

BAB III

LANDASAN TEORI

3.1 Lembaga Kursus dan Pelatihan

Kursus merupakan suatu lembaga pelatihan dari satuan pendidikan non formal. Dan metode pembelajaran berlangsung seperti halnya kegiatan belajar mengajar pada umumnya. Perbedaannya adalah biasanya kursus mempelajari satu keterampilan dan dengan waktu yang sangat singkat. Pelatihan adalah pemberian suatu kegiatan yang berisi pengetahuan, keterampilan, informasi untuk dapat merubah kehidupan seseorang ke arah yang lebih baik.

Pelatihan adalah proses meningkatkan pengetahuan dan keterampilan karyawan. Pelatihan mungkin juga meliputi perubahan yang ada pada diri karyawan tersebut. Umumnya hasil yang diinginkan dari pelatihan ialah penguasaan atau peningkatan. Proses pelatihan dikendalikan oleh pemilik keahlian yang diajarkan atau ahli yang membantu mengembangkan keterampilan melalui pengalaman terstruktur [7].

3.2 Sistem

Sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan. Adapun manfaat sistem yaitu untuk menyatukan atau mengintegrasikan semua unsur yang ada dalam suatu ruang lingkup, dimana komponen-komponen tersebut tidak dapat berdiri sendiri. Komponen atau sub sistem harus saling berintegrasi dan berhubungan untuk membentuk satu kesatuan sehingga sasaran dan tujuan

sistem tersebut bisa tercapai [8]. Berdasarkan dari definisi diatas, dapat disimpulkan bahwa sistem adalah sekumpulan elemen-elemen atau fungsi-fungsi tertentu yang bekerja dalam suatu proses untuk mencapai suatu tujuan tertentu.

3.3 Informasi

Informasi adalah kumpulan data yang diproses dan diolah menjadi data yang memiliki arti bagi penerimanya yang menggambarkan suatu kejadian - kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan suatu keputusan dipahami [9]. Berdasarkan pendapat yang dikemukakan diatas menunjukkan bahwa informasi merupakan hasil pengolahan data yang berguna sebagai dasar untuk mengambil keputusan.

3.4 Sistem Informasi

Sistem informasi dapat didefinisikan sebagai serangkaian komponen yang saling berhubungan yang mengumpulkan (atau mendapatkan), memproses, menyimpan, dan mendistribusikan informasi yang mendukung pengambilan keputusan dan pengawasan di dalam sebuah organisasi [10]. Disimpulkan dari definisi tersebut, sistem informasi adalah sekumpulan elemn atau fungsi yang membentuk sistem dan memberikan output berupa laporan yang berguna bagi pemakai sistem informasi tersebut.

3.5 Website

World Wide Web atau *Web* adalah salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet. *Web* adalah suatu metode untuk menampilkan informasi di internet, baik berupa teks, gambar, suara

maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *browser*.

Website adalah istilah untuk sekelompok halaman web (*web pages*) yang umumnya merupakan bagian dari nama domain *WWW* atau subdomain di internet. Sebuah situs bersifat statis dan dinamis terdiri dari serangkaian bangunan yang saling berhubungan, masing-masing dihubungkan oleh jaringan (*hyperlink*). Bersifat statis ketika isi informasi tetap, jarang berubah, dan informasi diberikan secara sepihak oleh pemilik *website*. Ketika konten informasi situs *web* terus berubah, itu dinamis. Situs *website* yang memiliki fungsi informasi pada umumnya lebih menekankan pada kualitas bagian kontennya, karena tujuan situs tersebut adalah menyampaikan isinya [11].

3.6 XAMPP

Xampp adalah program yang dapat mengubah komputer kita menjadi server. Menggunakan *Xampp* terdiri dari membuat jaringan lokal anda sendiri artinya kita dapat membuat situs *web offline* untuk masa percobaan di komputer kita. Jadi fungsi dari *Xampp* server itu sendiri merupakan server *website* kita untuk cara memakainya. Disebut server karena dalam hal ini komputer yang akan kita pakai harus memberikan pelayanan untuk mengakseskan *web*, untuk itu komputer kita harus menjadi server.

