

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Konsep dasar sistem meliputi dasar dasar apa saja yang penulis pakai untuk membuat sistem perancangan sistem informasi pendaftaran praktek kerja lapangan atau riset di web pusat pendidikan dan pelatihan kementerian ketenagakerjaan.

2.1.1 Praktek Kerja Lapangan

Praktek kerja lapangan juga ialah suatu kegiatan pendidikan yang wajib di ikuti untuk siswa/mahasiswa, praktek kerja lapangan bisa diartikan sebagai pelatihan, Menurut (Widodo, 2015:82) pelatihan didefinisikan sebagai berikut "serangkaian aktivitas individu dalam meningkatkan keahlian dan pengetahuan secara sistematis sehingga mampu memiliki kinerja yang profesional di bidangnya". Pelatihan adalah proses pembelajaran yang memungkinkan pegawai melaksanakan pekerjaan yang sekarang sesuai dengan standar.

Menurut (Widodo, 2015:84) mengemukakan bahwa tujuan pelatihan yang dilakukan oleh perusahaan sebagai berikut "untuk meningkatkan produktivitas, meningkatkan kualitas, mendukung perencanaan SDM, meningkatkan moral anggota, memberikan kompensasi yang tidak langsung, meningkatkan kesehatan dan keselamatan kerja, mencegah kedaluarsa kemampuan dan pengetahuan personel, meningkatkan perkembangan kemampuan dan keahlian personel".

2.1.2 Riset

Riset juga ialah suatu kegiatan pendidikan yang wajib di ikuti untuk mahasiswa, riset merupakan bagian dari metode penelitian dimana didalamnya terdapat kegiatan penyelidikan (*investigasi*) atau *eksplorasi* terhadap suatu masalah yang dilakukan menurut kaidah dan metodologi tertentu secara ilmiah dan sistematis.

Menurut (Sugiyono, 2015:2) metode penelitian didefinisikan sebagai berikut “Metode penelitian diartikan cara ilmiah untuk mendapatkan data dengan tujuan data kegunaan tertentu”. Menurut (Sugiyono, 2015:23) metode penelitian kuantitatif adalah “Metode kuantitatif dapat diartikan sebagai metode penelitian yang berlandaskan pada filsafat positivisme, digunakan untuk meneliti pada populasi atau sampel tertentu pengumpulan data menggunakan instrumen penelitian, analisis data bersifat kuantitatif/statistik, dengan tujuan untuk menguji hipotesis yang ditetapkan”.

Menurut (Sugiyono, 2015:38) Mendefinisikan objek penelitian sebagai berikut “Objek penelitian adalah suatu atribut atau sifat atau nilai dari orang, objek atau kegiatan yang mempunyai variasi tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya”. Menurut (Sugiyono, 2015:156) Mendefinisikan objek penelitian sebagai berikut “Instrumen penelitian adalah merupakan alat ukur seperti tes, kuesioner, pedoman wawancara dan pedoman observasi yang digunakan peneliti untuk mengumpulkan data dalam suatu penelitian”.

2.2. Teori Pendukung

Pengembangan perangkat lunak tidak lepas dari peran penggunaan teori pendukung yang membantu untuk merencanakan prosedur sistem berjalan yang dipakai disebuah program

2.2.1. ERD

Menurut (Shalahuddin & Rosa, 2015:35) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODMBS maka perancangan ERD tidak perlu dilakukan”.

Entity relationship diagram adalah pemodelan data utama dan akan membantu mengorganisasikan data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas,logical record structure representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas.

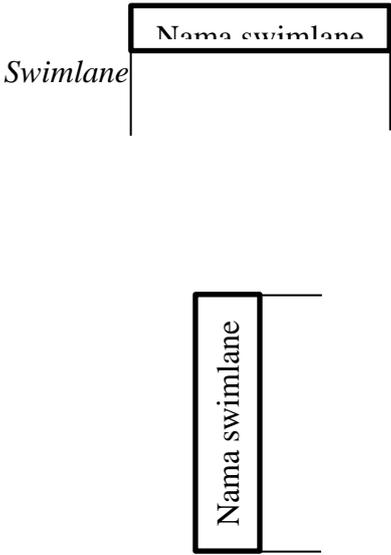
2.2.2. Activity Diagram

Menurut (Sukamto & Shalahuddin, 2014:161) “diagram aktivitas atau *activity diagram* adalah menggambarkan aliran kerja atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak “.

