

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Web

Dalam pembuatan Tugas Akhir ini membutuhkan teori yang mendukung dalam mempelajari maupun merancang *Web*. Menurut (Rerung, 2018) dalam buku “Pemrograman Web Dasar” menyatakan bahwa *Web* adalah jaringan computer yang terdiri dari kumpulan situs internet yang menawarkan teks dan grafik dan suara dan sumber daya animasi melalui *hypertext transfer protocol*.

2.1.1. Website

Website pada awalnya adalah ruang informasi dalam internet dengan menggunakan teknologi *hiperteks*, pemakai dituntun untuk menemukan informasi dengan mengikuti link yang disediakan dalam dokumen *web* yang ditampilkan dalam *Browser web*.

1. Website

Menurut Marisa (2017:2) mengemukakan bahwa:

Website dapat diartikan sebagai sekumpulan halaman-halaman yang digunakan untuk mempublikasikan informasi berupa teks, gambar, dan program multimedia lainnya, berupa animasi (gambar gerak, tulisan gerak), suara, dan atau gabungan dari semuanya itu baik yang bersifat statis atau dinamis yang membentuk satu rangkaian bangunan yang terkait antara satu halaman dengan halaman yang lain yang sering disebut dengan *hyperlink*.

Menurut Abdulloh (2016:1) menjelaskan bahwa, “*Website* atau disingkat dengan *web*, dapat diartikan sekumpulan halaman yang terdiri atas beberapa halaman yang

berisi informasi dalam bentuk data digital, baik berupa teks, gambar, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet”.

2. *Internet*

Menurut Kustiyahningsih dan Namisa dalam (F. Sidik, 2019) mengemukakan bahwa “internet adalah sebuah jaringan komputer dunia, semua berbicara dengan bahasa yang sama. Banyak keuntungan yang didapat dari jaringan komputer, diantaranya produktifitas dan efisien”.

3. *Web server*

Menurut Hadi (2016:161) “Pengertian *Web Server* bisa merujuk pada *Hard Ware* ataupun *Software* yang menyediakan layanan akses kepada *user* melalui protokol komunikasi HTTP atau HTTPS atas berkas-berkas yang terdapat pada suatu situs *Web* menggunakan aplikasi tertentu seperti *web browser*”.

Menurut Solichin (2016:7) Menyatakan bahwa:
Web Server merupakan sebuah perangkat lunak dalam server yang berfungsi menerima permintaan (*request*) berupa halaman *web* melalui HTTP atau HTTPS dan *klien* yang dikenal dengan *Browser Web* dan mengirimkan kembali (*response*) hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk halaman HTML.

4. *Web Browser*

Menurut Abdulloh (2016:5) yaitu “*Web Browser* digunakan untuk menampilkan hasil website yang telah dibuat. *Web Browser* yang paling sering digunakan diantaranya Mozilla Firefox, Google Chrome, Internet Explorer, Opera, dan Safari”.

Menurut Sibero (2013:12) “*Web Browser* adalah aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi *web*”.

5. HTTP

Menurut Rerung (2018:8) mengungkapkan bahwa: “*Hyper Teks Transfer Protocol* (HTTP) adalah protokol yang dipergunakan untuk mentransfer dokumen dalam bentuk WWW. Sejarah protokol HTTP pertama kali dipergunakan dalam WWW pada tahun 1990”.

2.1.2. Bahasa Pemrograman

Menurut Abdullah (2016:1) menyatakan bahwa: Bahasa pemrograman merupakan Bahasa yang dapat di pahami oleh komputer. Ada banyak Bahasa pemrograman yang memiliki fungsi yang berbeda-beda, diantaranya Bahasa pemrograman untuk membuat aplikasi desktop, membuat game, membuat aplikasi *Web*, membuat aplikasi Handphone dan sebagainya. Dalam buku ini ada 5 macam Bahasa pemrograman yang memiliki peran sendiri-sendiri dalam membangun sebuah *website*. Kelima Bahasa tersebut yaitu JQuery, HTML, CSS, Javascript, dan PHP.