Dapat disimpulkan *Xampp* adalah aplikasi tools untuk menyediakan paket lunak yang berisi konfigurasi *Apache*, *MySQL*, *PHP*, dan *Perl* untuk

membantu kita dalam proses pembuatan aplikasi *web* yang menyatu menjadi satu sehingga memudahkan kita dalam membuat program *web* [12].

3.7 *Database*

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. *Database* adalah representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. *Database* merupakan sekumpulan informasi yang saling berkaitan pada suatu subjek tertentu pada tujuan tertentu pula. *Database* adalah susunan *record* data operasional lengkap dari suatu organisasi atau perusahaan, yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dalam komputer sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna [13].

3.8 *MySQL*

MySQL adalah salah satu perangkat lunak *Database Management 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*.

Sebagai *software database* dengan konsep *database modern*, *MySQL* memiliki banyak sekali kelebihan. Beberapa kelebihan dari *MySQL* adalah *Portability, Open Source, Multiuser, Performance Tuning, Column Types,*

Command dan Function, Security, Scalability dan Limits, Connectivity, Localisation, Interface, Clients dan Tools, Struktur Tabel [14].

3.9 PHP

PHP merupakan bahasa *scripting* seperti *HTML*. Dalam pengembangan *web* pada *HTML* yang memungkinkan dibuatnya aplikasi dinamis dan terdapat proses pengolahan data dan pemrosesan data. Semua *syntax* yang diberikan akan sepenuhnya dijalankan pada *server*, sedangkan yang dikirimkan ke *browser* hanya hasilnya saja. *PHP* merupakan bahasa berbentuk *script* yang ditempatkan dalam *server* dan diproses di *server*.

PHP dikenal sebagai bahasa *scripting*, yang menggabungkan tag *HTML*, berjalan di *server*, dan digunakan untuk membuat halaman *web* dinamis seperti *Active Server Pages (ASP)* atau *Java Server Pages (JSP)*. *PHP* adalah perangkat lunak sumber terbuka. *PHP* pada awalnya merupakan kepanjangan dari *Personal Home Page*. Seiring dengan perkembangan aplikasi dan kegunaannya, kepanjangan *PHP* kemudian berubah menjadi *PHP Hypertext Pre-Processor* [15].

3.10 Framework Laravel

Framework adalah sebuah arsitektur yang terbuka yang dibuat berdasarkan pada standar pengembangan perangkat lunak yang diterima secara umum. Penggunaan *Framework* secara signifikan mengurangi penggunaan waktu, usaha dan sumber daya yang dibutuhkan untuk mengembangkan dan *maintenance* aplikasi *web*. *Framework* yang sangat familiar pada era sekarang yaitu *Laravel*.

Laravel adalah *framework* pengembangan *web* dalam PHP. *Laravel* adalah *framework* yang menekankan kesederhanaan dan fleksibilitas dalam desainnya. Seperti *framework* lainnya, *laravel* didasarkan pada MVC (*Model-View-Controller*). *Laravel* adalah salah satu kerangka kerja terbaik untuk pengembang *PHP* untuk membuat aplikasi yang elegan dan dinamis. *Laravel* adalah *framework* terbaik tahun 2014. *Laravel* hadir dengan alat baris perintah yang disebut *Artisan* yang dapat digunakan untuk membundel paket dan menginstal paket dari baris perintah.

Ada banyak kelebihan yang dimiliki dari *laravel*. Kelebihan tersebut diantaranya : *expressif*, simple (karena adanya *Eloquent ORM*), *accessible* (dibuat dengan dokumentasi yang selengkap mungkin). Selain itu di *Laravel* terdapat banyak fitur. Fitur tersebut diantaranya *bundles* (sebuah fitur dengan sistem pengemasan modular dan berbagai bundle telah tersedia untuk digunakan dalam aplikasi), *Eloquent ORM* (penerapan *PHP* lanjutan dari pola “*active record*” menyediakan metode internal untuk mengatasi kendala hubungan antara objek *database*), *Application logic* (aplikasi pengembangan dari *Controllers* maupun bagian dari deklarasi *Route*), *Reverse routing* (hubungan antara *Link* dan *Route*), *Restful controllers* (option untuk memisahkan logika dalam melayani *HTTP GET* dan permintaan *POST*), *Class auto loading* (otomatis loading untuk class *PHP*), *View composers* (kode unit logical yang dijalankan ketika sebuah *view* di load), *IoC Container* (objek baru yang dihasilkan dengan mengikuti prinsip control pembalik), *Migrations* (versi sistem *control* untuk skema *database*), *Unit Testing* (untuk

mendeteksi regresi), *Automatic Pagination* (menyederhanakan tugas dari penerapan halaman) [16].

