

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Website**

Menurut Rudianto (2011) *website* atau disingkat dengan *web* merupakan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk digital baik itu teks, gambar, animasi yang disediakan melalui jalur internet sehingga dapat diakses dari seluruh dunia. *Website* dapat dibagi menjadi 2 jenis, yaitu :

1. *Website Statis :*

Merupakan *web* yang halamannya tidak berubah, biasanya untuk melakukan perubahan yang dilakukan secara manual dengan mengubah kode. *Website* statis informasinya merupakan informasi satu arah, yakni hanya berasal dari pemilik *software*-nya saja. Hanya bisa di perbarui oleh pemiliknya saja. Contoh *website* statis ini yaitu profil perusahaan.

2. *Website Dinamis :*

Merupakan *web* yang halaman selalu update, biasanya terdapat halaman *back end* (halaman administrator) yang digunakan untuk menambah atau mengubah konten. *Web* dinamis membutuhkan *database* untuk menyimpan. *Website* dinamis mempunyai arus informasi dua arah, yakni berasal dari pengguna dan pemilik. Sehingga pembaruan dapat dilakukan oleh pengguna dan juga pemilik *website*[4]

#### **2.2 User Interface (UI)**

*User interface* atau antarmuka pengguna merupakan penghubung interaksi antara manusia dan *system*, yang artinya harus merefleksikan kapabilitas seseorang dan kebutuhannya, *user interface* harus berguna sehingga dapat mencapai tujuan pengguna, *user interface* harus mudah dipelajari sehingga sistem mudah digunakan dan tidak membuat frustrasi. *User interface* yang baik adalah yang memungkinkan pengguna untuk fokus pada informasi dan tugas yang dikerjakan, bukan mekanisme yang digunakan untuk menyajikan informasi dan melakukan tugas[5][6].

## 1. Paradigma Interaksi

Terdapat tiga paradigma dominan dalam perancangan konseptual dan visual sebuah antarmuka yaitu: berpusat-pada-implementasi (BPI), metaforik dan idiomatik.

### a. Antarmuka Berpusat Pada Implementasi

Antarmuka BPI sangat banyak dijumpai di kalangan industri komputer. Antarmuka ini diekspresikan dalam bentuk atau cara mereka dibentuk. Tingkat keberhasilan penggunaannya bergantung kepada seberapa jauh pengguna memahami cara bekerjanya program. Perancangan antarmuka BPI mengharuskan perancang untuk secara eksklusif berfokus pada model implementasinya[5].

### b. Antarmuka Metaforik

Antarmuka metaforik bergantung kepada hubungan intuitif yang dibuat pengguna pada saat mereka melihat simbol visual dari suatu komponen antarmuka dengan fungsinya. Dalam hal ini tidak ada keharusan bagi pengguna untuk memahami mekanisme suatu program, sehingga paradigma ini merupakan satu langkah maju dari BPI, hanya saja daya tarik dan manfaatnya dibesar-besarkan secara tidak proporsional.

Metafor dalam konteks antarmuka dan perancangan interaksi berarti metafor visual: gambar yang digunakan untuk menyajikan suatu maksud atau atribut tertentu dari sebuah benda. Pengguna mengenali suatu gambaran metafor dan memahami maksudnya. Metafor sangat variative, mulai dari gambar sederhana pada sebuah tombol toolbar sampai ke keseluruhan layar, dari simbol gunting yang berarti cut sampai ke buku cek berukuran besar [5].

### c. Antarmuka Idiomatik

Perancangan idiomatik didasarkan pada cara pengguna belajar sesuatu dan menggunakan berbagai idiom – misalnya polisi tidur, jago merah atau ringan tangan. Antarmuka idiomatik memecahkan persoalan-persoalan yang dimiliki oleh kedua paradigma di atas dengan tidak berfokus kepada pengetahuan teknis atau intuisi tentang bagaimana suatu benda berfungsi, tetapi berfokus pada idiom visual dan perilaku sederhana untuk menyelesaikan suatu tugas.