Activity diagram Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses,sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan activity diagram dibuat berdasarkan sebuah atau beberapa use case pada use case diagram.

Tabel II.1 Tabel *activity diagram*

Simbol/	Deskripsi
---------	-----------

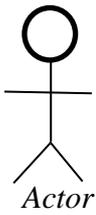
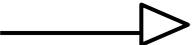
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ 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

2.2.3. Use Case

Menurut (Rosa & Shalahudin, 2014:155) berpendapat “*use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat”.

Use case Sebuah use case adalah situasi dimana system digunakan untuk memenuhi satu atau lebih kebutuhan pemakai, use case merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, design testing, dan dokumentasi, *use case* menggambarkan kebutuhan sistem dari sudut pandang di luar sistem.

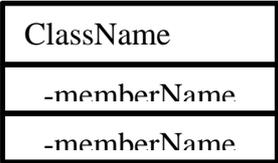
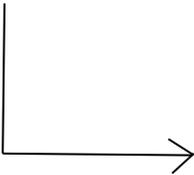
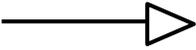
Tabel II.2 Tabel *usecase*

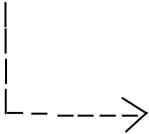
Symbol	Deskripsi
<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama Use Case
<p>Aktor/ <i>actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informaasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama actor
<p>Asosiasi / <i>association</i></p> 	Komunikasi antara aktor dan use case yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor
<p>Ekstensi/ <i>extend</i></p> <p>-<<<i>extend</i>>> -></p>	Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dinamakan <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>intherince</i> pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
<p>Generalisasi/ <i>generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari laainnya.

2.2.4. Class Diagram

Menurut (Rosa & Shalahudin, 2014:141) mengatakan bahwa “diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun system”

Tabel II.3 Tabel *class diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur system
Antarmuka/ <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 multiplicity
Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umumkhusus)

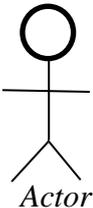
Kebergantungan/ <i>dependensi</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
--	---

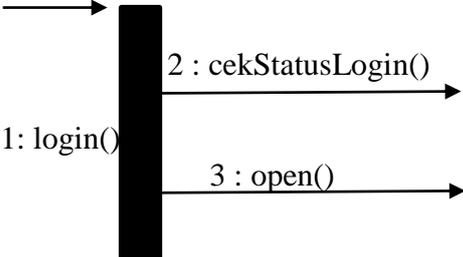
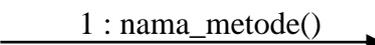
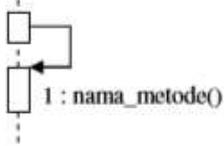
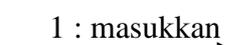
2.2.5. Sequence Diagram

Menurut (Rosa & Shalahudin, 2014:165) mengatakan bahwa “diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek”.

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Tabel II.4 Tabel *sequence diagram*

Symbol	Deskripsi
Aktor/ <i>actor</i> 	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 diawal <i>frase</i> nama actor
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek

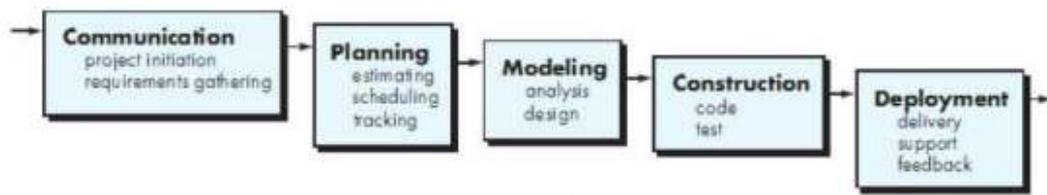
<p>Objek</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p><i>Nama objek : nama kelas</i></p> </div>	<p>Menyatakan objek yang berinteraksi Pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>

