

BAB III

LANDASAN TEORI

3.1.Aplikasi

Website adalah fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lain (*hypertext*), baik diantara *page* yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. Sebuah situs *web* biasanya ditempatkan setidaknya pada sebuah *server web* yang dapat diakses melalui jaringan seperti *internet*, ataupun jaringan wilayah lokal yang disebut melalui alamat *internet* yang dikenali sebagai *URL (Uniform Resource Locator)*. Gabungan atas semua situs yang dapat diakses publik di *internet* disebut pula sebagai *World Wide Web* atau lebih dikenal dengan singkatan *WWW*. Meskipun setidaknya halaman beranda situs *internet* umumnya dapat diakses publik secara bebas, pada praktiknya tidak semua situs memberikan kebebasan bagi publik untuk mengaksesnya. Beberapa situs *web* mewajibkan pengunjung untuk melakukan pendaftaran sebagai anggota atau bahkan meminta pembayaran untuk dapat menjadi anggota untuk dapat mengakses isi yang terdapat dalam situs *web* tersebut, misalnya situs-situs yang menampilkan situs-situs berita, layanan surel (*e-mail*), dan lain-lain. Pembatasan-pembatasan ini umumnya dilakukan karena alasan keamanan, menghormati privasi atau karena tujuan komersial tertentu.[5]

3.2. PHP (Perl Hypertext Preprocessor)

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*. Maka dari itu kode program yang akan ditulis dalam *PHP* tidak akan terlihat oleh pengguna atau *user* sehingga keamanan halaman *website* akan terjamin. Selain itu *PHP* juga didesain untuk membuat halaman *website* yang dinamis, yaitu sebuah halaman *website* yang dapat membuat suatu tampilan berdasarkan perintah terbaru, seperti menampilkan isi basis data ke halaman *website*. [6]

Bahasa pemrograman *PHP* atau singkatan dari *Personal Home Page* merupakan sebuah bahasa skrip yang tertanam dalam *HTML* untuk dieksekusi bersifat *server side*. *PHP* merupakan salah satu bahasa pemrograman yang bersifat *open source*, sehingga *source code PHP* dapat disebarluaskan dan diubah secara bebas. Jika anda ingin mendownload versi terbaru *PHP* anda dapat mendownloadnya di situs resminya. *PHP* juga mampu berjalan di berbagai *web server* seperti *PWS* (*Personal Web Server*), *Apache*, *IIS* (*Internet Information Server*) dan *Xitami*. Selain itu, *PHP* juga dapat berjalan di banyak sistem operasi yang sudah banyak beredar saat ini seperti *Microsoft Windows* (Semua Versi), *Mac OS*, *Linux* dan *Solaris*. *PHP* dapat dibangun untuk modul *web server Apache* dan *binary* yang dapat berjalan sebagai *CGI* (*Common Gateway Interface*). *PHP* dapat mengatur *cookies*, mengirim *HTTP*