Kebanyakan elemen antarmuka grafis yang intuitif sesungguhnya merupakan idiom visual. Jendela, papan judul, penutup kotak, pemisah layar,

*hyperlink*, *drop-down* dan lain-lain, merupakan benda-benda yang kita pelajari secara idiomatik bukan secara metaforik.

Menurut Cooper and Reimann (2003), perancangan idiomatik merupakan masa depan perancangan antarmuka. Dalam paradigma ini, pengguna bergantung kepada kemampuan alamiah manusia untuk memahami tentang “bagaimana” dan “mengapa”. Jumlah idiom yang dapat digunakan bisa dibilang tak terbatas dibandingkan dengan jumlah metafor yang lebih terbatas. Metafor memberikan sedikit manfaat dibanding beban tambahan dikemudian hari[5].

## **2. Pentingnya Desain *User Interface* yang Baik**

Desain UI yang baik sangat penting untuk sebuah perangkat lunak, karena jika sebuah perangkat lunak memiliki antarmuka yang buruk akan mengakibatkan hal buruk juga kepada penggunanya. Berikut ini adalah beberapa alasan kenapa desain UI yang baik itu sangat penting:

- a. Banyak sistem dengan fungsionalitas yang baik tapi tidak efisien, membingungkan dan tidak berguna karena desain UI yang buruk.
- b. Antarmuka yang baik merupakan jendela untuk melihat kemampuan sistem serta jembatan bagi kemampuan perangkat lunak.
- c. Desain yang buruk akan membingungkan, tidak efisien, bahkan menyebabkan frustrasi.

Berikut adalah hasil penelitian kenapa desain UI yang baik itu sangat penting:

- a. Pengguna bekerja 20% lebih produktif dengan layar yang sederhana.
- b. Pengguna layar yang dimodifikasi menyelesaikan transaksi 25% lebih cepat dan error berkurang 25% dari sebelumnya.
- c. *Window* yang didesain dengan efektif menghemat \$20.000 dalam 1 tahun.

Fungsi *searching* yang diperbaiki dapat meningkatkan *success rate* hingga 15% dan waktu pencarian 50% lebih cepat.

## **3. Komponen *User Interface***

Antarmuka pengguna atau *user interface* (UI) terdiri dari empat komponen sebagai berikut:

1. Model Pengguna

Model pengguna merupakan model konseptual yang diinginkan pengguna dalam memanipulasi informasi dan proses yang akan dibuat dalam aplikasinya.

2. Bahasa Perintah

Bahasa perintah merupakan bahasa sebagai piranti yang digunakan untuk memanipulasi model.

3. Umpan Balik

Umpan balik merupakan kemampuan program untuk membantu pengguna dalam mengoperasikan program yang telah dihasilkan.

4. Penampilan Informasi

Penampilan informasi merupakan petunjuk status informasi atau program ketika pengguna melakukan tindakan, perlu dirancang pesan yang efektif untuk membantu pengguna dalam menggunakan aplikasi.

- 4. Prinsip Umum Perancangan *User Interface***

Dalam melakukan perancangan *user interface* tentunya ada prinsip-prinsip yang dapat digunakan sebagai acuan agar memberikan *user interface* sesuai dengan kebutuhan pengguna. Berikut adalah prinsip-prinsip umum dalam perancangan sebuah *user interface* [5]:

1. *User Compability*

Produk yang dibangun harus disesuaikan dengan kebutuhan pengguna, karena semua pengguna tidak sama dan pengguna bukanlah seorang *developer* ataupun *designer*. Seorang perancang juga harus memahami psikologi dari calon pengguna perangkat lunak seperti karakteristik dan sifat pengguna.

2. *Product Compability*

Produk yang dihasilkan harus memiliki kompatibilitas antar produk harus diperhatikan dan dipertahankan. Ada kemungkinan antarmuka yang dibangun memiliki beberapa perbedaan dengan proses yang dilakukan secara manual.

3. *Task Compability*

Dalam perancangan sistem, struktur dan aliran sistem harus sesuai dan mendukung tugas yang dimiliki oleh pengguna sehingga dapat membantu pengguna dalam menyelesaikan tugas dan mencapai tujuan mereka.

