

## **BAB II**

### **DASAR TEORI**

#### **2.1 Pengertian *E-Learning***

Istilah *E-Learning* terdiri dari dua kata yaitu : *Elektronik* dan *Learning*. *Electronic* atau biasa disebut elektronik ini merupakan sesuatu yang dibuat dengan sistem elektronika. Sedangkan *Learning* sendiri merupakan pembelajaran atau belajar.

Menurut Horton (2006), *E-Learning* ini merupakan kegiatan pembelajaran yang dilakukan tidak secara tatap muka di dalam kelas yang bisa membuat peserta didik menjadi bosan karena pengajar lebih dominan menyampaikan materi ajar namun kegiatan pembelajaran dilakukan dengan menggunakan teknologi internet dan komputer. Sedangkan menurut Linde (2004) juga berpendapat bahwa *E-Learning* merupakan pembelajaran *Formal* dan *informal* dengan menggunakan media elektronik seperti internet, intranet, *CD-ROM*, *video tape*, *DVD*, *TV*, *Smartphone*, *PDA*, dan sebagainya.

Pada pengertian diatas dapat disimpulkan bahwa sistem pembelajaran elektronik atau biasa disebut *E-Learning (Electronic Learning)* dapat didefinisikan sebagai semua kegiatan pelatihan yang menggunakan media elektronik atau teknologi informasi. Istilah yang sering digunakan untuk menggantikan *E-Learning* adalah *web-based learning*, *online learning*, *computer-based training/E-Learning*, *distance learning*, *computer aided instruction*.

##### **2.1.1 Fungsi dan Manfaat *E-Learning***

Menurut Yaniawati (2010) *E-Learning* memiliki fungsi dalam pembelajaran seperti suplemen (tambahan), komponen (pelengkap) dan substitusi (pengganti).

###### **a. Suplemen (tambahan)**

Peserta didik bisa memilih untuk menggunakan ataupun tidak menggunakan *E-Learning* dalam pembelajaran. Atau dengan kata lain mereka tidak diwajibkan untuk mengakses materi ajar *E-Learning*.

b. Komplemen (pelengkap)

Pengajar dapat melengkapi materi ajar yang belum disampaikan di kelas dengan menggunakan *E-Learning*. Selain itu guru juga dapat melakukan pengayaan atau *reinforcement* dan remedial kepada peserta didik setelah pembelajaran dilakukan di dalam kelas.

c. Substitusi (pengganti)

Model kegiatan pembelajaran konvensional (dilakukan di kelas dan tatap muka secara utuh), dapat digunakan dengan menggunakan internet. Pembelajaran yang dilakukan dengan menggunakan internet bertujuan untuk membuat peserta didik melakukan pembelajaran sesuai dengan waktu, aktivitas dan kondisi mereka (fleksibel).

### 2.1.2 Tipe *E-Learning*

Menurut Rosen (2009) ada dua mode pengiriman pada *E-Learning* antara lain sebagai berikut :

- a. *Synchronous*, dalam pelatihan *synchronous*, murid dan guru bertemu ditentukan waktu untuk sesi yang dipimpin oleh instruktur.
- b. *Asynchronous*, dalam pelatihan *asynchronous*, murid menggunakan materi yang tersedia melalui web yang cukup lengkap untuk digunakan setiap waktu, yang memungkinkan siswa untuk mengakses sesuai kebutuhannya.

### 2.1.3 Karakteristik *E-Learning*

Menurut Rosenberg (2001) karakteristik *E-Learning* bersifat jaringan, yang membuatnya mampu memperbaiki secara cepat, menyimpan atau memunculkan kembali, mendistribusikan, dan sharing pembelajaran dan informasi.

Karakteristik *E-Learning* menurut Nursalam (2008:135) adalah sebagai berikut:

1. Memanfaatkan jasa teknologi elektronik.
2. Memanfaatkan keunggulan komputer (digital media dan *computer networks*).

3. Menggunakan bahan ajar yang bersifat mandiri (*self-learning materials*) kemudian disimpan di komputer, sehingga dapat diakses oleh dosen dan mahasiswa kapan saja dan dimana saja.
4. Memanfaatkan jadwal pembelajaran, kurikulum, hasil kemajuan belajar, dan hal-hal yang berkaitan dengan administrasi pendidikan dapat dilihat setiap saat di komputer.

#### **2.1.4 Manfaat *E-Learning***

Manfaat *E-Learning* menurut Pranoto, dkk adalah (Pranoto, Alvin, dkk, 2009:309) :

- a. Penggunaan *E-Learning* untuk menunjang pelaksanaan proses belajar dapat meningkatkan daya serap mahasiswa atas materi yang diajarkan.
- b. Meningkatkan partisipasi aktif dari mahasiswa.
- c. Meningkatkan partisipasi aktif dari mahasiswa.
- d. Meningkatkan kemampuan belajar mandiri mahasiswa.
- e. Meningkatkan kualitas materi pendidik dan pelatihan.
- f. Meningkatkan kemampuan menampilkan informasi dengan perangkat teknologi informasi, dimana dengan perangkat biasa sulit dilakukan.