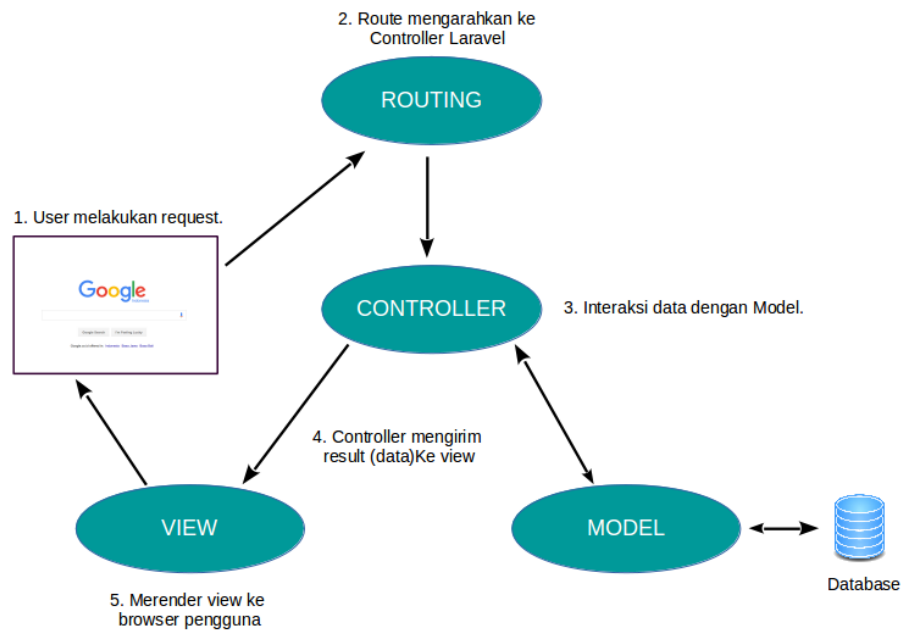
header, mengatur *authentication* dan *redirect user*. Keunggulan yang dimiliki oleh bahasa pemrograman *PHP* yaitu mempunyai kemampuan untuk melakukan koneksi ke berbagai macam *software* sistem management berbasis data atau *DBMS* (*Database Management System*) sehingga dapat membuat suatu halaman *website* dinamis. Selain itu, *PHP* juga memiliki konektivitas yang baik dengan beberapa *DBMS* tertentu seperti *MySQL*, *Oracle*, *mSQL*, *Sybase*, *Microsoft SQL Server*, *Adabas*, *Solid*, *PostgreSQL*, *Unix dbm*, *Velocis*, *FilePro*, *dBase*, dan juga semua *database* yang ber-interface *ODBC*. [7]

3.3. Laravel

Laravel adalah sebuah *framework PHP* yang dirilis dibawah lisensi *MIT*, dibangun dengan konsep *MVC* (*model view controller*). Sama seperti *framework PHP* lainnya. *Laravel* dibangun dengan basis *MVC* (*Model-View-Controller*). Yang disebut dengan *Model* biasanya bagian yang berurusan dengan *database*. *View* adalah bagian antarmuka atau bagian depan aplikasi, segala sesuatu yang diproses dalam sistem akan ditampilkan pada *view*. sedangkan *Controller* adalah bagian yang menangani atau menengahi antar *model* dan *view* jadi *controller* yang berperan sebagai pengendali dari sebuah sistem. *Laravel* dilengkapi *command line tool* yang bernama “*Artisan*” yang bisa digunakan untuk *packaging bundle* dan *instalasi bundle*. [8]

3.3.1. Sistem Kerja Laravel

Setiap aplikasi memiliki beberapa tahapan untuk mengerjakan begitu pula



dengan *framework Laravel*. Alur kerja dari *Laravel* dapat dilihat pada gambar 3.1.

Gambar 3.1 Alur Kerja Framework Laravel

1. User melakukan inputan *request* melalui *browser* atau *form*.
2. *Request* tersebut kemudian akan di terima *controller* untuk diproses *requestnya*.
3. Dianggap *request* tersebut melibatkan *request* ke *database*, maka diteruskanlah *request* tersebut ke *Model*. *Model* kemudian mengambil data yang diminta *controller*.

4. Setelah mendapatkan data yang diminta *user*, maka akan dikembalikan ke *controller*.
5. Kemudian *Controller* akan meneruskan data yang diberikan *Model* ke *View*.
6. *View* kemudian memberikan data kepada *user* sesuai *request*.