4. *Work Flow Compability*

*Work Flow Compability* merupakan pengorganisasian sistem yang harus dilakukan secara baik sehingga dapat memfasilitasi transisi antar tugas pengguna.

#### 5. *Consistency*

Dalam perancangan sistem, antarmuka yang dibangun harus konsisten dan sesuai dengan sistem nyata serta sesuai dengan produk yang dihasilkan sebelumnya.

#### 6. *Familiarity*

Perancangan antarmuka sebaiknya disesuaikan dengan antarmuka pada umumnya serta familiar terhadap pengguna. Perancangan meliputi tata letak, simbol, bahasa dan lain sebagainya sehingga pengguna akan dengan mudah menggunakan dan mengenali antarmuka tersebut.

#### 7. *Simplicity*

Kesalahan umum yang sering terjadi pada perancangan antarmuka adalah berusaha untuk menyediakan semua fungsionalitas. Hal ini mengakibatkan antarmuka sulit untuk digunakan dan tidak sesuai dengan kebutuhan pengguna.

#### 8. *Direct Manipulation*

*Direct manipulation* dimaksudkan agar pengguna dapat melihat aksinya secara langsung pada objek yang terlihat.

#### 9. *Control*

Jangan biarkan pengguna merasa di kontrol oleh mesin, karena hal ini dapat menyebabkan frustrasi dan demoralisasi bagi pengguna.

#### 10. *What You See Is What You Get (WYSIWYG)*

WYSIWYG yang berarti apa yang pengguna lihat adalah apa yang pengguna akan dapatkan. Contohnya, saat pengguna mencetak sebuah dokumen di printer maka keluaran yang didapatkan oleh pengguna akan sama persis dengan apa yang dilihatnya di komputer.

#### 11. *Flexibility*

*Flexibility* adalah memberikan pengguna banyak kontrol dan mengkoordinir keahlian pengguna yang bervariasi.

#### 12. *Responsiveness*

*Responsiveness* memungkinkan sistem untuk memberikan respon dengan segera setiap masukan dari pengguna.

#### 13. *Invisible Technology*

Pengguna sebaiknya mengetahui sedikit mungkin detail teknis bagaimana sistem diimplementasikan. Sehingga pengguna tidak memiliki rasa khawatir dan ketakutan menggunakan aplikasinya.

#### 14. *Robustness*

Sistem harus mampu mentolerir kesalahan manusia baik disengaja maupun tidak disengaja dan yang umumnya tidak dapat dihindari. Penggunaan bahasa dalam mentolerir kesalahan juga harus diperhatikan jangan sampai bahasa yang digunakan seperti membentak atau menyalahkan pengguna.

#### 15. *Protection*

Sistem yang dirancang harus dapat memproteksi dari hasil-hasil yang menyebabkan “bencana” karena kesalahan umum yang dibuat oleh pengguna.

#### 16. *Easy of Learning*

Sistem yang dibangun harus mudah dipelajari sehingga pengguna tidak harus diberikan pelatihan penggunaan sistem.

#### 17. *Easy of Use*

Sistem yang dibangun harus mudah digunakan baik bagi pengguna pemula maupun pengguna yang sudah berpengalaman.

### **5. Waktu Tanggap**

Secara umum pengguna menginginkan bahwa program aplikasinya dapat memberikan waktu tanggap yang sependek-pendeknya. Waktu tanggap yang berbeda-beda dapat mempengaruhi konsentrasi pengguna yang pada gilirannya akan mempengaruhi kinerja pengguna [5].

Waktu tanggap yang lama, lebih dari 14 detik, akan menyebabkan perhatian pengguna terpecah ke aktivitas lain, sehingga pengguna cenderung untuk melakukan aktivitas lain sampai sistem menyelesaikan apa yang harus ia kerjakan.

### **6. Penanganan Kesalahan**

Dalam dunia komputer kesalahan kecil dapat berakibat fatal sehingga harus diupayakan agar kesalahan-kesalahan yang mungkin dilakukan dapat dihindari atau ditangani dengan cara-cara tertentu agar tidak mengakibatkan terhentinya eksekusi program aplikasi. Kesalahan dibagi menjadi dua, yakni kesalahan sintaksi pada saat penulisan program dan kesalahan logika yang terjadi pada saat program sedang dijalankan.