3.11 MVC (*Model View Controller*)

Framework aplikasi *web* biasanya menerapkan pola desain yang disebut *Model*, *View*, dan *Controller*, atau lebih dikenal dengan MVC.

Modul *model* berisi kelas yang mewakili tabel dalam *database* yang berisi *instances* yang digunakan untuk memanipulasi *database*. *Model* biasanya digunakan sebagai penghubung antara modul *controller* dan *database* saat *controller* ingin mengambil dan menggunakan data di *database*.

Modul *controller* adalah kelas yang dibuat oleh programmer untuk menangani logika aplikasi dan *user events*. Dalam aplikasi yang menggunakan pola *MVC*, *controller* bertindak sebagai otak dari sistem, menyediakan jembatan antara *model* dan *view*. *Controller* juga berfungsi menerima *request* dari *user* dan kemudian memprosesnya.

Modul *view* berfungsi untuk menerima dan menampilkan data yang dikirimkan oleh *controller*. *View* pada aplikasi berbasis *web* biasanya berupa sekumpulan halaman *HTML*. *MVC* membantu mengurangi kerumitan pembuatan desain dan meningkatkan fleksibilitas dan penggunaan kembali kode [17].

3.12 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah

software yang berorientasikan pada objek. *UML* merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik [18]. Terdapat beberapa diagram *UML* yang sering digunakan dalam pengembangan sebuah sistem, yaitu :

a. *Use Case Diagram*

Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat aktor yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.

Tabel 3.1 Simbol-Simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

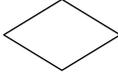
Simbol	Nama	Keterangan
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen <i>independent</i> (Mandiri) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek <i>descendent</i> (Anak) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek <i>ancestor</i> (Induk).
<<include>> 	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
<<extend>> 	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

Simbol	Nama	Keterangan
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen - elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

b. *Activity Diagram*

Merupakan gambaran alir dari aktivitas-aktivitas didalam sistem yang berjalan.

Tabel 3.2 Simbol – Simbol *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>Decision</i>	Suatu titik atau <i>point</i> pada <i>activity diagram</i> yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.
	<i>State Transition</i>	Menunjukkan kegiatan berikutnya setelah kegiatan sebelumnya.

c. *Sequence Diagram*

Menggambarkan interaksi antar objek didalam dan di sekitar sistem yang berupa *message* yang digambarkan terhadap waktu.

Tabel 3.3 Simbol – Simbol *Sequence Diagram*

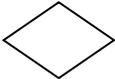
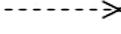
Simbol	Nama	Keterangan
	<i>Life Line</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi.

d. *Class Diagram*

Merupakan gambaran struktur dan deskripsi dari *class*, *package*, dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.

Tabel 3.4 Simbol – Simbol *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>).

Simbol	Nama	Keterangan
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.
	<i>Class</i>	Himpunan dari obek-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.
	<i>Realization</i>	Operasi yang benar – benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Suatu hubungan <i>semantic</i> antara dua things dimana perubahan pada suatu things (<i>independent</i>) mungkin mempengaruhi <i>semantic things</i> (<i>independent</i>) lain.
	<i>Association</i>	Penghubung objek satu dengan objek lainnya.

3.13 *Blackbox Testing*

Blackbox Testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Blackbox Testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. *Blackbox Testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan

melatih seluruh syarat-syarat fungsional suatu program [19]. Keuntungan penggunaan metode *Blackbox Testing* adalah :

- a. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.
- b. Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.
- c. *Programmer* dan *tester* keduanya saling bergantung satu sama lain.