3.4. MySQL

MySQL merupakan *RDBMS* (*server database*) yang mengelola *database* dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak *user*. *MySQL* adalah *database server open source* yang cukup populer keberadaanya. Dengan berbagai keunggulan yang dimiliki, membuat *software database* ini banyak digunakan oleh praktisi untuk membangun suatu *project*. Adanya fasilitas *API* (*Application Programming Interface*) yang dimiliki oleh *Mysql*, memungkinkan bermacam-macam aplikasi komputer yang ditulis dengan berbagai bahasa pemograman dapat mengakses basis data *MySQL*. *MYSQL* termasuk jenis *RDBMS* (*Relational Database Management System*). Sedangkan *RDMS* sendiri akan lebih banyak mengenal istilah seperti *table*, baris dan kolom digunakan dalam perintah-perintah di *MYSQL*. [9]

3.5. PHPStorm

PhpStorm adalah *IDE* lintas *platform* komersial untuk *PHP*, yang dibangun oleh perusahaan *JetBrains* yang bermarkas di Republik Ceko. *PhpStorm* menyediakan editor untuk *PHP*, *HTML*, dan *JavaScript* dengan analisis kode *on-the-fly*, pencegahan kesalahan, dan *refactoring* otomatis untuk kode *PHP* dan *JavaScript*. [10]


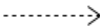
3.6. Unified Modelling Language (UML)






UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *UML* bertujuan untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.[11]



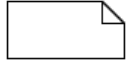
3.6.1. Use Case Diagram

Use *case* diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan informasi yang akan dibuat. Simbol-simbol yang ada pada *use case* diagram dapat dilihat pada Tabel 1:

Tabel 3.1. Simbol Use Case Diagram.

No.	Simbol	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


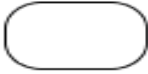



No.	Simbol	Nama	Keterangan
			(<i>independent</i>).
3.		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</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.

No.	Simbol	Nama	Keterangan
8.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
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.

3.6.2. Activity Diagram

Activity diagram atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sisten atau menu yang ada pada perangkat lunak. Simbol-simbol yang ada pada diagram aktivitas dapat dilihat pada Tabel 3.2 berikut:

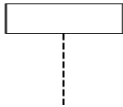


Tabel 3.2. Simbol Activity Diagram.

No	Simbol	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</i> <i>Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

3.6.3. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Tabel 3.3. Simbol Sequence Diagram

No.	Simbol	Nama	Keterangan
1.		<i>Life Line</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



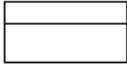


3.6.4. Class Diagram



Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini. *Forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya, merubah kode program menjadi model.

Diagram kelas memiliki fitur-fitur yang menyertainya yaitu atribut dan operasi. Atribut (*attribute*) dan operasi (*operation*) menggambarkan perilaku

suatu kelas serta perluasannya seperti *stereotypes*, *tagged values*, dan batasan (*constraints*) merupakan fitur-fitur sebuah kelas.

