meskipun begitu, platform *web* perlahan berevolusi menjadi *web 2.0* yang tidak selalu berjalan pada *browser*.

7. *Artificial Intelligent pada Software*

Perangkat lunak yang dapat melakukan komputasi dan menerapkan algoritma tertentu untuk menyelesaikan masalah yang kompleks

3.2. Sistem Informasi

Tata Sutabri dalam bukunya yang berjudul Konsep Sistem Informasi menjelaskan Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan [8]. Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. Perencanaan sistem informasi bertujuan untuk menjadikan sistem

informasi yang efektif dan efisien, guna sistem yang efektif dan efisien tidak lain untuk mendapatkan keunggulan dalam kompetisi. Faktor efisiensi setiap sistem berbeda. Oleh karena itu bahan perencanaan sistem informasi harus berkisar pada keempat tingkat.

Menurut James O'Brien (2010, p26) sistem adalah sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam proses transformasi yang teratur. Menurut O'Brien (2010, p34) informasi adalah data yang telah dikonversi ke dalam konteks yang bermakna dan berguna bagi pengguna akhir tertentu. Sehingga dapat ditarik kesimpulan bahwa menurut James O'Brien (2010, p4) sistem informasi dapat merupakan kombinasi teratur dari orang-orang, hardware, software, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi [9].

Menurut O'Brien terdapat 3 peran utama sistem informasi dalam bisnis yaitu mendukung proses bisnis dan operasional, mendukung pengambilan keputusan dan mendukung strategi untuk keunggulan kompetitif.

3.3. PHP (Hypertext Preprocessor)

Menurut Arief (2011), PHP adalah Bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side-scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya akan dikirimkan ke *browser* dengan format HTML [10].

Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh *user* sehingga keamanan halaman web yang lebih terjamin. PHP dirancang untuk membuat halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti isi *database* ke halaman web.

3.4. Laravel

Laravel adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (*model view controller*). Laravel adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu[11].

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti : manipulasi data, *controller*, dan *user interface*.

1. Model, model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
2. View, *view* adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
3. *Controller*, *controller* merupakan bagian yang menjembatani *model* dan *view*.

3.5. MySQL

MySQL merupakan salah satu perangkat lunak *Database* Manajemen System (DBMS) yang sering digunakan saat ini, yang didistribusikan secara gratis di bawah lisensi GPL (GNU *General Public License*). Sehingga setiap orang mudah untuk mendapatkan dan bebas untuk menggunakan MySQL. SQL sendiri merupakan suatu bahasa yang dipakai didalam pengambilan data pada relational *database* atau *database* yang terstruktur. Jadi mysql adalah *database management system* yang menggunakan bahasa sql sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server* [12].

Kelebihan yang dapat di miliki oleh MySQL, sebagai berikut:

- a. Bersifat *open source*, yang memiliki kemampuan untuk dapat dikembangkan lagi.
- b. Menggunakan Bahasa SQL (*Structure Query Language*), yang merupakan standar Bahasa dunia dalam pengolahan data.
- c. *Super performance* dan *realible*, tidak bias diragukan, pemrosesan *databasenya* sangat cepat dan stabil.
- d. Sangat mudah dipelajari (*easy of use*)
- e. Memiliki dukungan *support (group)* pengguna MySQL.
- f. Mampu lintas *platform*, dapat berjalan diberbagai sistem operasi.
- g. *Multiuser*, dimana MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami konflik.

3.6.UML (Unified Modeling Language)

UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek).

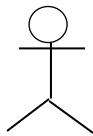
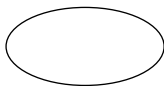
Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami [13].







Berikut ini adalah penjelasan dari masing-masing diagram:

- a. *Use case* merupakan alat komunikasi tingkat tinggi untuk mewakili persyaratan sistem. Diagram menunjukkan interaksi antara pengguna dan entitas eksternal lainnya dengan sistem yang sedang dikembangkan.

Simbol dari use case diagram ditunjukkan pada tabel berikut:

Tabel 3.1 Simbol-simbol pada Use case Diagram.




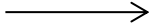

Simbol	Nama	Keterangan
	<i>Actor</i>	Pelaku, seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun.
	<i>Use case</i>	Menggambarkan bagaimana seseorang menggunakan sistem.