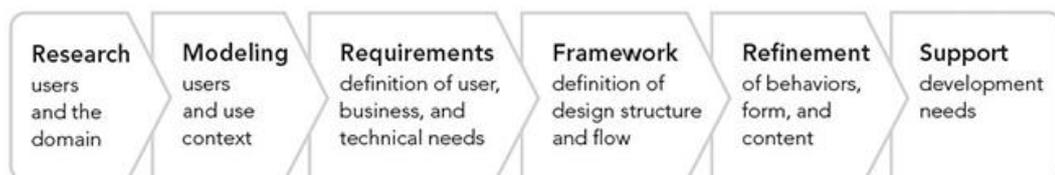
### 2.3 *User Experience (UX)*

*User experience* adalah persepsi seseorang dan responnya dari penggunaan pada sebuah *system*, produk, atau jasa. *User Experience (UX)* menilai seberapa kepuasan dan kenyamanan seseorang pengguna terhadap sebuah produk, sistem, dan jasa. Sebuah prinsip dalam membangun UX yang baik adalah khalayak mempunyai kekuasaan dalam menentukan tingkat kepuasan sendiri (*customer rule*). Seberapa pun bagus fitur sebuah produk, sistem, atau jasa, tanpa khalayak yang dituju dapat merasakan kepuasan, kaidah, dan kenyamanan dalam berinteraksi maka tingkat UX akan menjadi rendah. Perkembangan dunia *digital* dan *mobile* menjadikan UX menjadi lebih kompleks dan multidimensi. Kini seseorang dapat mengakses sebuah situs *web* dari berbagai perangkat digital.

Perancangan UX pun mengalami perkembangan, karena nyaman dalam sebuah piranti akan berbeda dengan piranti lainnya. Mengakses *website* dari perangkat komputer desktop akan berbeda dengan mengakses *website* yang sama lewat perangkat *smartphone*. Begitu juga media digital yang ditampilkan semakin beragam dengan hadirnya *social media*. Namun dalam hal penyampaian konten sebuah *brand* harus mengeluarkan satu bahasa yang sama di berbagai *channel* dan media. Untuk itu konsep UX perlu dipelajari lebih lanjut agar komunikasi *brand* tetap fokus dan solid[7].

### 2.4 *GDD (Goal Directed Design)*

adalah sebuah metode untuk membuat perancangan *user interface* aplikasi yang berfokus pada tujuan pengguna sehingga pengguna merasa puas dan senang dengan desain yg dibuat. *Metode goal directed design* menyediakan solusi yang memenuhi kebutuhan dan tujuan dari pengguna. Proses GDD dibagi menjadi enam tahap yaitu *Research*, *Modeling*, *Requirement Definition*, *Framework Definition*, *Refinement*, dan *Support*[2][3][8] seperti yang ditunjukkan pada Gambar dibawah ini.



**Gambar 1.1 Goal Directen Design**

Berikut penjelasan mengenai tahapan proses metode *Goal Directed Design*

1. *Research*

Tahap *research* menggunakan Teknik etnografi (observasi, wawancara, kuisioner atau teknik pengumpulan data lain) untuk menghasilkan data kualitatif tentang calon pengguna sesungguhnya dari suatu produk. Salah satu hasil utama dari tahap ini yaitu munculnya serangkaian pola perilaku. Pola ini juga akan mendeskripsikan tujuan dan motivasi pengguna sehingga menghasilkan *user persona* yang tepat untuk aplikasi yang akan dibangun. Pola perilaku dan tujuan dapat digunakan dalam membuat *persona* dalam tahap *modeling*.

2. *Modeling*

Pada tahap ini, pola yang telah didapatkan dari tahap *research* digunakan untuk memodelkan seorang pengguna. *Persona* mendeskripsikan *behavior* (perilaku), *attitude* (sikap), *goals* (tujuan) dan motivasi pengguna yang diamati dan diidentifikasi selama tahap *research*. Semua aspek dari sebuah *persona* dimaksudkan untuk mengetahui apa yang dibutuhkan oleh *user*.