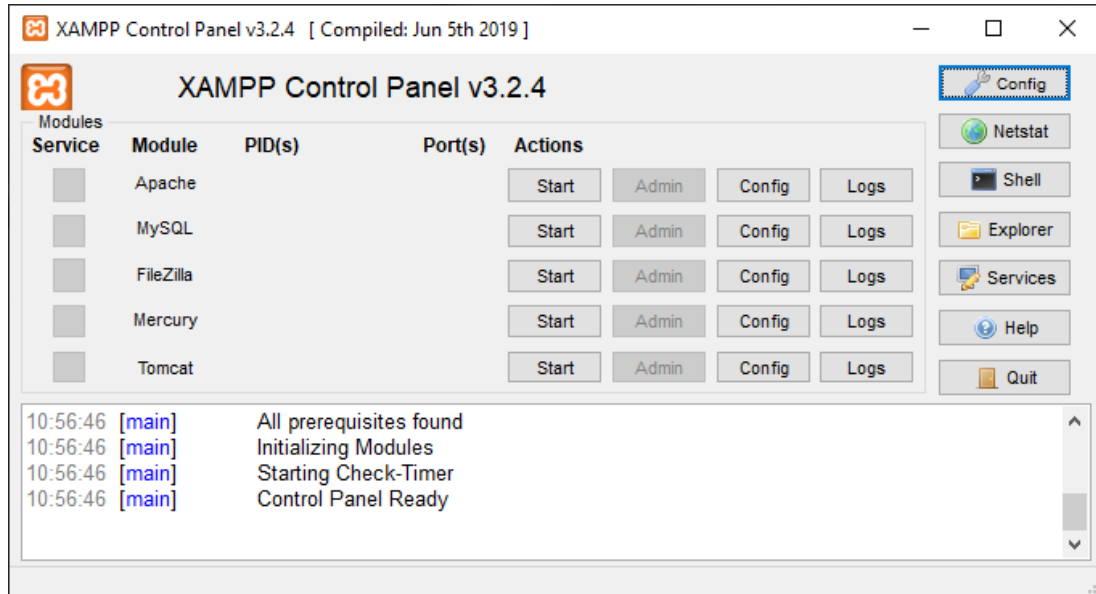
Tabel 3.4. Simbol Class Diagram

No.	Simbol	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 aktor.
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

No.	Simbol	Nama	Keterangan
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.

3.7. Xampp

XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman *web*, khususnya *PHP* dan *MySQL*. *Xampp* perangkat lunak *open source* yang diunggah secara gratis dan bisa dijalankan di semua semua operasi seperti *windows*, *linux*, *solaris*, dan *mac*.



Tampilan control panel xampp seperti gambar 3.2.

Gambar 3.2. Tampilan Control Panel XAMPP

Di dalam folder utama xampp, terdapat beberapa folder penting yang perlu diketahui. Untuk lebih memahami setiap fungsinya, dapat melihat penjelasannya sebagai berikut:

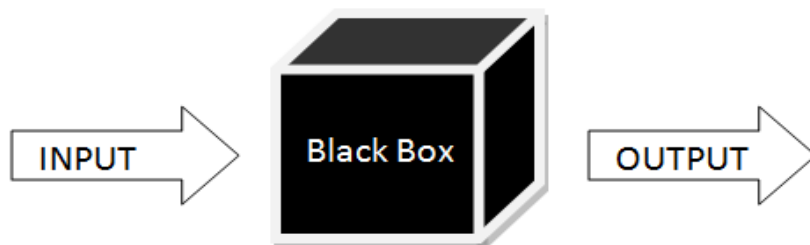
Tabel 3.5. penjelasan folder xampp

Folder	Keterangan
<i>Apache</i>	Folder utama <i>Apache Webserver</i>
<i>Htdocs</i>	Folder utama untuk menyimpan data-data latihan <i>web</i> , baik <i>PHP</i> maupun <i>HTML</i> biasa.
<i>Manual</i>	Berisi <i>subfolder</i> yang didalamnya terdapat manual program dan

Folder	Keterangan
	<i>database</i> termasuk manual <i>PHP</i> dan <i>MySQL</i>
<i>MySQL</i>	Folder utama untuk <i>database MySQL Server</i> .
<i>PHP</i>	Folder utama untuk program <i>PHP</i>

3.8. Blackbox Testing

Blackbox testing juga disebut pengujian tingkah laku, memusat pada kebutuhan fungsional perangkat lunak. Teknik pengujian *blackbox* memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan *fungsional* untuk suatu program. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data (pengaksesan basis data), kesalahan performansi, kesalahan



inisialisasi dan akhir program. Metode *blackbox testing* seperti gambar 3.3.

Gambar 3.3 Metode Blackbox Testing

Blackbox testing untuk menemukan hal-hal berikut:

1. Menemukan fungsi-fungsi yang tidak benar atau hilang di dalam suatu *software*.
2. Mencari kesalahan *interface* yang terjadi saat *software* dijalankan.
3. Untuk mengetahui kesalahan dalam struktur data atau akses *database eksternal* di dalam suatu aplikasi.
4. Menguji kinerja dari *software*.
5. Menginisialisasikan dan mencari kesalahan dari terminasi *software* itu sendiri.