Simbol	Nama	Keterangan
	<i>Relasi</i>	Hubungan dengan <i>use case</i> lain, maupun relasi antara aktor dan <i>use case</i> .
	<i>Batasan Sistem</i>	Untuk memperlihatkan batasan sistem dalam diagram <i>use case</i> .
	<i>Dependency</i>	Suatu hubungan semantic antara dua <i>things</i> dimana perubahan pada suatu <i>things</i> (<i>independent</i>) mungkin mempengaruhi semantic <i>things</i> (<i>independent</i>) lain.
	<i>Include</i>	Mengspesifikasikan pada <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Mengspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Note</i>	Elemen fisik yang eksis pada saat aplikasi dijalankan dan

Simbol	Nama	Keterangan
		kemudian mencerminkan suatu sumber daya komputasi.

- b. *Activity Diagram* menangkap alur dari sebuah sistem, termasuk tindakan utama dan poin keputusan. Diagram ini berguna untuk mendokumentasikan proses bisnis.



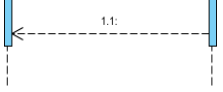
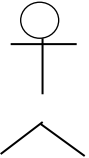
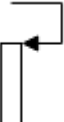
Tabel 3.2. Simbol-simbol pada Activity Diagram

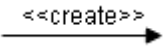
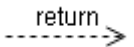
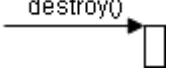
Simbol	Nama	Keterangan
	<i>Start State</i>	Menunjukkan dimulainya suatu <i>workflow</i>
	<i>End State</i>	Menggambarkan akhir dari pada sebuah <i>activity diagram</i>
	<i>Activities</i>	Menggambarkan sebuah pekerjaan atau tugas dalam <i>workflow</i>
	<i>State Transition</i>	Menunjukkan kegiatan apa berikutnya setelah kegiatan sebelumnya.
	<i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.

c. *Sequence Diagram* secara khusus menjabarkan sebuah *Use Case*.

Diagram ini menunjukkan sejumlah objek dan pesan yang melewati suatu objek.



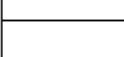

Tabel 3 3. Simbol-simbol pada Sequence Diagram


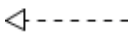
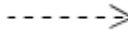

Simbol	Nama	Keterangan
	<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Call Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	<i>Return Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	<i>Actor</i>	Pelaku, seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun.
	<i>Self Message</i>	Jenis pesan yang mewakili permohonan dari <i>lifeline</i> yang sama.

Simbol	Nama	Keterangan
	<i>Create</i>	Menyatakan suatu objek yang membuat objek yang lain, arah panah mengarah pada objek.
	<i>Return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode yang dapat menghasilkan suatu kembalian ke objek tertentu.
	<i>Destroy</i>	Menyatakan suatu objek yang mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, dan sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

- d. *Class Diagram* menggambarkan struktur statis dari kelas dalam sistem dan menggambarkan atribut, operasi dan hubungan antara kelas.

Tabel 3.4 Simbol-simbol pada Class Diagram.

Simbol	Nama	Keterangan
	<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>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

Simbol	Nama	Keterangan
	<i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<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
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

3.7. *Wamp Server*

WampServer adalah platform pengembangan web berbasis Windows untuk aplikasi web dinamis menggunakan *server* Apache2, bahasa *scripting* PHP, dan *database* MySQL[13]. Ini juga memiliki PHPMyAdmin untuk lebih mudah mengelola basis data. Fitur lain yang ada pada *wamp server* :

1. TrayIcon yang memungkinkan untuk dengan mudah mengelola dan mengkonfigurasi *server* tanpa menyentuh file.
2. Layanan apache dan MySQL
3. Beralih ke mode *online/offline*
4. *Instal* dan ubah versi Apache, MySQL dan PHP
5. Kelola pengaturan konfigurasi *server*
6. Akses file konfigurasi

3.8. *Black Box Testing*

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 [15]. *Black Box Testing* berusaha untuk menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan dalam struktur atau akses databse eksternal.
3. Kesalahan perilaku (*behavior*) atau kesalahan kinerja.
4. Kesalahan antarmuka.