3. *Requirement Definition*

Pada tahap ini menggunakan metode desain berbasis skenario yang mendeskripsikan alur dari task secara detail berdasarkan tujuan dari *persona* yang telah didapatkan pada tahap *modeling*. Tujuan pengguna berfungsi sebagai *filter* untuk *task* dan *sub task* dan sebagai panduan untuk membuat skenario. Tahap ini memaparkan informasi dan kemampuan *persona* yang dibutuhkan untuk mencapai tujuan pengguna. Hasil dari fase ini adalah *requirement definition* yang menyeimbangkan bisnis, kebutuhan pengguna dan teknis dari desain yang diperlukan.

4. *Framework Definition*

Pada tahap ini membuat konsep produk secara menyeluruh, menentukan sebuah kerangka dasar (*wireframe*) untuk *product behavior*, dan desain visual. Fase ini menghasilkan definisi *interaction framework*, konsep desain yang stabil dan menunjukkan struktur yang formal dan logis untuk detail yang akan datang.

5. *Refinement*

Tahap ini sama halnya dengan tahap *Framework Definition*, tetapi dengan peningkatan focus yang lebih detail dan implementasi. Tahap ini berfokus pada

pembangunan detail desain pada setiap komponen atau elemen *user interface*, seperti menentukan ikon, warna, visual style dan elemen visual lain yang sesuai dengan tujuan dan pengalaman pengguna. Hasil akhir dari fase ini yaitu dokumentasi dari desain: spesifikasi bentuk dan perilaku yang ada disajikan dalam bentuk laporan atau persentasi.

#### 6. *Support*

Tahap ini merupakan tahap penerapan desain yang dihasilkan pada tahap *refinement*. Pada tahapan ini, desain *user interface* yang telah dibuat pada tahapan *refinement* diimplementasikan menjadi sebuah model aplikasi *prototype*.

### 2.5 *User Persona*

*User Persona* atau *Persona* adalah model pengguna yang direpresentasikan sebagai manusia individu yang spesifik. *Persona* bukan orang yang sebenarnya tetapi perpaduan langsung dari penelitian dan pengamatan orang nyata. Salah satu alasan *persona* sangat baik digunakan sebagai model pengguna adalah dari personifikasinya. *Persona* memasukan empati ke pada tim perancangan dan pengembangan agar dekat dengan tujuan/*goals* dari pengguna[8].

*User Persona* adalah dokumen yang sangat penting yang harus dibuat untuk menganalisis para pengguna. *User Persona* adalah pondasi untuk dokumentasi pengguna yang memperdalam dan memperluas wawasan dari kepribadian pengguna. *Persona* membuat data lebih dapat dipahami dengan memberikannya sebuah nama dan sebuah wajah, bukan angka-angka dan kata-kata tanpa konteks. *Designer* menggunakan *persona* di setiap tahapan dari proses dan setiap pertanyaan dalam perancangan sebuah desain[9]. *User Persona* yang baik dapat berisi beberapa data seperti dibawah ini[9]:

- a. *Photo & Name*
- b. *Demographics*
- c. *Personality*
- d. *Technological Expertise*
- e. *Platforms*
- f. *Goals (Motivations)*
- g. *Personal Quotes/Mottos*

Setelah *user persona* didapatkan berikutnya adalah mengelompokan *persona* tersebut kedalam beberapa kelompok yaitu: *primary*, *secondary*, *supplemental*, *customer*, *served*, dan *negative* [8].