Dalam Membuat *Website* ini penulis menggunakan bahasa pemrograman yang masing-masing memiliki peran dalam membangun sebuah website diantaranya yaitu:

1. PHP

Menurut (B. Sidik, 2014) Menyimpulkan bahwa: PHP Merupakan secara umum dikenal sebagai bahasa pemrograman scrip-scrip yang membuat dokumen HTML secara *on the fly* yang di eksekusi di *server web*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*.

Menurut Abdullah (2016:3) Memberikan batasan bahwa, “ PHP singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programming*, yaitu bahasa pemrograman yang diproses disisi *server*, fungsi utama PHP dalam membangun *website* adalah untuk melakukan pengolahan data pada database. Jadi penulis mengambil kesimpulan bahwa PHP dalam *website* yaitu untuk pengolahan data pada *website* dengan bahasa-bahasa pemrograman.

2. HTML

Menurut Abdulloh (2016:2) menyatakan bahwa “ HTML singkatan dari *Hyper Text Markup Language*, yaitu scrip yang berupa tag-tag untuk membuat dan mengatur struktur *website*. Beberapa tugas utama HTML dalam membangun *website*, diantaranya sebagai berikut :

- a. Menentukan layout *website*
- b. Memformat teks dasar, seperti pengaturan paragraph dan format font.
- c. Membuat list.
- d. Membuat tabel.
- e. Menyisipkan gambar, video, dan audio.
- f. Membuat *link*.
- g. Membuat formulir.

3. Javascript

Menurut (Abdulloh, 2016) memberikan batasan bahwa “Peran *Javascript* dalam membuat *website* adalah memberikan efek animasi yang menarik dan interaktivitas dalam penanganan *event* yang dilakukan oleh pengguna *website*”. Sedangkan menurut (Sulistiono, 2018) menyatakan bahwa, “*Javascript* adalah bahasa pemrograman tingkat tinggi untuk membuat *website* dinamis”.

Jadi penulis menyimpulkan bahwa *Javascript* berguna untuk membuat *website* yang menarik dan dinamis.

4. CSS (*Cascoding Style Sheet*)

Menurut (F. Sidik, 2019) mengemukakan bahwa “CSS adalah kumpulan kode-kode yang berurutan dan saling berhubungan untuk mengatur format/tampilan suatu halaman HTML. CSS dapat dituliskan pada bagian `<body>`, `<head>` suatu dokumen HTML atau diletakkan disebuah file eksternal”.

Ada dua acara yang bisa diterapkan untuk menggunakan CSS pada *web*. Cara yang pertama dengan membuat CSS langsung di dalam suatu *file* HTML (*Internal/inline style sheet*). Cara yang kedua dengan memanggil CSS tersebut dari *file* CSS tersendiri (*external style sheet*).

5. *Jquery*

Menurut Sulistiono (2018:4) menyatakan bahwa “*JQuery* sama dengan *Javascript Library*, yaitu kumpulan kode atau fungsi *javascript* siap pakai sehingga mempermudah dan mempercepat kita dalam membuat kode *Javascript*”.

2.1.3. Basis Data

Menurut Winarno dan Utomo dalam Prayitno (2015), bahwa pengertian *database* “biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam table-tabel yang saling berhubungan satu sama lain, dengan menggunakan *field*/kolom pada tiap tabel yang ada”.

Menurut Soekamto & Salahuddin (2018:43) “Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

1. Beberapa keuntungan dan kerugian penggunaan basis data menurut Lubis (2016:8), keuntungannya terhadap perusahaan antara lain :
 - a. Terkontrolnya kerangkapan data dan *inkonsistensi*.
 - b. Terpeliharanya keselarasan data.
 - c. Data dapat dipakai secara bersama-sama.
 - d. Memudahkan penerapan standarisasi.
 - e. Memudahkan pencapaian Batasan-batasan pengamanan.
 - f. Terpeliharanya *integrasi* data.

- g. Terpeliharanya keseimbangan atas perbedaan kebutuhan data dari setiap aplikasi.
 - h. Program atau data *independent*.
2. Kerugian penggunaan basis data pada sebuah perusahaan, antar lain :
- a. Mahal dalam implementasinya.
 - b. Rumit atau kompleks.
 - c. Penanganan proses *recovery dan back up* sulit.
 - d. Kerusakan pada sistem basis data dapat mempengaruhi departemen yang terkait.