<p>Pesan tipe return</p> <p style="text-align: center;">1 : keluaran -----▶</p>	<p>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</p>
<p>Pesan tipe <i>destroy</i></p> <p style="text-align: center;"><<destroy>> ————▶</p>	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada <i>destroy</i></p>

2.2.6. Model Pengembangan Perangkat Lunak

Disiplin ilmu yang penulis pilih sebagai dasar untuk menentukan pembuatan aplikasi berdasarkan model pengembangan perangkat lunak yaitu model air terjun (waterfall).

Menurut (Pressman, 2015:42) "model waterfall adalah model klasik yang bersifat sistematis, berurutan dalam membangun software". Nama model ini sebenarnya adalah "Linear Sequential Model". Model ini sering disebut juga dengan "classic life cycle" atau metode waterfall. Model ini termasuk ke dalam model generic pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam Software Engineering (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Fase-fase dalam Waterfall Model menurut referensi Pressman :



Sumber: (Pressman, 2015:42)

Gambar II.1 Waterfall Pressman (Pressman, 2015:42)

1. Communication (Project Initiation & Requirements Gathering)

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan customer demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi software. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

2. Planning (Estimating, Scheduling, Tracking)

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan tracking proses pengerjaan sistem.

3. Modeling (Analysis & Design)

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur software, tampilan

interface, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

4. Construction (Code & Test)

Tahapan Construction ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

5. Deployment (Delivery, Support, Feedback)

Tahapan Deployment merupakan tahapan implementasi software ke customer, pemeliharaan software secara berkala, perbaikan software, evaluasi software, dan pengembangan software berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya.

2.2.7. Struktur Navigasi

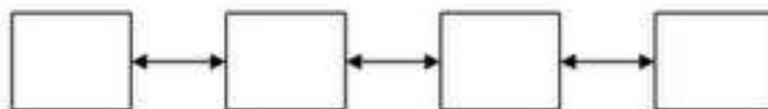
Menurut (Imaniawan & Elsa, 2017) “Struktur Navigasi dapat diartikan sebagai alur dari suatu program yang menggambarkan rancangan hubungan antara area yang berbeda sehingga memudahkan proses pengorganisasian seluruh elemen *website*”.

Navigasi termasuk struktur terpenting dalam pembuatan suatu aplikasi multimedia dan gambarnya harus sudah ada pada tahap perancangan dan rantai kerja dari beberapa area yang berbeda dan dapat membantu mengorganisasikan seluruh elemen pembuatan *web* dengan memberikan perintah atau pesan.

Ada 4 macam bentuk dasar dari navigasi yang bisa digunakan dalam proses pembuatan *web*, yaitu:

1. *Linear* (Satu Alur)

Linear (satu alur) merupakan struktur yang hanya mempunyai satu rangkaian cerita yang berurut. Dengan kata lain struktur ini hanya hanya dapat menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelum atau satu halaman sesudah dan tidak dapat menampilkan dua halaman sebelumnya atau dua halaman sesudahnya. Salah satu yang terpenting dari sturktur ini adalah tidak diperkenankan terjadinya percabangan.

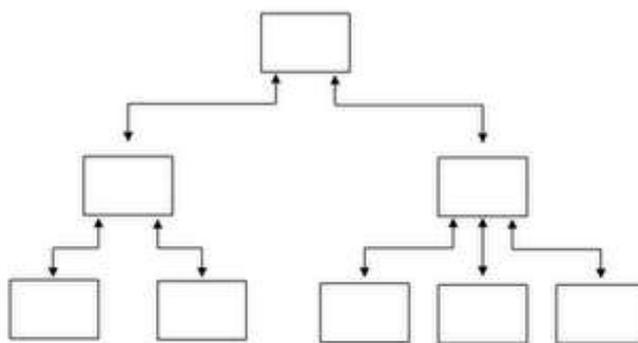


Sumber : Imaniawan & Elsa, 2017

Gambar II.2 Navigasi *Linear*

2. *Hierarchical* (Hirarki)

Struktur *Hierarchical* ini percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai *master page* (halaman utama kesatu), halaman utama ini akan mempunyai halaman percabangan yang dikatakan *slave page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama *master page* (halaman utama kedua) dan seterusnya. Yang terpenting dari struktur penjejakan ini tidak diperkenankan adanya tampilan secara *linear*.