- a. *Primary Persona*, merupakan target utama dari perancangan desain antarmuka. Sebuah produk bisa hanya memiliki satu *primary persona* per antarmuka, tetapi dimungkinkan untuk beberapa produk memiliki beberapa antarmuka yang berbeda, dan masing-masing ditargetkan pada *primary persona* yang berbeda.
- b. *Secondary Persona*, merupakan *persona* yang hampir sepenuhnya puas dengan semua kebutuhan antarmuka *persona primer*. Dalam *secondary persona* tidak dianjurkan memiliki tiga atau empat lebih *persona*, karena hal ini akan mengakibatkan ruang lingkup membesar dan tidak fokus pada tujuan produk.
- c. *Supplemental Persona*, merupakan *persona* yang tidak termasuk kedalam *primary persona* ataupun *secondary persona*. Kebutuhan dari *persona* ini sudah diwakili oleh kombinasi dari *primary persona* dan *secondary persona*.
- d. *Customer Persona*, merupakan *persona* yang berfokus kepada pelanggan dari produk bukan kepada pengguna akhir.
- e. *Served Persona*, merupakan *persona* yang tidak menggunakan produk, tetapi mereka secara langsung dipengaruhi oleh pengguna produk.
- f. *Negative Persona*, merupakan *persona* yang harus dihindari, dan merupakan kebalikan dari *primary persona*.

## 2.6 Usability Testing

Bauer (2010) memberikan definisi *usability testing* atau uji ketergunaan sebagai berikut, “*Usability testing has traditionally meant testing for efficiency, ease of learning, and the ability to remember how to perform interactive tasks without difficulty or errors.*” Dengan perkataan lain, uji ketergunaan adalah mengukur efisiensi, kemudahan dipelajari, dan kemampuan untuk mengingat bagaimana berinteraksi tanpa kesulitan atau kesalahan[10]. Sejak mulai

dikembangkannya internet para pakar di bidang uji ketergantungan menekankan uji ketergantungan dengan dua hal pokok yaitu :

1. *Ease of learning*

Mengukur ketergunaan dengan membandingkan waktu yang diperlukan pemakai dalam mempelajari sistem komputer yang sama sekali belum dikenalnya untuk melakukan sesuatu, dengan waktu yang diperlukan untuk melakukan hal yang sama dengan cara lain.

2. *Ease of use*

Mengukur jumlah tindakan yang perlu dilakukan untuk menyelesaikan suatu pekerjaan. Sebagai contoh membandingkan jumlah-jumlah klik mouse pada dua desain.

Dari pendapat diatas maka dapat disimpulkan *usability* sangat penting untuk keberlangsungan sebuah *website*. Jika sebuah *website* sulit untuk digunakan maka pengguna akan pergi bahkan tidak akan mengunjungi *website* tersebut. Jika sebuah *homepage* gagal memberikan penjelasan atau informasi sulit di mengerti maka penawaran tidak dapat dilakukan melalui situs, orang-orang tidak akan mengunjungi *website* tersebut. *Website* harus dirancang seekonomis mungkin dengan prinsip *human centred design*, sehingga memudahkan manusia untuk menggunakan *website* tersebut. Ujian ketergunaan situs web merupakan kombinasi dari lima aspek yaitu :

1. *Learnability* adalah kriteria yang mengukur tingkat kemudahan suatu situs untuk dipelajari dan digunakan, khususnya bagi pengguna yang baru pertama kali melihat dan menjelajahi situs tersebut.
2. *Efficiency* adalah kriteria yang mengukur tingkat performansi pengguna ketika menggunakan situs.
3. *Memorability* adalah kriteria kuallitatif yang dapat dilihat melalui kemudahan pengguna dalam menggunakan lagi produk (situs) setelah beberapa saat tidak mengunjungi situs.
4. *Errors* adalah kriteria kuantitatif pada *web usability*, errors menilai situs melalui banyaknya kesalahan yang dilakukan oleh pengguna ketika melakukan tugas skenario yang diberikan.

5. *Satisfaction* adalah pengukuran kualitatif yang dirasakan atau juga ditunjukkan oleh pengguna, ketika sistem atau situs yang digunakan menyenangkan dan juga mudah dipakai akan memberikan kenyamanan kepada pengguna atau juga memberikan kenyamanan kepada pengguna atau juga memberikan rasa puas ketika pengguna dapat menyelesaikan tugas yang diberikan dengan baik.