2.1.4. Penyewaan

Menurut (Rosiska & Nopiana, 2017) mengungkapkan bahwa: “Penyewaan merupakan pemindahan hak guna pakai dari hak pemilik kepada pihak penyewa dalam jangka waktu tertentu, dan tarif yang telah ditentukan”.

2.2. Peralatan Pendukung

Dalam pembuatan *Website* ini juga tidak hanya memerlukan teori konsep dasar program saja, namun diperlukan peralatan pendukung untuk memahami isi dari tulisan.

2.2.1. UML (*Unified Modeling Language*)

Menurut Sugiarti (2018:101), *Unified Modeling Language* adalah sebuah bahasa yang telah mendjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak.

Menurut Sugiarti dalam (Sugiarti, 2018) menyatakan bahwa “ UML adalah bahasa grafis yang kompleks dan kaya dengan fitur, salah satu model diagramnya adalah Activity Diagram.

1. *Activity Diagram*

Menurut Sugiarti (2018:136) *Activity Diagram* mendeskripsikan aliran kerja dan perilaku sistem, diagram ini hampir sama dengan pdiagram status karena diagram-diagramnya merupakan status suatu pekerjaan.

2. *Use Case Diagram*

Menurut (Sugiarti, 2018) Diagram *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih actor dalam sistem yang akan dibuat. Sedangkan menurut Rosa & Shalahuddin (2014:155) menjelaskan “Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami”.

Penulis menyimpulkan *use case* merupakan gambaran dari suatu sistem atau dapat diartikan *use case* adalah uraian pelaksanaan suatu sistem yang berisi alur penggunaan dan siapa saja yang dapat menggunakan fungsi di dalamnya.

3. *Squence Diagram*

Menurut Rachmaniah (2018:82) menyatakan bahwa: *Squence Diagram* memperlihatkan bagaimana objek berinteraksi satu dengan yang lainnya disertai urutan terjadinya interaksi tersebut serta difokuskan pada *message Interchange* antara *lifeline* (Objek), penting diperhatikan bahwa *sequence diagram* menunjukkan interaksi yang berlangsung pada suatu *scenario* tertentu, proses direpresentasikan secara *vertical*, sedangkan interaksi dinyatakan dalam tanda panah.

4. *Deployment Diagram*

Menurut Rachmaniah (2018:58) menyatakan bahwa “*Deployment Diagram* menunjukkan arsitektur fisik dari sistem saat desain fisik ke implementasi, diagram ini

dapat digunakan untuk menunjukkan komponen perangkat lunak yang akan di *deploy* ke arsitektur fisik”.

5. *Class Diagram*

Menurut Sugiarti (2018:120) menyatakan bahwa “Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian *class-class* yang akan dibuat untuk membangun sistem”. Sedangkan menurut (Rachmaniah, 2018) menyimpulkan “*Class Diagram* adalah elemen mendasar dari setiap solusi yang berbasis objek”.

2.2.2. Struktur Navigasi

Menurut Anwar dan Irawan (2017:114) menjelaskan bahwa: “ Struktur Navigasi adalah gabungan dari struktur referensi informasi situs *web* dan mekanisme *link* yang mendukung pengunjung untuk melakukan pejelajahan situs”.

Ada 4 macam bentuk dasar dari peta navigasi yang biasa digunakan dalam proses pembuatan aplikasi multimedia, yaitu:

1. Struktur Navigasi *Linear*

Struktur navigasi *linear* merupakan struktur yang hanya mempunyai satu rangkaian cerita yang berurut untuk menampilkan satu demi satu tampilan layer secara berurut menurut urutannya dan tidak diperbolehkan adanya percabangan. Tampilan yang dapat ditampilkan adalah salah satu halaman sebelumnya atau salah satu halaman sesudahnya.

2. Struktur navigasi Hirarki

Struktur navigasi hirarki merupakan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai *Master Page* (halaman utama kesatu), halaman utama ini mempunyai halaman

percabangan yang dikatakan *slave page* (halama 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*.

3. Struktur Navigasi *Non Linier*

Struktur navigasi *non linear* merupakan pengembangan dari struktur *linier*. Pada struktur ini diperkenankan membuat penjejakan bercabang. Pemakai bebas menelusuri *website* tanpa dibatasi oleh suatu rute. Percabangan yang dibuat pada struktur *non linier* ini berbeda dengan percabangan hirarki, karena pada percabangan *non linier* ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama tidak ada *Master Page dan Slave Page*.

4. Struktur Navigasi Campuran (*Composite*)

Struktur navigasi campuran (*composite*) merupakan gabungan dari struktur navigasi sebelumnya dan disebut juga struktur navigasi bebas, maksudnya adalah jika suatu tampilan membutuhkan percabangan maka dibuat percabangan. Struktur ini paling banyak digunakan dalam aplikasi multimedia.

2.2.3. XAMPP

Menurut Nugroho dalam (Tujni & Megawaty, 2017) mengatakan bahwa: “XAMPP adalah suatu bundel web server yang populer digunakan untuk coba-coba di *windows* karena kemudahan instalasinya. Bundel Program Open Source tersebut berisi *antara lain server web Apache*, interpreter PHP, dan basis data MySQL”.

2.2.4. PHP *MyAdmin*

Menurut Nugroho dalam (Tujni & Megawaty, 2017) mengungkapkan bahwa: “PHPMyAdmin adalah aplikasi *web* yang dibuat dari pemrograman PHP dan diramu dengan JavaScript. PHPMyAdmin juga dapat disebut sebagai *tools* yang berguna untuk mengakses yang ada *database MySQL server* dalam bentuk tampilan *web*”.

2.2.5. MySQL

Menurut Kadir dalam (2018:170) menyatakan bahwa: “MySQL (baca: :Mai-es-kyu-el”) merupakan sistem manajemen *database* terkenal yang sekarang dimiliki oleh *Oracle* dan salah satu produknya yang bernama *MSQL community server* bersifat *Open Source*”.

Sedangkan menurut Nugroho dalam (Tujni & Megawaty, 2017) mengungkapkan bahwa: “MySQL merupakan *database* yang berbasis server. Anda bisa menggunakan *Database MySQL*. apabila memiliki izin hak akses didalamnya. Hal ini seperti halnya pada saat anda hendak menggunakan klien MySQL untuk masuk pada server MySQL”.

Keunggulan dari MySQL adalah :

1. Bersifat open source.
2. Sistem software-nya tidak memberatkan kerja server atau komputer karena dapat bekerja di background.

2.2.6. ERD (*Entity Relationship Diagram*)

Menurut Rosidin dan Lubis dalam (F. Sidik, 2019) mengemukakan bahwa “*Entity Relationship Diagram* merupakan sebuah diagram yang digunakan untuk merancang hubungan antar tabel-tabel dalam basis data”.

Adapun Komponen-komponen ERD adalah sebagai berikut :

1. Entitas

Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. sederhananya entitas menunjuk pada individu suatu objek, sedang himpunan entitas menunjuk pada rumpun (*family*) dari individu tersebut.

Dilihat dari jenisnya entitas terbagi atas dua yaitu :

a. Entitas Kuat (*Strong Entity*)

Entitas kuat adalah entitas yang dapat berdiri sendiri tidak bergantung pada entitas lainnya, entitas kuat memiliki atribut *key* dan entitas kuat digambarkan sebagai kotak persegi panjang bergaris tunggal. Contoh entitas kuat adalah entitas pegawai.

b. Entitas Lemah (*Weak Entity*)

Entitas lemah adalah entitas yang tidak dapat berdiri sendiri. Entitas lemah merupakan hasil dari pembentukan entitas kuat, entitas lemah tidak memiliki atribut *key* dan entitas lemah digambarkan sebagai kotak persegi panjang bergaris ganda. Jika entitas kuat yang membentuk entitas lemah dihapus maka secara otomatis entitas lemah akan terhapus. Contoh entitas lemah adalah entitas pegawai kontrak pegawai tetap.

2. Atribut

Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik (*Property*) dari entitas tersebut. Penentuan atau pemilihan atribut-atribut yang relevan bagi sebuah entitas merupakan hal yang penting lainnya dalam pembentukan model data.

3. Relasi

Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

4. Garis

Sebagai penghubung antara himpunan relasi dengan himpunan entitas dengan himpunan atributnya.

Adapun Kardinalitas Relasi pada ERD adalah sebagai berikut:

a. Satu ke satu (*One to One*)

Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B =, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan entitas pada himpunan entitas A.

b. Satu ke banyak (*One to Many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak satu entitas pada himpunan B.

c. Banyak ke banyak (*Many to Many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

2.2.7. *Logical Record Structure (LRS)*

Menurut Andriansyah dalam (Nuraini, 2018) mengemukakan bahwa, “Logical Relationship Structure (LRS) merupakan transformasi dari penggambaran ERD dalam bentuk yang lebih jelas dan mudah untuk di pahami. Penggambaran LRS hampir mirip dengan penggambaran normalisasi file, hanya saja tidak digambarkan symbol asterisk (*) sebagai simbol *primary key* (kunci utama) dan *foreign key* (kunci tamu).

2.2.8. Implementasi dan Pengujian Web

Pengujian unit focus pada usaha verifikasi pada unit terkecil pada desain perangkat lunak (komponen atau modul perangkat lunak). Setiap unit perangkat lunak diuji agar dapat diperiksa apakah aliran masukan (*input*) dan keluaran (*output*) dari unit sudah sesuai dengan yang diinginkan. Pengujian unit biasanya dilakukan saat kode program dibuat. Karena dalam sebuah perangkat lunak banyak memiliki unit-unit kecil maka untuk menguji unit-unit kecil ini biasanya dibuat program kecil (*main program*) untuk mengkaji unit-unit perangkat lunak.

Menurut Sukamto dan M. Shalaludin dalam (Nuraini, 2018) mengatakan bahwa, “*Black-Box Testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk

mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah :

1. Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.



Abdulloh, R. (2016). *Easy and Simple Web Programming*. Jakarta: PT. Elex Media Komputindo.

Hadi, A. (2016). *Administrasi Jaringan Komputer*. Jakarta: Kencana.

Marisa, F. (2017). *WEB PROGRAMMING (CLIENT SIDE AND SERVER SIDE)* (1 ed.). Yogyakarta: Grup Penerbit CV BUDI UTAMA.

Nuraini, R. (2018). Sistem Informasi Penyewaan Kamar Hotel Berbasis Web. *Pemrograman Web*. Diambil dari

<https://repository.bsi.ac.id/index.php/repo/viewitem/17582>

Rachmaniah, M. (2018). *Pengembangan Perangkat Lunak dan Sistem Informasi* (1 ed.). Bogor: IPB Press.

Rerung, R. R. (2018). *Pemrograman Web Dasar* (1 ed.). Yogyakarta: Grup Penerbitan CV BUDI UTAMA.

Rosa, & Shalahuddin, M. (2014). *Rekayasa Perangkat Lunak* (Cetakan ke). Bandung: Informatika Bandung.

Rosiska, E., & Nopiana, P. R. (2017). *Aplikasi Sistem Informasi Akuntansi Jasa Penyewaan Kapal Berbasis Web Pada PT. Sekumbang Permata Engineering*. 134–144.

Sidik, B. (2014). *PEMROGRAMAN WEB PHP* (Edisi Revi). BANDUNG: Informatika Bandung.

Sidik, F. (2019). *Rancang Bangun Sistem Informasi Penjualan Box Motor Berbasis Web*. 3, 8–14.

Solichin, A. (2016). *Pemrograman Web dengan PHP dan MySQL*. Jakarta:

Universitas Budi Luhur.

Sugiarti, Y. (2018). *DASAR-DASAR PEMROGRAMAN JAVA NETBEANS DATABASE, UML, DAN INTERFACE* (1 ed.; N. Nur M, ed.). Bandung.

Sulistiono, H. (2018). *Coding Mudah dengan Codegniter, JQuery, Bootsrap, dan Datatable*. Jakarta: PT. Elex Media Komputindo.

Tujni, B., & Megawaty. (2017). *PELATIHAN PEMBUATAN WEB DENGAN PHP PADA SMP NEGERI 27*. 1(1), 37–40.