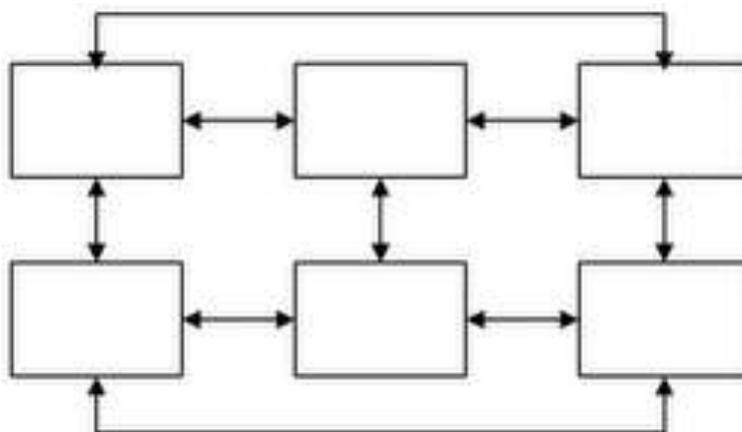


Sumber : Imaniawan & Elsa, 2017

Gambar II.3 Navigasi *Hierarchical* (Hirarki)

3. *Non Linear* (Tidak Berurut)

Struktur penjejakan *non Linear* (tidak berurut) merupakan pengembangan dari struktur penjejakan *linear*. Pada struktur ini diperkenankan membuat penjejakan bercabang. Pemakai bebas menelusuri *website* tanpa dibatasi oleh suatu rute dimana kontrol navigasi dapat mengakses ke semua halaman manapun. Percabangan yang dibuat pada struktur *non linear* ini berbeda dengan percabangan pada *Hierarchical*, karena pada percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama tidak ada *Master page* dan *Slave page*.

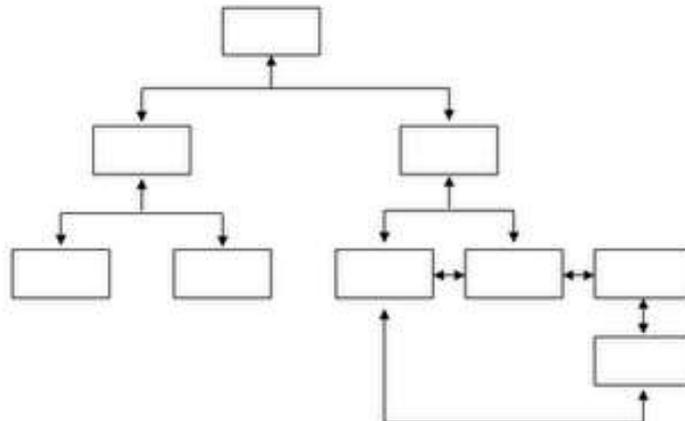


Sumber : Imaniawan & Elsa, 2017

Gambar II.4 Navigasi *Non Linear*

4. *Composite* (Campuran)

Composite (campuran) atau disebut juga struktur penjejakan bebas merupakan gabungan dari ketiga struktur sebelumnya yaitu *Linear*, *Non Linear*, dan *Hierarchical*. Jika suatu tampilan membutuhkan percabangan, maka dapat dibuat percabangan, dan bila dalam percabangan tersebut terdapat suatu tampilan yang sama kedudukannya maka dapat dibuat struktur *Linear* dalam percabangan tersebut. Setiap struktur peta penjejakan seperti yang baru dibahas mempunyai fungsi dan tujuan tersendiri, tidak ada yang lebih baik atau lebih buruk. Pengguna peta penjejakan bergantung kepada kebutuhan dan tujuan dari *Web* yang hendak dibuat. Semakin kompleks peta penjejakan yang digunakan, maka semakin sulit pembuatan *Page* dari peta penjejakan tersebut.



Sumber : Imaniawan & Elsa, 2017

Gambar II.5 Navigasi *Composite* (Campuran)

2.2.8. Basis Data

Basis data diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data merupakan representasi fakta dunia nyata yang mewakili

suatu objek. Menurut (Imaniawan & Elsa, 2017) “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”

Berikut ini adalah beberapa definisi basis data:

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

Aplikasi basis data yang digunakan untuk membangun website ini adalah sebagai berikut :

1. MySQL

Menurut (Sari, 2017) “MySQL adalah suatu RDBMS (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data”.

Menurut (Destiana & Hadidah, 2016) “MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional RDBMS (*Relational Database Management System*) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*)”.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan

pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah. Sebagai *software* DBMS, MySQL memiliki sejumlah fitur seperti yang di jelaskan di bawah ini:

a. *Multiplatform*

MySQL tersedia pada beberapa *platform* (*Windows, Linux, Unix* dan Lain-lain).

b. Andal cepat dan mudah digunakan

MySQL tergolong sebagai *database server* (*server* yang melayani terhadap permintaan *database*) yang dapat menangani *database* yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses *database*, dan sekaligus mudah untuk digunakan. Berbagai *tool* pendukung juga tersedia (walupun dibuat oleh pihak lain). Perlu diketahui MySQL dapat menangani sebuah tabel yang berukuran dalam *terabyte* (1 *terabyte* = 1024 *Gigabyte*). Namun, ukuran sesungguhnya bergantung kepada batasan sistem oprasi. Sebagai contoh : pada sistem Solaris 9/10, batasan ukuran file sebesar 16 *terabyte*.

c. Jaminan ke amanan akses

MySQL mendukung keamanan *database* dengan berbagai kriteria pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia, sedangkan *user* lain tidak boleh.

d. Dukungan MySQL

Seperti tersirat dalam namanya, MySQL mendukung perintah SQL (*Structured Query Language*). Sebagaimana diketahui, SQL merupakan

setandar dalam pengaksesan *database relational*. Pengetahuan akan SQL akan memudahkan siapapun untuk menggunakan MySQL.

2. PhpMyAdmin

Menurut (Saryoko, 2015) “PhpMyAdmin merupakan aplikasi berbasis web yang dikembangkan menggunakan bahasa pemograman PHP . Melalui PhpMyAdmin, *user* dapat melakukan perintah *query* tanpa harus mengetikkan seperti pada MS DOS. Perintah tersebut misalnya administrasi user dan *privileges*, *export* dan *import database*, manajemen *database*, manajemen *table* dan struktur *table*. PhpMyAdmin sangat *user friendly*, sehingga mudah untuk digunakan walaupun oleh pengguna baru”.

Setiap RDBMS (*Relation Database Management System*) seperti *Oracle*, *SQL Server*, *MySQL* dan lain-lain, pasti memiliki *tool* yang dapat digunakan untuk mempermudah pengoprasian *database*. *Oracle* memiliki *TOAD*. *SQL Server* memiliki *Enterprose Manager* dan *SQL Query Analyzer*. Sedangkan *MySQL* memiliki *tool* atau aplikasi yang disebut *PhpMyAdmin*.

2.2.9.Website

Untuk mengakses *internet* diperlukan suatu program aplikasi *Web Browser*. Program aplikasi ini mempunyai kemampuan untuk menampilkan suatu *Web Page* yang ditulis dalam format *HTML*. Menurut (Bekti, 2015:35) Website merupakan “kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman”.

Menurut (Saryoko, 2015) memeberikan penjelasan bahwa *internet* adalah ”kumpulan yang luas dari jaringan komputer besar dan kecil yang saling berhubungan menggunakan jaringan komunikasi yang ada diseluruh dunia”.

Internet berasal dari kata *Interconnection Networking* yang berarti kata jaringan yang saling berhubungan. Didalam mengatur integrasi dan komunikasi jaringan internet dibutuhkan sebuah *protocol* yang disebut TCP/IP (*Transmission Control Protocol/Internet Protocol*).

Secara umum *internet* dapat diartikan sebagai pertukaran informasi dan komunikasi. Semua informasi bisa didapatkan dengan mudah dan bebas di *internet* tanpa ada batasan.

2.3.0. Bahasa Pemrograman

Untuk mengembangkan proses pembuatan website penulis menggunakan beberapa bahasa pemrograman diantaranya, adalah:

1. HTML (*HyperText Markup Language*)

HTML akronim dari *Hypertext Markup Language*. Menurut (Destiana & Hadidah, 2016) “HTML (Hypertext Markup Language) merupakan sebuah kombinasi teks dan informasi yang dipakai untuk membuat sebuah halaman situs. HTML (Hypertext Markup Language) merupakan bahasa (kode) yang digunakan untuk membuat halaman web”.

Mengapa disebut *hypertext*, karena pemakai dapat membaca tampilan *website* tanpa harus baris demi baris, dapat meloncat dari satu baris ke baris yang lain. Disebut *markup language*, karena susunan instruksinya banyak menggunakan tanda-tanda yang disebut *tag*.

Halaman *web* yang biasa anda lihat ketika *browsing* di internet selalu ditulis menggunakan bahasa HTML. Walaupun halaman *web* tersebut dibuat dengan

menggunakan bahasa seperti PHP, JSP, ASPX, hasil akhirnya yang ditampilkan oleh browser tetap adalah HTML murni.

2. PHP (*Personal Home Page*)

PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang berhasil *open source*. Menurut (Destiana & Hadidah, 2016) “PHP merupakan bahasa berbentuk *script* yang ditempatkan di dalam *server* baru kemudian di proses”. Adapun menurut (Imaniawan & Elsa, 2017) “PHP merupakan bahasa standar yang digunakan dalam dunia *website*, PHP adalah bahasa program yang berbentuk skrip yang diletakkan di dalam *server web*”.

Kemudian hasil pemrosesan dikirimkan kepada web browser klien. Bahasa pemrograman ini dirancang khusus untuk membentuk web dinamis. Artinya, pemrograman PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini.

Secara khusus, PHP dirancang untuk membentuk aplikasi web dinamis. Artinya, ia dapat membentuk satu tampilan berdasarkan permintaan terkini. Misalnya, anda bisa menampilkan isi database ke halaman web. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *Cold Fusion*, ataupun *Perl*. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *common line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

3. CSS (*Cascading Style Sheet*)

Menurut (Imaniawan & Elsa, 2017) “*Cascading Style Sheet* memiliki arti gaya menata halaman bertingkat, yang berarti setiap satu elemen yang telah diformat dan memiliki anak yang telah diformat, maka anak dari elemen tersebut secara otomatis mengikuti format elemen induknya”.

CSS mempunyai 2 bagian utama yaitu *selectors* dan deklarasi. Yang dimaksud *selectors* biasanya element HTML yang ingin diubah, sedangkan deklarasi biasanya terdiri dari properti dan nilai. Properti sendiri adalah atribut *style* yang di ingin diubah, dan setiap properti memiliki nilai.

2.3.1. Implementasi Dan Pengujian Web

Implementasi ini berdasarkan observasi dan studi kasus, yaitu judul TA yang dibuat , dan materi tentang implementasi dan pengujian website yang merupakan materi pembelajaran di website, sebagaimana yang sudah dirancang pada Bab III sebelumnya.

Menurut (Rosa & Salahuddin, 2015:275) mengemukakan “*Blackbox testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”

Pengujian pada *Black Box* berusaha menemukan kesalahan dalam kategori seperti dibawah ini :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *Black Box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada

perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.