### 2.6.1 Komponen *Usability Testing*

Sastramihardja (2006) mengemukakan komponen usability testing terdiri dari :

- a. *Learnability*, ditandai dengan tingkat keberhasilan penyelesaian task untuk setiap jenis partisipan dan rasio halaman yang dikunjungi rata-rata hasil yang didapat 90%.
- b. *Efficiency*, kelompok pengguna dalam mengerjakan task cukup beragam, terlihat dari deviasi standar setiap jenis partisipan. Secara keseluruhan proses *navigasi lancer* (mayoritas pengguna tidak pernah melakukan penekanan tombol back).
- c. *Satisfaction*, pengguna ditunjukkan dengan komentar yang positif, misalnya menyarankan peningkatan estetika rancangan[10].

## 2.7 Pengertian WEB

Menurut Hidayatullah (2014:1) mengatakan bahwa “Web dapat diartikan sekumpulan halaman yang berupa laman yang berisi informasi dalam bentuk data digital baik itu berupa text, gambar, animasi, video, dan audio lainnya yang disediakan melalui jalur koneksi internet”.

Halaman *website* biasanya berupa dokumen yang ditulis dalam format *Hyper Text Markup Language* (HTML), yang bisa diakses melalui HTTP.

### 2.7.1 World Wide Web

Menurut Hidayatullah (2014:3) mengatakan bahwa “*World Wide Web* atau WWW adalah suatu program yang ditemukan oleh Tim Bernes-Lee pada tahun

1991”. Awalnya Bernes-Lee hanya ingin menemukan cara untuk menyusun arsiparsip risetnya.

Pada tahun 1989, Bernes-Lee membuat pengajuan untuk proyek pembuatan hiperteks global, kemudian pada bulan Oktober 1990, “*Warning Wera Wanua*” sudah dapat dijalankan dalam lingkungan CERN (Pusat Penelitian Fisika Partikel Eropa). Pada musim panas pada tahun 1991, *World Wide Web* atau WWW secara resmi digunakan secara luas pada jaringan internet

### **2.7.2 Web Browser**

Menurut Kustiyaningsih (2011:8) mengatakan bahwa “*Web browser* adalah *software* yang digunakan untuk menampilkan informasi dari *server* web. *Software* ini kini telah dikembangkan dengan menggunakan *user interface* grafis, sehingga pemakai data melakukan ‘*point and click*’ untuk pindah antar dokumen.

### **2.7.3 Web Server**

Menurut Kustiyaningsih (2011:8) mengatakan bahwa “*Web server* yaitu: “Komputer yang digunakan untuk menyimpan dokumen–dokumen web, komputer ini akan melayani permintaan dokumen web dari kliennya. *Web browser* seperti *explorer* atau *navigator* berkomunikasi melalui jaringan (termasuk jaringan internet) dengan *web server*, menggunakan HTTP. *browser* akan mengirimkan *request* ke *server* untuk meminta dokumen tertentu atau layanan lain yang disediakan oleh *server*. *Server* memberikan dokumen satu layanannya jika tersedia juga dengan menggunakan protocol HTTP”.

## **2.8 Bahasa Pemrograman**

Bahasa pemrograman adalah instruksi standar untuk memerintah komputer. Dengan ini memungkinkan seorang *programmer* dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan ataupun diteruskan. Dalam penelitian ini dalam pembangunan sistem terdapat beberapa bahasa pemrograman yang digunakan antara lain sebagai berikut[11].

### **2.8.1 PHP**

PHP merupakan *script* untuk pemrograman berbasis *web server-side*. Dengan menggunakan PHP maka pengelolaan suatu situs web menjadi lebih

mudah. Proses update data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan *script* PHP[12].

PHP sangatlah mudah dipelajari karena sintaks-sintaks PHP mirip dengan bahasa C, dan Pascal. PHP juga disenangi karena dikembangkan sebagai *web specific language* sehingga menyediakan fungsi-fungsi khusus yang membuat pengembangan suatu web dapat dilakukan dengan mudah. Sebagai bahasa pemrograman web, PHP menyediakan koneksi dengan *database*, protokol, dan lain sebagainya. PHP memiliki beberapa kelebihan sebagai berikut.