Beberapa manfaat diatas dapat disimpulkan bahwa Pertama dari pembelajaran ini sangat Fleksibel. *E-Learning* memberi fleksibilitas dalam memilih waktu dan tempat untuk mengakses perjalanan. Kedua Belajar Mandiri, *E-Learning* memberi kesempatan bagi pembelajar secara mandiri memegang kendali atas keberhasilan belajar. Ketiga Efisiensi Biaya, *E-Learning* memberi efisiensi biaya bagi administrasi penyelenggara, efisiensi penyediaan sarana dan fasilitas fisik untuk belajar dan efisiensi biaya bagi pembelajar adalah biaya transportasi dan akomodasi.

### 2.1.5 Kelebihan dan Kekurangan *E-Learning*

Adapun kelebihan Menurut L. Tjokro Tjokro (2009:187), *E-Learning* memiliki banyak kelebihan yaitu :

- a. Lebih mudah diserap, artinya menggunakan fasilitas multimedia berupa gambar, teks, animasi, suara, video.
- b. Jauh lebih efektif dalam biaya, artinya tidak perlu instruktur, tidak perlu minimum audiensi, bisa dimana saja, bisa kapan saja, murah untuk diperbanyak.
- c. Jauh lebih ringkas, artinya tidak banyak formalitas kelas, langsung pada pokok bahasan, mata pelajaran sesuai kebutuhan.
- d. Tersedia 24 jam/hari–7 hari/minggu, artinya penguasaan materi tergantung pada semangat dan daya serap siswa, bisa dimonitor, bisa diuji dengan *e-test* (L. Tjokro, Sutanto, 2009:140).

Selain kelebihan, tentu ada pula kekurangan *E-Learning* menurut yang diuraikan oleh Nursalam (2008:140) yaitu :

- a. Kurangnya interaksi antara pengajar dan pelajar atau bahkan antar pelajar itu sendiri.
- b. Kecenderungan mengabaikan aspek akademik atau aspek sosial dan sebaliknya membuat tumbuhnya aspek bisnis/komersial.
- c. Proses belajar mengajar cenderung ke arah pelatihan daripada pendidikan.
- d. Berubahnya peran pengajar dari yang semula menguasai teknik pembelajaran konvensional, kini juga dituntut mengetahui teknik pembelajaran yang menggunakan *ICT* (*information, communication, dan technology*).
- e. Tidak semua tempat tersedia fasilitas internet (mungkin hal ini berkaitan dengan masalah tersedianya listrik, telepon, ataupun komputer).
- f. Kurangnya sumber daya manusia yang menguasai internet.
- g. Kurangnya penguasaan bahasa komputer.
- h. Akses pada komputer yang memadai dapat menjadi masalah tersendiri bagi peserta didik.

- i. Peserta didik bisa frustrasi jika mereka tidak bisa mengakses grafik, gambar, dan video karena peralatan yang tidak memadai.
- j. Tersedianya infrastruktur yang bisa dipenuhi.
- k. Informasi dapat bervariasi dalam kualitas dan akurasi sehingga penduan dan fitur pertanyaan diperlukan.
- l. Peserta didik dapat merasa terisolasi.

## **2.2 Website**

*Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

Sebuah situs web adalah sebutan bagi sekelompok halaman web (*web page*), yang umumnya merupakan bagian dari suatu nama domain (*domain name*) atau subdomain di *World Wide Web* (WWW) di internet. WWW terdiri dari seluruh situs web yang tersedia kepada publik. Halaman-halaman sebuah situs web diakses dari sebuah *URL* yang menjadi “akar” (*root*), yang disebut *homepage* (halaman induk; sering diterjemahkan menjadi “beranda”, “halaman muka”), dan biasanya disimpan dalam *server* yang sama.

## **2.3 Pembelajaran Daring**

Menurut undang-undang nomor 20 Tahun 2003 pembelajaran adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar. Pendidik harus memenuhi kualifikasi sesuai dengan tingkatan peserta didik yang diajari, mata pelajaran yang diampu, dan ketentuan yang intruksional lainnya. Disamping itu, pendidik harus menguasai sumber belajar dan media pembelajaran agar tercapai tujuan pembelajaran.

Pembelajaran Daring sangat dikenal di kalangan masyarakat dan akademik dengan istilah pembelajaran *online* (*online learning*). Istilah lain yang sangat umum diketahui adalah pembelajaran jarak jauh (*learning distance*). Pembelajaran Daring

merupakan pembelajaran yang berlangsung di dalam jaringan dimana pengajar dan yang diajar tidak bertatap muka secara langsung. Pembelajaran daring dapat dilakukan dari mana dan kapan saja tergantung pada ketersediaan alat pendukung yang digunakan.

## **2.4 Komponen Aplikasi**

Setiap membuat suatu sistem atau aplikasi pasti ada beberapa komponen pendukungnya, berikut adalah komponen-komponen pendukung pembuatan sistem *E-Learning* ini :

### **2.4.1 Bahasa Pemrograman**

Bahasa pemrograman adalah bahasa yang digunakan untuk menerjemahkan setiap perintah dalam *Website* yang pada saat diakses. Jenis bahasa program sangat menentukan statis, dinamis atau interaktifnya sebuah *Website*. Semakin banyak ragam bahasa program yang digunakan maka akan terlihat *Website* semakin dinamis, dan interaktif serta terlihat bagus. Beragam bahasa program saat ini telah hadir untuk mendukung kualitas *Website*. Jenis jenis bahasa program yang banyak dipakai para desainer *Website* antara lain *HTML*, *ASP*, *PHP*, *JSP*, *Java Scripts*, *Java applets* dsb.

Dalam pembuatan program ini kami menggunakan bahasa pemrograman *HTML (CSS)*, *PHP*, *Java Script*, serta *Framework*. Berikut adalah *Framework* yang kami gunakan dalam pembuatan sistem *E-Learning* ini yaitu *Bootstrap 4.5*, *jQuery 3.5.1*, dan *Laravel*.

#### **1. Hypertext Markup Language (HTML)**

*Hypertext Markup Language (HTML)* adalah Bahasa markah standar untuk dokumen yang dirancang untuk ditampilkan di peramban internet. Ini dibantu oleh teknologi seperti *Cascading Style Sheets (CSS)* dan Bahasa *scripting* seperti *JavaScript* dan *VBScript*.

Peramban internet menerima dokumen *HTML* dari *server web* atau dari penyimpanan lokal dan membuat dokumen menjadi halaman web multimedia. *HTML* menggambarkan struktur halaman web secara *semantic* dan isyarat awal yang disertakan untuk penampilan dokumen.

Elemen *HTML* digambarkan oleh *tag*, ditulis menggunakan tanda kurung sudut. *Tag* seperti `<img />` dan `<input />` langsung memperkenalkan konten ke dalam halaman. *Tag* lain seperti `<p>` mengelilingi dan memberikan informasi tentang teks dokumen dan mungkin menyertakan *tag* lain sebagai sub-elemen. Peramban tidak menampilkan *tag HTML*, tetapi menggunakannya untuk menafsirkan konten halaman.

*HTML* dapat menyematkan program, yang ditulis dalam Bahasa *scripting* seperti *JavaScript*, yang memengaruhi perilaku dan konten halaman web. Dimasukkan *CSS* mendefinisikan tampilan dan tata letak konten.

Sama seperti hal teknis lainnya dalam dunia web, *HTML* juga punya kelebihan dan kekurangannya:

a. Kelebihan *HTML*

1. Bahasa yang digunakan secara luas dan memiliki banyak sumber serta komunitas yang besar
2. Dijalankan secara alami di setiap *web browser*
3. Memiliki *learning curve* yang mudah
4. *Open-source* dan sepenuhnya gratis
5. Bahasa *markup* yang rapi dan konsisten
6. Standard web yang resmi di maintain oleh *World Wide Web Consortium (W3C)*
7. Mudah diintegrasikan dengan bahasa *backend*, seperti *PHP* dan *Node.js*

b. Kekurangan *HTML*

1. Paling sering digunakan untuk halaman web statis. Untuk fitur dinamis, bisa menggunakan *JavaScript* atau Bahasa *backend*, seperti *PHP*.
2. *HTML* tidak memungkinkan user untuk menjalankan logic. Alhasil, semua halaman web harus dibuat terpisah meskipun menggunakan elemen yang sama, seperti *header* dan *footer*.

3. Fitur-fitur baru tidak bisa digunakan secara tepat di sebagian *browser*.
4. Terkadang perilaku *browser* susah untuk diprediksi (misalnya *browser* lama tidak selalu bisa *render tag* yang lebih baru)

## 2. *Cascading Style Sheets (CSS)*

*Cascading Style Sheets (CSS)* adalah bahasa pemrograman desain yang berguna untuk menyederhanakan proses pembuatan *Website*. *CSS* merupakan Bahasa pemrograman yang dipakai untuk mendesain halaman depan atau tampilan *Website (front end)*. *CSS* menangani tampilan dan ‘rasa’ dari halaman web.

Ada banyak hal yang dapat dilakukan menggunakan *CSS* dibandingkan dengan Bahasa pemrograman inti seperti *HTML* dan *PHP*. Ketika menggunakan *CSS*, dapat mengatur warna *teks*, jenis *font*, baris antar paragraf, ukuran kolom, dan jenis *background* yang dipakai.

Tidak hanya itu *CSS* juga bisa untuk mendesain *layout*, variasi tampilan di berbagai perangkat yang berbeda, dan berbagai efek yang dipakai di dalam *Website*.

Ada beberapa keuntungan yang bisa didapatkan Ketika menggunakan *CSS*, seperti:

- a. Mempercepat proses desain

Jika sesuai dengan pengertian *CSS*, tidak perlu menyalin satu per satu ke setiap *file* halaman apabila menggunakan desain yang sama di beberapa halaman *HTML*. Cukup menegetikkan satu kali fungsi *CSS* kemudian menggunakannya di berbagai halaman *HTML*. Fungsi *CSS* yang dibuat dalam satu *file* dapat dipanggil ke berbagai halaman web tanpa harus menyalin baris kode fungsi berkali-kali.

- b. Halaman lebih cepat dimuat

Apabila menggunakan *CSS*, tidak perlu menuliskan atribut *tag HTML* di setiap *file*. Hanya cukup menulis satu aturan *CSS* dan menerapkannya di berbagai *file* yang membutuhkannya hanya



dengan memanggilnya. Jadi satu *file* hanya mengandung sedikit baris kode yang dimuat. Sedikit kode inilah yang akan membuat proses download menjadi lebih cepat.

c. Proses pemeliharaan mudah

CSS memudahkan untuk mengubah tampilan di berbagai halaman. Hanya dengan mengubah fungsi *style* di *file CSS* maka seluruh tampilan yang menggunakan fungsi tersebut akan berubah secara otomatis.

d. *Style* lebih beragam dibandingkan *HTML*

CSS mempunyai atribut lebih beragam dibandingkan dengan *HTML*.

e. Kompatibel dengan berbagai macam perangkat

CSS memungkinkan konten dapat dioptimasi di lebih dari satu perangkat. Misal Ketika memproses sebuah dokumen. Jika menggunakan CSS, bisa menyesuaikan tampilan dokumen di perangkat versi lama sekaligus di versi yang baru.

f. CSS menjadi standar pengembangan *Website*

Hampir seluruh *Website* yang ada di internet menggunakan CSS di dalamnya. Selain tampilannya yang lebih menarik, kebanyakan *browser* populer saat ini juga mendukung CSS.

### 3. *Hypertext Pre-Processor (PHP)*

*Hypertext Pre-Processor (PHP)* adalah sebuah bahasa pemrograman *server-side scripting* yang bersifat *open source*. Bahasa pemrograman ini banyak digunakan untuk pengembangan *Website*.

Sebagai sebuah *scripting language*, *PHP* menjalankan instruksi pemrograman saat proses runtime. Hasil dari instruksi tentu akan berbeda tergantung data yang diproses. Inilah yang membuat *PHP* sering digunakan untuk membangun *Website* yang dinamis seperti toko *online*.

*PHP* merupakan bahasa pemrograman *server-side*, maka *script* dari *PHP* nantinya akan diproses di *server*. Jenis *server* yang sering digunakan bersama dengan *PHP* antara lain *Apache*, *Nginx*, dan *LiteSpeed*.

Selain itu, *PHP* juga merupakan bahasa pemrograman yang bersifat *open source*. Pengguna bebas memodifikasi dan mengembangkan sesuai dengan kebutuhan mereka.

#### 4. *JavaScript*

*JavaScript* adalah bahasa pemrograman tingkat tinggi dan dinamis. *JavaScript* populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Google Chrome, Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode *JavaScript* dapat disisipkan dalam halaman web menggunakan tag `<script>`. *JavaScript* merupakan salah satu teknologi inti *World Wide Web* selain *HTML* dan *CSS*. *JavaScript* membantu membuat halaman web interaktif dan merupakan bagian aplikasi web yang esensial.

Awalnya hanya diimplementasi sebagai *client-side* dalam penjelajah web, kini engine *JavaScript* disisipkan ke dalam perangkat lunak lain seperti dalam *server-side* dalam *server* web dan basis data, dalam program *non web* seperti perangkat lunak pengolah kata dan pembaca *PDF*, dan sebagai *runtime environment* yang memungkinkan penggunaan *JavaScript* untuk membuat aplikasi *desktop* maupun *mobile*.

Sama seperti hal teknis lainnya dalam dunia web, *JavaScript* juga punya kelebihan dan kekurangannya:

##### a. Kelebihan *JavaScript*

1. Tidak membutuhkan compiler karena web *browser* mampu menginterpretasikannya dengan *HTML*
2. Lebih mudah dipelajari jika dibandingkan dengan bahasa pemrograman lainnya
3. *Error* atau kesalahan lebih mudah dicari dan ditangani
4. Dapat dialih tugaskan ke elemen halaman web atau even tertentu, misalnya klik atau *mouseover*
5. *JavaScript* dapat digunakan di berbagai *browser*, *platform*, dan lain-lain

6. Dapat menggunakan *JavaScript* untuk memvalidasi input dan mengurangi keinginan untuk mengecek data secara manual
7. Dengan *JavaScript*, *Website* menjadi lebih interaktif dan juga mampu menarik perhatian lebih banyak pengunjung
8. Lebih cepat dan ringan jika dibandingkan dengan bahasa pemrograman lainnya

b. Kekurangan *JavaScript*

1. Berisiko tinggi terhadap eksploitasi
2. Dapat dimanfaatkan untuk mengaktifkan kode berbahaya di komputer pengguna
3. Tidak selalu didukung oleh berbagai *browser* dan perangkat
4. *JavaScript code snippet* agak banyak
5. Dapat di-*render* secara berbeda pada masing-masing perangkat yang malah mengarah ke inkonsistensi

## 5. *Framework*

*Framework* adalah kerangka kerja yang digunakan untuk mengembangkan aplikasi berbasis *desktop* atau aplikasi berbasis *Website*. *Framework* sengaja diciptakan untuk membantu *developer* mengembangkan aplikasi lebih cepat serta tersusun dan terstruktur.

Dengan menggunakan *Framework* akan lebih mudah untuk membuat aplikasi, karena hanya perlu menyusun komponen-komponen pemrograman yang sudah jadi. Sehingga *developer* dan *programmer* tidak perlu melakukan koding program yang diulang-ulang.

*Framework* memiliki fungsi utama untuk mempermudah para *developer* mengembangkan *software* dan aplikasi. Karena *Framework* memiliki program yang tersusun dengan struktur *MVC (Model View Controller)*. Selain itu, ada tiga fungsi *Framework* yang dapat diketahui dibawah ini:

a. Program menjadi lebih terstruktur dan tersusun

Saat *developer* mengerjakan pengembangan *software* atau aplikasi yang besar, maka program yang akan ditulis di dalamnya menjadi semakin banyak. Terkadang, semakin banyaknya program yang ditulis akan menjadikan proses *debugging* semakin lambat. Selain itu, saat mencari kode program yang *error* akan semakin sulit karena program yang tidak terstruktur.

Oleh karena itu, dengan adanya *Framework* dapat menjadikan program menjadi lebih terstruktur sehingga dapat dengan mudah menemukan kode yang perlu diperbaiki. Bahkan beberapa jenis *Framework* ada yang menerapkan konsep *MVC (Model View Controller)* yang akan memudahkan *developer* untuk memisahkan antara logika dan *View*.

b. Praktis Untuk *Developer*

Saat mengembangkan *software* atau aplikasi, akan lebih praktis ketika menggunakan *Framework*. Alasannya karena *Framework* sudah menyediakan kode berupa *function* dan *class* jadi proses pembuatan *software* atau aplikasi akan menjadi lebih cepat. Cukup memanggil *function* atau *class* tersebut ke dalam kode program.

c. Memiliki keamanan yang lebih unggul

Selama bertahun-tahun, keamanan untuk *software* dan aplikasi telah menjadi perhatian utama bagi *developer*. Beberapa dari *developer* telah menggunakan segala cara untuk mengamankan *software* dan aplikasi yang telah mereka buat.

Setelah *Framework* hadir, *developer* boleh sedikit tenang soal keamanan program yang telah dibuat. Karena setiap celah keamanan dari *Framework* sudah diidentifikasi oleh 10-100+ *developer* ahli. *Framework* juga terus diperbarui versinya untuk menawarkan fitur baru dan menangani *bug* yang meminimalisir celah keamanan *Framework*.

*Framework* memiliki tiga jenis *platform*, yaitu *Ajax Framework*, *web application Framework* dan *application Framework*. Dari ketiga *platform* tersebut memiliki perbedaan penggunaannya. Berikut adalah penjelasan singkat dari ketiga jenis *Framework*.

**a. *Ajax Framework***

*Ajax Framework* adalah singkatan dari Asinkron *JavaScript* dan *XML*. *Ajax* adalah teknik baru untuk membuat aplikasi web yang lebih baik, lebih cepat, dan lebih interaktif dengan bantuan *XML*, *HTML*, *CSS*, dan *Java Script*.

*Ajax* merupakan salah satu *Framework* yang sangat disenangi oleh para *developer* karena dengan menggunakan *Ajax* dapat:

1. *Update* halaman web tanpa memuat ulang halaman tersebut
2. *Request* data dari *server*, setelah halaman dimuat
3. *Receive* data dari *server*, setelah halaman dimuat
4. Kirim data ke *server*, menggunakan *backend* halaman

**b. *Web Application Framework***

*Web Application Framework* adalah *software* yang dirancang untuk mendukung pengembangan web *application*, *web services*, *web resources* dan *web API*. Dengan adanya *web application Framework* mengembangkan *Website* jadi lebih cepat dan lebih mudah.

**c. *Application Framework***

*Application Framework* adalah *library* yang terdiri dari *software* yang digunakan oleh *developer* untuk mengembangkan struktur aplikasi. *Application Framework* berfungsi sebagai alat untuk membuat *template* dan struktur aplikasi menggunakan cara yang lebih efisien dan lebih cepat.

## 6. *jQuery*

*jQuery* adalah *library JavaScript* yang cukup andal, ringkas, dan mempunyai fitur yang cukup lengkap. *Library* ini membuat pemrosesan di *HTML* seperti perubahan dan manipulasi dokumen, *event handling*, animasi, dan *Ajax* dapat menjadi lebih sederhana. Hal ini didukung dengan *API* yang mudah digunakan dan dapat bekerja di berbagai macam *browser*.

Menggunakan kombinasi *versatility* (keserbagunaan) dan *extensibility* (bisa dikembangkan), *jQuery* sudah mengubah cara ribuan bahkan jutaan *developer* menggunakan bahasa pemrograman *JavaScript*. Ini membuktikan bahwa *jQuery* merupakan salah satu *library* yang cukup populer di *JavaScript*.

Fungsi dari *jQuery* ini sendiri adalah sebagai *library JavaScript* yang akan membantu mengatur interaksi antara *JavaScript* dengan *HTML* yang berjalan di sisi klien.

*jQuery* mempunyai beberapa fitur unggulan, berikut beberapa fitur unggulan *jQuery* antara lain:

1. Manipulasi *HTML/DOM*
2. Manipulasi *CSS*
3. *HTML Event Methods*
4. Efek dan Animasi
5. *AJAX*
6. *CSS3 Compliant*
7. *Cross-Browser*

Sama seperti hal teknis lainnya dalam dunia web, *jQuery* juga punya kelebihan dan kekurangannya:

- a. Kelebihan *jQuery*
  1. Dapat berinteraksi baik dengan berbagai macam tipe bahasa pemrograman lain
  2. Mendukung berbagai macam *plugin*
  3. Membuat sebuah animasi dasar dengan sangat mudah

b. Kekurangan *jQuery*

1. Perangkat lunak *open source* terkadang mempunyai beberapa masalah
2. Saat ini sudah banyak versi *jQuery* yang tersedia, ini berarti beberapa versi sudah tidak kompatibel dengan daripada yang lainnya
3. *jQuery* menyediakan fitur *lightweight interface* yang mungkin saja akan menimbulkan sebuah permasalahan di kemudian hari

## 7. *Bootstrap*

*Bootstrap* adalah *Framework open-source* khusus *front end* yang awalnya dibuat oleh Mark Otto dan Jacob Thornton untuk mempermudah dan mempercepat pengembangan web di *front end*. *Bootstrap* memiliki semua jenis *HTML* dan *template* desain berbasis *CSS* untuk berbagai fungsi dan komponen, seperti navigasi, sistem *grid*, *carousel* gambar, dan tombol (*button*).

*Framework* ini memang menghemat waktu *developer* karena tidak perlu mengelola *template* berkali-kali. Namun, fungsi utama dari *Bootstrap* adalah untuk membuat situs yang responsif. Interface *Website* akan bekerja secara optimal di semua ukuran layar baik di layar smartphone maupun layar komputer/laptop.

*Developer* tak perlu lagi mendesain situs khusus untuk perangkat tertentu. Trafik dan jangkauan audiens versi *desktop* tidak akan hilang dan tetap diarahkan ke *Website* versi *mobile*.

Karena berisikan sekumpulan sintaks yang menjalankan fungsi spesifik, *Bootstrap* memiliki berbagai tipe *file* di dalamnya. Berikut tiga *file* utama yang mengelola interface user dan fungsionalitas *Website*:

a. *Bootstrap.css*

*Bootstrap.css* adalah *Framework* yang mengatur dan mengelola *layout Website*. *HTML* mengatur konten dan struktur halaman web, sedangkan *CSS* berurusan dengan *layout* situs. Oleh karena itu,

kedua struktur ini harus bekerja bersama-sama untuk melakukan tindakan tertentu.

Dengan CSS, bisa menghadirkan tampilan yang seragam di banyak halaman *Website*. Semua ini berkat keberadaan fungsinya. Anda pun tak perlu lagi menghabiskan waktu berjam-jam lamanya hanya untuk mengubah, misalnya, lebar desain.

Jika menggunakan CSS, yang perlu dilakukan adalah merujuk atau mengarahkan halaman web ke *file* CSS. Nantinya semua perubahan dapat dilakukan di satu *file* saja.

Fungsi CSS tidak terbatas hanya di gaya teks saja, te tapi juga dapat digunakan untuk membuat aspek lainnya di halaman web, seperti tampilan tabel dan gambar. Sayangnya, CSS punya banyak *declaration* dan *selector*. Jadi, butuh waktu untuk mengingat keduanya.

b. *Bootstrap.js*

*File* ini merupakan bagian inti dari *Bootstrap*. Di dalamnya terdapat *file JavaScript* yang bertanggung jawab atas interaktivitas *Website*.

*Developer* menggunakan *jQuery* untuk menghemat waktu karena tidak perlu menulis sintaks *JavaScript* berulang kali. *jQuery* sendiri adalah *open source* yang populer dan juga *library JavaScript* lintas *platform* yang memperbolehkan user-nya untuk menambah berbagai fungsionalitas ke *Website*. Berikut contoh fungsi dan kegunaan *jQuery*:

1. Menjalankan permintaan *Ajax*, seperti mengurangi (*subtract*) data dari lokasi lain secara dinamis
2. Membuat *widget* menggunakan koleksi *plugin JavaScript*
3. Membuat animasi menggunakan *CSS property*
4. Membuat konten *Website* terlihat lebih dinamis
5. Walaupun *Bootstrap* dengan *CSS property* dan elemen *HTML*-nya bisa berfungsi normal, tetap saja *Framework* ini



memerlukan *jQuery* untuk membuat desain yang responsif. Jika tidak, hanya bisa menikmati *CSS* yang statis dan datar

### c. *Glyphicons*

Ikon berperan penting pada *front-end Website*. Bahkan terkadang ikon dikaitkan dengan tindakan tertentu serta data yang ada di dalam *interface user*. Untuk *task* ini, *Bootstrap* menggunakan *Glyphicons*.

*Bootstrap* menawarkan *Glyphicons Halflings* yang bisa dipakai secara gratis. Versi gratisnya punya tampilan yang standard tapi sudah lebih dari cukup dengan adanya fungsi-fungsi yang penting.

Jika menginginkan ikon yang lebih *stylish*, *Glyphicons* menyediakan berbagai *set premium* yang akan membuat *Website* terlihat lebih menarik dan juga sesuai dengan *niche*. Masing-masing ikon serta ikon yang dikhususkan untuk suatu tema dapat diunduh secara gratis di sejumlah *Website*, seperti *FlatIcon*, *GlyphSearch*, dan *Icons8*.

Tampilan sebagian ikon dapat dimodifikasi dengan *CSS*, sedang sebagian lainnya sudah *default*. Pilih dan gunakan ikon yang sesuai dengan konsep dan tema *Website*.

## 8. *Laravel*

*Laravel* adalah satu-satunya *Framework* yang membantu untuk memaksimalkan penggunaan *PHP* di dalam proses pengembangan *Website*.

*PHP* menjadi bahasa pemrograman yang sangat dinamis, tapi semenjak adanya *Laravel*, dia menjadi lebih *powerful*, cepat, aman, dan simpel. Setiap rilis versi terbaru, *Laravel* selalu memunculkan teknologi baru di antara *Framework PHP* lainnya.

*Laravel* fokus di bagian *end-user*, yang berarti fokus pada kejelasan dan kesederhanaan, baik penulisan maupun tampilan, serta menghasilkan fungsionalitas aplikasi web yang bekerja sebagaimana mestinya. Hal ini membuat *developer* maupun perusahaan menggunakan *Framework* ini

untuk membangun apa pun, mulai dari proyek kecil hingga skala perusahaan kelas atas.

*Laravel* mengubah pengembangan *Website* menjadi lebih elegan, ekspresif, dan menyenangkan. Selain itu, *Laravel* juga mempermudah proses pengembangan *Website* dengan bantuan beberapa fitur unggulan, seperti *Template Engine*, *Routing*, dan *Modularity*.

Manfaat *Laravel* dalam Proses Pengembangan *Website* antara lain sebagai berikut:

- a. Pertama, *Website* menjadi lebih scalable (mudah dikembangkan)
- b. Kedua, terdapat namespace dan tampilan yang membantu untuk mengorganisir dan mengatur sumber daya *Website*
- c. Ketiga, proses pengembangan menjadi lebih cepat sehingga menghemat waktu karena *Laravel* dapat dikombinasikan dengan beberapa komponen dari *Framework* lain untuk mengembangkan *Website*

#### **2.4.2 Web Editor**

*Web Editor* merupakan program aplikasi yang berfungsi untuk mengetikkan perintah-perintah dokumen web baik *client side scripting* maupun *server-side scripting*. Saat ini banyak tersedia web Editor mulai dari yang paling sederhana hingga yang lebih smart. Mulai dari web editor yang berbayar hingga yang gratis. Dalam pembuatan sistem ini kami menggunakan web editor *Sublime Text* dan *Visual Studio*.

#### **2.4.3 Database Server**

*Database Server* ialah sebuah program komputer yang menyediakan layanan pengelolaan basis data dan melayani komputer atau program aplikasi basis data yang menggunakan model klien/*server*. Di sini kami menggunakan *XAMPP*, yang di dalamnya terdapat *Apache* dan *MySQL* untuk dikoneksikan ke *PHP MyAdmin* agar *Database* bisa terhubung.

#### 2.4.4 *Web Hosting*

*Hosting* berasal dari kata *host*. Komputer yang terhubung dalam jaringan. Memanfaatkan fasilitas yang tersedia dalam suatu computer yang terhubung dengan jaringan. *Hosting* menyediakan sumber daya *server-server* untuk disewakan sehingga memungkinkan organisasi atau individu menmpatkan informasi di internet, *server Hosting* terdiri dari gabungan *serverserver* atau sebuah *server* yang terhubung dengan jaringan internet berkecepatan tinggi. (Bonafit Nugroho : 2008.)

#### 2.4.5 Perancangan Sistem


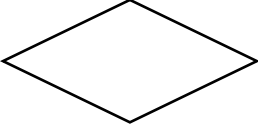
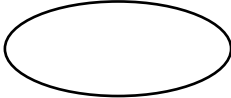

Sebelum pembuatan sistem, kami merancang model sistem ini menggunakan beberapa konsep yaitu :

a. Entity Relationship Diagram (ERD)

Dalam melakukan analisis data, *ERD* dapat digunakan untuk menggambarkan masing-masing entitas dan relasi antar entitas dari bentuk notasi grafik menjadi sebuah diagram data sehingga segala pemrosesan data secara transactional dapat tergambar dengan jelas.

*Entity Relationship Diagram* merupakan model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *ERD* merupakan metode yang sering dipakai untuk membantu menganalisa pembuatan basis data suatu sistem, sehingga dengan adanya *Entity Relationship Diagram* hubungan antar table dapat diidentifikasi dengan jelas.

Dengan *ERD* pembuatan perencanaan *Website* dapat dideskripsikan. Selain itu juga membantu mempermudah penulisan dalam membuat *Website*.

Simbol	Keterangan
	Entitas adalah kumpulan dari objek yang dapat diidentifikasi secara unik.
	Relasi adalah hubungan yang terjadi antara salah satu lebih entitas.
	Atribut adalah karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas.
	Hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya.

**Tabel 2.1 Komponen ERD**

*b. Conceptual Data Model (CDM)*

*Conceptual Data Model (CDM)* adalah model yang digunakan untuk memodelkan struktur dari keseluruhan aplikasi basis data secara detail dan logis. *Conceptual Data Model* dalam penerapannya dapat disamakan dengan *Entity Relationship Diagram* yang fungsinya memang sama yaitu memodelkan struktur dari basis data. Pada *Conceptual Data Model* belum terdapat *foreign key* yang menghubungkan antara relasi satu dengan lainnya. *Foreign key* tersebut nantinya akan muncul pada saat *Conceptual Data Model* digeneralisasikan menjadi *Physical Data Model*.

*Conceptual Data Model (CDM)* merupakan model yang merepresentasikan table yang merupakan entitas yang berisi atribut. Setiap entitas memiliki satu *primary key* yang bersifat unik (nilainya tidak sama dengan nilai lainnya) dan setiap entitas berhubungan dengan entitas lain yang disebut *relationship*.

Pada *relationship* terdapat beberapa jenis yaitu: *one to one* (hubungan satu tabel yang memiliki nilai satu dengan tabel lain), *one to many* (hubungan satu tabel yang memiliki nilai yang banyak dengan tabel lain), dan *many to many* (hubungan satu tabel yang memiliki nilai yang banyak dengan tabel lain). Jenis-jenis objek dalam *Conceptual Data Model* yaitu:

1. *Entity*

Untuk membuat entitas, klik item pada palette dengan label *entity*, kemudian klik space putih di sebelah kanan. Untuk membuat beberapa entitas, klik terus sebanyak jumlah entitas yang dibutuhkan.

2. *Relationship*

Untuk membuat relasi antara 2 buah entitas, klik *item* pada palette dengan label *relationship*, kemudian hubungkan kedua entitas. Beri nama pada relasi yang sudah dibuat dengan melakukan double klik pada relasi dengan nama pada *field name* pada *tab general*.

3. *Inheritance*

Untuk membuat inheritance dari sebuah entitas, terlebih dahulu buat entitas-entitas lain yang merupakan *child* dari entitas *parent*. Lalu klik *item* pada *palette* dengan label *inheritance*, hubungkan entitas *parent* dengan salah satu entitas *child*. Untuk menghubungkan entitas *child* yang lain, Tarik garis antara lambang *inheritance* (bentuk setengah lingkaran) dengan entitas *child*. Kemudian beri nama pada *inheritance* anda.

- c. *Physical Data Model (PDM)*

*Physical Data Model* merupakan representasi fisik dari *Database* yang akan dibuat dengan mempertimbangkan *DBMS* yang akan digunakan. *Physical Data Model* dapat dihasilkan (di generate) dari *CDM* yang valid. Jenis-jenis objek dalam *PDM* diantaranya sebagai berikut yaitu tabel, *View*, *reference*.

*Physical Data Model (PDM)* merupakan model yang merepresentasikan tabel yang terstruktur, termasuk nama kolom, tipe data kolom, *primary key*, *foreign key* dan *relationship* yang menghubungkan satu tabel dengan tabel lainnya. *PDM* dan *CDM* memiliki kemiripan dari segi entitas, atribut, dan *relationship*. Perbedaannya yaitu pada *CDM* tidak terdapat *foreign key*, sedangkan pada *PDM* terdapat *foreign key* dimana satu tabel bergantung pada tabel lainnya. *Foreign key* yaitu *primary key* yang berada pada tabel yang memiliki hubungan dengan tabel tersebut.

## **2.5 Ruang Lingkup Proses Bisnis**

Pada awal Tahun 2020, dunia sedang waspada dengan adanya penyakit *Covid-19* yang disebabkan oleh virus coronan. Sejak adanya pandemic *Covid-19* ini semua kegiatan sehari-hari di segala bidang terganggu. Contohnya di dunia pendidikan, karena dampak dari virus ini semua kegiatan belajar mengajar dilakukan secara daring (dalam jaringan). Oleh karena itu akses teknologi saat ini sangat dibutuhkan dalam era pandemic seperti ini. Untuk mempermudah pembelajaran secara daring, kami berupaya untuk membuat sebuah media pembelajaran sebagai tempat untuk mengakses segala sesuatu yang berhubungan dengan belajar mengajar di SD tersebut. Media pembelajaran tersebut yakni *E-Learning* yang dapat diakses oleh para guru maupun siswanya.

Penggunaan *E-Learning* dalam proses belajar mengajar merupakan salah satu upaya untuk meningkatkan efektivitas serta kualitas proses pembelajaran yang pada nantinya juga dapat meningkatkan kualitas hasil belajar siswa.

Sistem ini bisa dipakai oleh semua instansi pendidikan pada tingkat Sekolah Dasar. Dalam pembuatan program ini, kami mengambil studi kasus tidak riil, sehingga program ini bisa dikatakan bersifat umum atau tidak terikat pada suatu instansi tertentu.