- a. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya
- b. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah
- c. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan

Dalam sisi pemahaman, PHP adalah bahasa *script* yang paling mudah karena memiliki referensi yang banyak. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

### 2.8.2 HTML

*Hyper Text Markup Language* (HTML) adalah kumpulan dari berbagai simbol atau *tag* yang dituliskan dalam sebuah file yang dimaksudkan untuk menampilkan halaman pada *web browser*. *Tag* tersebut selalu diawali dengan <contoh> dan diakhiri dengan </contoh> dimana contoh *tag* tersebut biasanya berisi seperti b, i, u, dan seterusnya. Sedangkan *tag* yang tidak perlu di akhir dengan </> antara lain br, *input*, dan lainnya.

HTML bukanlah sebuah bahasa pemrograman pada umumnya, seperti Java, C, C++, visual basic dan sejenisnya, melainkan bahasa *markup* / markah yang ditulis dengan perintah tag-tag atau element yang menaungi (mengapit) konten di

dalamnya yang akan ditampilkan pada sebuah halaman web oleh browser atau HTML interpreter (penerjemah HTML) lainnya[12].

### 2.8.3 CSS

CSS (*Cascading Style Sheet*) adalah suatu teknologi yang digunakan untuk mempermudah pembuatan suatu tampilan sebuah website yang berisi rangkaian instruksi yang menentukan bagaimana suatu text akan tertampil dihalaman web. Perancangan desain text dapat dilakukan dengan mendefinisikan fonts (huruf), color (warna), margins (ukuran), latar belakang (background), ukuran font (font size). Dengan menggunakan CSS dapat mempercepat pembuatan situs sekaligus mempermudah dalam proses editing[13].

## 2.9 Perangkat Lunak Pendukung

Dalam penelitian pembangunan sistem ini terdapat beberapa perangkat lunak pendukung diantaranya sebagai berikut.

### 2.9.1 Atom Text Editor

Atom adalah *text editor* yang bersifat *free source* dan *open source* bagi macOS, Linux, dan Microsoft Windows. Atom dilengkapi dengan plug-in yang ditulis dalam Node.js yang dikembangkan oleh GitHub . Atom merupakan aplikasi desktop yang dibuat memakai teknologi web[14]. berikut adalah keunggulan-keunggulan yang dimiliki atom, adalah :

1. Keunggulan *text editor* satu ini yang pertama adalah Atom merupakan salah satu aplikasi open source alias gratis. Dengan menggunakan Atom Anda tidak akan dituntut untuk membeli lisence product ini dengan tarif tertentu.
2. Keunggulan dari *Text Editor* Atom yang kedua adalah, dengan menggunakan *text editor* Atom Anda sebagai programmer bisa langsung terkoneksi dengan project Anda yang ada di Github. Tentunya jika Anda menginginkannya, Anda sudah harus melakukan instalasi git svn.
3. Yang ketiga, dengan menggunakan *text editor* Atom, Anda bisa dengan mudah melakukan instalasi packages yang Anda butuhkan demi

mendukung performa dan kebutuhan Anda sebagai programmer dalam menulis code.

4. *Text editor* Atom juga menyediakan banyak theme yang dapat Anda gunakan untuk mempercantik text editor kesayangan agar Anda tidak bosan saat menulis dan melihat lautan teks coding.
5. Ada juga fitur *customize styling* yang artinya Anda bebas merubah settingan Atom yang mau Anda pakai, karena pada dasarnya memang atom dibuat menggunakan *electron* dan *javascript*, jadi untuk style sama seperti Anda menulis file *css*.

### **2.9.2 XAMPP**

XAMPP adalah paket perangkat lunak yang tersedia secara bebas yang mengintegrasikan distribusi untuk *web server* Apache, MySQL, PHP dan Perl menjadi satu instalasi mudah. XAMPP juga menginstal phpMyAdmin, aplikasi web yang dapat digunakan untuk mengelola *database* MySQL sehingga dapat menyederhanakan proses instalasi, XAMPP sebagai pengembangan dalam lingkup komputer lokal, sehingga melalui program ini, programmer web dapat menguji aplikasi web yang dikembangkan dan dapat mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet.