

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Penelitian Terdahulu**

Penelitian terdahulu digunakan sebagai dasar dan acuan dalam mencari gambaran dan kerangka berfikir sekaligus sebagai salah satu bahan kajian dalam penelitian ini. Penelitian terdahulu yang menjadi acuan penelitian ini antara lain:

1. Tesis dengan judul “*Automatic Generation of Web CRUD Applications*” oleh Marco Livraghi [1]. Penelitian ini membahas tentang bagaimana membuat generator CRUD untuk aplikasi berbasis web secara otomatis. Penulis menggunakan pendekatan teknik dan ide yang hampir sama dalam penelitian penulis yaitu menggunakan *Dynamic Tablebased CRUD Form Generating*. Perbedaannya dengan penelitian penulis adalah pada desain, implementasi dan kerangka kerjanya di mana penulis menggunakan framework CodeIgniter dan Bootstrap, sedangkan penelitian Marco Livraghi menggunakan framework Spring dan AngularJS. Kesimpulan dari penelitian Marco Livraghi adalah teknik automatic generation pada aplikasi CRUD adalah solusi untuk memangkas waktu dan biaya pengembangan dan perancangan aplikasi.
  
2. Jurnal dengan judul “*Analysis of System Requirements of Go-Edu Indonesia Application as a Media to Order Teaching Services and Education in Indonesia*” oleh Marlina B. Winanti *et al* [2]. Penelitian ini membahas mengenai kebutuhan fungsional dan non-fungsional pada aplikasi Go-Edu

Indonesia. Pada penelitian ini, disimpulkan bahwa aplikasi Go-Edu Indonesia dibagi menjadi 2 versi aplikasi yakni versi untuk pengajar dan versi untuk pelajar/siswa. Pada penelitian tersebut, dijelaskan bahwa sistem harus dapat menyimpan/memasukan data yang dibutuhkan seperti: **Nama, Tempat dan Tanggal Lahir, Gender, Agama, Alamat, Nomor Telepon, Pendidikan, No Identitas, Kartu Identitas, dan Photo**. Sistem Go-Edu Indonesia juga harus dapat mengelola profil dan mengunggah bukti kompetensi seperti sertifikat kompetensi atau sertifikat pelatihan. Selain itu, sistem Go-Edu Indonesia dapat mengelola *order* dan pembayaran melalui transfer, tunai maupun *eMoney*, melihat *history* atau riwayat transaksi, jadwal dan lokasi pengajar. Penulis menjadikan penelitian ini sebagai salah satu bahan acuan kebutuhan sistem dasar dan akan dikembangkan oleh Penulis dalam penelitian Penulis. Perbedaan dengan penelitian Penulis adalah pada ranah model/proses bisnis dan produk akhirnya yaitu produk hasil penelitian Penulis adalah produk platform yang mengusung penyediaan fasilitas pembelajaran melalui media elektronik dan fasilitas pembelajaran jarak jauh (*Distance Learning*) alih-alih mempertemukan pengajar dan pelajar.

3. Jurnal dengan judul "*Classroom Booking Information System Integrated with Course Scheduling Information System*" oleh Syahrul Mauluddin *et al* [3]. Penelitian ini bertujuan untuk membuat sistem informasi peminjaman kelas yang terintegrasi dengan sistem informasi penjadwalan dosen. Penelitian ini menggunakan pendekatan berorientasi objek yang sama dengan penelitian Penulis. Meskipun dalam penelitian tersebut tidak dijelaskan bagaimana

integrasi dengan sistem penjadwalan dosennya, penelitian ini memberikan beberapa wawasan tentang permasalahan yang ada di sekretariat jurusan yang bermanfaat sebagai salah satu acuan Penulis dalam penelitian Penulis.

4. Jurnal dengan judul “Pembangunan Sistem Aplikasi Kuliah *Online* Sebagai Sarana Penunjang Kegiatan Perkuliahan Di Universitas Pendidikan Indonesia Kampus Tasikmalaya” oleh Nizar Rabbi Radliya [4]. Penelitian ini menghasilkan sistem kuliah online dan membahas mengenai *E-Learning* yang sama dengan penelitian Penulis. Penelitian ini menggunakan pendekatan berorientasi objek dan menggunakan metode pengembangan RUP (*Rational Unified Proses*). Penelitian ini digunakan oleh Penulis sebagai acuan yang memberikan gambaran mengenai *E-Learning* dan kuliah *online* diluar objek penelitian Penulis yaitu kuliah online UNIKOM. Diharapkan penelitian ini akan memberi wawasan kepada penulis dalam membangun platform yang secara umum dapat digunakan di institusi pendidikan yang beragam.
5. Jurnal dengan judul “Aplikasi *E-Learning* di SMP Negeri 46 Bandung” oleh Julian Chandra Wibawa *et al* [5]. Pada penelitian ini dibahas mengenai permasalahan kegiatan pembelajaran di SMP Negeri 46 Bandung yang belum optimal dan bagaimana membuat aplikasi *E-Learning* agar permasalahan tersebut dapat teratasi dengan memanfaatkan internet untuk menyediakan pembelajaran jarak jauh (*Distance Learning*). Dalam bagian kesimpulan dan saran pada jurnal ini diungkapkan agar mengembangkan sistem *E-Learning* dengan fasilitas *Chating* dan *Role Base Security* yaitu pengamanan

berdasarkan user dan password, yang hal tersebut akan dibangun oleh Penulis dalam penelitian Penulis saat ini.

6. Jurnal dengan judul "*E-Teaching/Epedagody Threats & Opportunities for Teacher in Heis*" oleh Allah Nawaz *et al* [6]. Penelitian ini membahas tentang peluang dan ancaman terhadap pengajar saat ini yang dituntut untuk dapat memiliki keterampilan teknis dalam bidang komunikasi dan informasi dalam rangka menyediakan materi pembelajaran secara online. Penelitian ini dilakukan di pakistan yang notabene merupakan negara berkembang yang sama halnya dengan Indonesia, sehingga penelitian ini sangat memadai dan relevan jika dijadikan acuan dan memandang permasalahan yang timbul di sisi pengajar Pakistan akan sama halnya dengan pengajar di Indonesia dalam menghadapi tantangan baru yaitu menyediakan pembelajaran melalui media elektronik dan internet untuk pembelajaran jarak jauh (daring). Penulis menjadikan penelitian ini sebagai salah satu acuan dalam merancang platform yang memudahkan pengajar dalam menyediakan materi pembelajaran dan melaksanakan pembelajaran jarak jauh.
7. Jurnal dengan judul "*Effective E-Learning and E-Teaching — A Theoretical Model*" oleh Maureen Snow Andrade [7]. Penelitian ini membahas model teoritis *E-Learning* dan *E-Teaching* sebagai komponen-komponen yang membangun dan meningkatkan efektivitas pembelajaran jarak jauh atau daring. Penelitian ini memberikan gambaran umum kepada Penulis untuk merancang kebutuhan dasar yang harus dibangun dalam menyediakan fasilitas pembelajaran jarak jauh secara efektif.

8. Artikel ilmiah dengan judul “*E-Teaching*” oleh Frank Thissen (*Deutsche Bibel Gesellschaft Academic*) [8]. Penulis artikel ini menyusun gagasan tentang apa dan bagaimana *E-Teaching* secara konseptual dan struktural. Di bahasan ini juga dikemukakan beberapa alat-alat (*tools*) yang harus disediakan untuk mendukung proses *E-Teaching* salah satunya yang terpenting adalah dengan menyediakan lingkungan pembelajaran *virtual* (kelas *virtual*).
  
9. Jurnal dengan judul “*E-teaching as companion to e-learning; supporting digital pedagogies and practice in higher education*” oleh Sue Watling [9]. Penelitian ini membahas tentang teknologi sebagai penguat pembelajaran melalui *Virtual Learning Environment* (Lingkungan Pembelajaran *Virtual*), namun lingkungan pembelajaran *virtual* yang dimanfaatkan dewasa ini lebih ke sekedar untuk menyimpan dokumen-dokumen *digital* (materi pembelajaran *digital*) bukan sebagai pendukung pembelajaran yang kolaboratif dan interaktif. Penelitian ini menjadi salah satu hal yang mendasari Penulis dalam mengusung penelitian dan rancang-bangun platform *E-Teaching* agar lingkungan pembelajaran dapat menjadi fasilitas pembelajaran yang kolaboratif, interaktif dan dapat menjadi instrumen tambahan yang dapat diandalkan oleh pengajar dalam melaksanakan kegiatan belajar mengajar secara *online*.

## 2.2 Landasan Teori

### 2.2.1 Pengertian *E-Learning*

*E-Learning* sampai saat ini masih belum memiliki definisi yang baku, namun para ahli berpendapat dan menjelaskan bahwa *E-Learning* adalah segala bentuk pembelajaran yang memanfaatkan media elektronik. Definisi ini lebih menekankan bahwa segala bentuk penggunaan alat elektronik untuk suatu pembelajaran merupakan *E-Learning* karena sesuai dengan penggunaan “e” di depan kata “*Learning*” yang artinya adalah singkatan dari “*elektronik*” [10].

Definisi lainnya dikemukakan oleh Hartley (2001), Rosenberg (2001), dan Kamarga (2002) yang menjelaskan *E-Learning* adalah penggunaan teknologi internet dan komputer untuk membantu proses belajar manusia. Jika pada definisi pertama *E-Learning* menekankan pada apapun pembelajaran yang menggunakan media elektronika, maka definisi ini menekankan penggunaan komputer dan jaringan. Dalam pemahaman ini *E-Learning* harus memiliki fasilitas atau instrumen terhubung antara sisi pengajar dan sisi pelajar yang memungkinkan pembelajaran jarak jauh dapat dilakukan secara sinkron.

Di samping pengertian di atas, menurut Mohammad Yazdi dalam jurnalnya menjelaskan *E-Learning* adalah suatu pengajaran dan pembelajaran jarak jauh yang menggunakan media elektronik untuk menyampaikan isi pembelajaran, interaksi, atau bimbingan. *E-Learning* memiliki beberapa karakteristik yang diantaranya adalah murid dan guru dapat memanfaatkan teknologi elektronik dan dapat memaksimalkan manfaat keunggulan komputer. Selain itu, *E-Learning* dapat

membuat murid menggunakan bahan ajar yang bersifat mandiri yang dimana murid akan belajar mandiri dalam mempelajari bahan ajar yang diberikan [11].

### **2.2.2 Karakteristik *E-Learning***

Karakteristik *E-Learning* meskipun memiliki perbedaan, namun tidak terlepas dari karakteristik pembelajaran tradisional/konvensional yang dilakukan dengan sistem tatap muka yang pada dasarnya adalah proses penyampaian pesan atau materi dari pengajar ke pelajar.

Pembelajaran konvensional dengan sistem tatap muka mengedepankan interaksi antar personal atau interaksi sosial karena bertemu secara langsung dan proses belajar dapat dikatakan berjalan jika antara pengajar dan pelajar bertemu dalam waktu dan tempat yang sama, misalnya dalam ruang kelas.

Perbedaan antara pembelajaran konvensional dan *E-Learning* adalah pada *E-Learning* yang memiliki fokus untuk meningkatkan kemampuan personal dan mewujudkan kemungkinan pelajar untuk belajar secara mandiri. *E-Learning* memiliki beberapa prinsip-prinsip dalam proses pembelajarannya seperti dikemukakan oleh Littlejohn & Peger (2007) antara lain:

#### **1. Personalisasi**

Pelajar dapat melakukan penyesuaian gaya belajar sesuai dengan kemampuan dan preferensinya dan dapat menentukan cara interaksi dengan pelajar yang lainnya.

#### **2. Keamanan**

*E-Learning* menjadi fasilitas yang aman dalam artian keamanan data yang

tidak bocor ke tangan pihak yang tidak memiliki hak terhadap data tersebut dan keamanan dalam hal jika terjadi bencana maka data pada *E-Learning* masih dapat tersedia dan tidak hilang, lain halnya jika data disimpan dalam kertas, jika terjadi kebakaran maka kertas dan data yang tersimpan akan hilang.

### 3. **Belajar Mandiri**

*E-Learning* memungkinkan pelajar untuk belajar dengan gaya dan preferensinya sendiri dan dapat mengulang-ngulang materi pelajaran yang sedang dia pelajari sesuai keinginannya sampai betul-betul dapat dipahami.

### 4. **Tracking**

Pengajar dapat melakukan pemantauan aktivitas peserta didiknya dengan menggunakan *E-Learning* secara individu maupun kelompok. Informasi dari pemantauan/*tracking* bermanfaat bagi pengajar sebagai penilaian baik bagi cara pengajaran maupun bagi penilaian tingkat pemahaman peserta didiknya.

### 5. **Aplikasi Pihak Ketiga**

*E-Learning* dapat memanfaatkan aplikasi dari pihak ketiga untuk mengembangkan materi atau bahan ajar seperti contohnya menggunakan aplikasi Youtube untuk menampilkan *video* atau merujuk pada *video* lain sebagai referensi tambahan.



Tabel II.1  
Karakteristik Pembelajaran Konvensional dan E-Learning

Pembelajaran Konvensional	E-Learning
<ul style="list-style-type: none"> <li>✓ Terbatas</li> <li>✓ Realtime</li> <li>✓ Kontrol pada Pendidik</li> <li>✓ Linear</li> <li>✓ Sumber Sekunder</li> <li>✓ Statis</li> </ul>	<ul style="list-style-type: none"> <li>✓ Tidak Terbatas</li> <li>✓ Fleksibel</li> <li>✓ Kontrol pada Peserta</li> <li>✓ Multidimensional</li> <li>✓ Sumber Primer</li> <li>✓ Dinamis</li> </ul>

(Sumber: D. Wahyuningsih – *E-Learning: Teori dan Aplikasi* [10])

### 2.2.3 Penerapan *E-Learning*

*E-Learning* memiliki tiga fungsi dalam penerapannya yaitu:

1. Fungsi *Supplement* yaitu sebagai tambahan di samping pembelajaran konvensional.
2. Fungsi *Complement* yaitu sebagai pelengkap pembelajaran konvensional. Fungsi ini juga disebut sebagai *hybrid learning* dimana selain menjalankan proses pembelajaran dengan sistem tatap muka, pelajar juga diwajibkan untuk turut serta dalam pembelajaran melalui *E-Learning*.
3. Fungsi *Replacement* yaitu sebagai pengganti pembelajaran konvensional. Fungsi ini diterapkan jika pembelajaran konvensional tidak dapat dilaksanakan atau dalam keadaan darurat yang tidak memungkinkan diselenggarakannya pembelajaran konvensional dengan sistem tatap muka.

### 2.2.4 Pengertian *E-Teaching*

Sama halnya dengan definisi *E-Learning*, *E-Teaching* sampai saat ini belum memiliki definisi yang baku. Namun, *E-Teaching* dapat dipahami sebagai aktivitas mengajar atau sebagai dukungan dalam proses pembelajaran dengan

menggunakan teknologi komputer dan internet. *E-Teaching* merupakan proses pembelajaran yang hampir sama dengan *E-Learning* namun memiliki perbedaan pada fokusnya, *E-Learning* berfokus pada pelajar, sedangkan *E-Teaching* berfokus pada pengajar [8]. *E-Teaching* merupakan instrumen untuk pengajar dalam melaksanakan proses mengajar melalui media elektronik dan dapat dilakukan dari jarak jauh baik secara sinkron maupun asinkron.

Berbeda dengan *E-Learning* yang memberikan kebebasan atau kontrol kepada pelajar, *E-Teaching* sepenuhnya dimonitor dan dikendalikan oleh pengajar. Sehingga *E-Teaching* memerlukan ruang virtual dimana pengajar dan pelajar dapat bertemu secara daring. Dalam konsep *E-Teaching*, ruang kelas pada pembelajaran konvensional atau sistem tatap muka diubah menjadi kelas *virtual* atau disebut *VLE* (*Virtual Learning Environment*).

Terdapat beberapa instrumen (tools) yang menjadi syarat *E-Teaching* dapat berjalan, antara lain:

1. Lingkungan belajar *virtual*/kelas *virtual* (VLE)
2. Instrumen untuk memicu proses belajar dan memberikan inspirasi kepada pelajar
3. Instrumen untuk menampilkan struktur dan informasi
4. Instrumen untuk komunikasi
5. Instrumen untuk membuat dan penyimpanan data atau materi pembelajaran
6. Instrumen untuk refleksi pelajar (menilai pemahaman pelajar)
7. Instrumen untuk memberikan umpan balik (*feedback*)

### 2.2.5 Pengertian Platform

*Platform* adalah arsitektur perangkat keras dan atau perangkat lunak yang bertindak sebagai pondasi dimana aplikasi, proses, atau teknologi dijalankan dan dikembangkan. Dalam bidang komputasi *platform* merujuk pada perangkat keras, perangkat lunak atau sistem komputer dimana aplikasi dapat berjalan. Sebuah aplikasi juga dapat bertindak sebagai platform jika menjadi basis aplikasi lain agar dapat berjalan. Contohnya adalah peramban *web* yang dapat menerima dan menjalankan *plug-in* dari pihak ketiga, dalam hal ini aplikasi peramban *web* merupakan sebuah *platform* [12].

Dewasa ini, hampir semua perusahaan pengembang perangkat lunak kelas dunia membangun platform. Beberapa jenis platform adalah sebagai berikut:

1. **Platform Utilitas**, yaitu *platform* yang menarik penggunaannya dengan menyediakan layanan utilitas gratis, contohnya: Google Search, Bing, Kayak, SkyScanner.
2. **Platform Distribusi Konten**, yaitu *platform* yang menyediakan layanan untuk mendistribusikan konten (atau iklan), contohnya Google Ads, Propeller Ads, dan Millennial Media.
3. **Platform Pengumpulan Data**, yaitu *platform* yang menyediakan layanan yang berguna bagi penggunaannya namun mendapatkan data dari penggunaan penggunaannya sendiri, contohnya: Google Maps, Waze, OpenSignal, Salesforce, dan InsideSales.
4. **Platform Interaksi Jaringan**, yaitu *platform* yang menyediakan layanan untuk pengguna dapat terhubung dengan pengguna lain dan membentuk

jaringan (jaringan sosial), contohnya: Facebook, Telegram, Whatsapp, LinkedIn dan BitCoin.

5. **Platform Teknologi**, yaitu *platform* yang menyediakan layanan berupa teknologi, contohnya: Microsoft Azure, Google Cloud Platform dan Amazon Web Services.
6. **Platform Marketplace**, yaitu *platform* yang menyediakan ruang bertemunya penjual dan pembeli sekaligus menyelesaikan proses pemesanan, pembayaran dan pengiriman, contohnya: Amazon, eBay, Tokopedia, Shopee dan Olx.
7. **Platform Layanan berdasarkan Permintaan**, yaitu *platform* yang menyediakan layanan *end-to-end* yang akan dilayani oleh pihak diluar penyedia *platform*, contohnya: Uber, Gojek, dan Grab.
8. **Platform Komputasi**, yaitu *platform* menyediakan layanan interaksi antara pengguna *platform* dengan pengembang melalui sebuah ruang penyimpanan dan penyediaan aplikasi (Google Playstore, AppStore), contohnya: IOS dan Android.
9. **Platform Konten Massal**, yaitu *platform* yang menyediakan layanan untuk mengumpulkan dan menayangkan konten yang berasal dari penggunanya, contohnya: Youtube dan Yelp.
10. **Platform E-Learning/E-Teaching**, yaitu *platform* yang menyediakan layanan untuk menyelenggarakan pembelajaran melalui media elektronik (*E-Learning*), contohnya: Google Classroom, Moodle, dan Edmodo.

### 2.2.6 PHP

PHP adalah singkatan dari “*PHP: Hypertext Preprocessor*” merupakan bahasa pemrograman dengan kode sumber terbuka (*open source*) yang digunakan secara luas oleh pengembang di seluruh dunia untuk membangun aplikasi berbasis web dengan pemrosesan di sisi server (*server side scripting*).

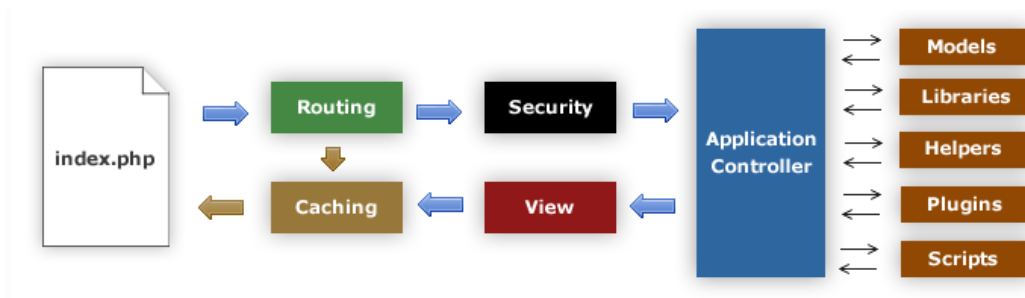
PHP dapat disisipkan secara langsung di kode HTML dengan sintax yang mirip dengan bahasa C yang mudah dipelajari. Sasaran utama PHP adalah untuk membuat halaman *web* secara dinamis dengan sangat cepat [13].

### 2.2.7 Model View Controller (MVC)

MVC adalah sebuah kaidah atau pola rancangan pengembangan perangkat lunak (*software design pattern*) yang memisahkan antara *User Interface (View)*, *Application Logic (Controller)* dan *Data (Model)* [14]. Kaidah MVC bertujuan agar pengembangan perangkat lunak dapat dilakukan dengan rapih, mudah dikelola dan mudah dikembangkan.

### 2.2.8 CodeIgniter

CodeIgniter adalah *framework* pengembangan aplikasi berbasis *web* dengan bahasa pemrograman PHP mengikuti kaidah MVC (*Model View Controller*). Tujuan utama dari framework ini adalah agar pengembang dapat membuat aplikasi berbasis web dengan cepat karena telah dilengkapi dengan serangkaian fungsi-fungsi dasar dalam bentuk *library* yang dibutuhkan pada pengembangan aplikasi berbasis *web* [15].



Gambar II.1 Diagram blok CodeIgniter

Berikut adalah penjelasan dari diagram balok CodeIgniter:

1. *Index.php* berfungsi sebagai pengontrol depan, menginisialisasi sumber daya dasar yang diperlukan untuk menjalankan CodeIgniter.
2. *Router* berfungsi untuk memeriksa permintaan HTTP untuk menentukan apa yang harus dilakukan dengan permintaan tersebut.
3. Jika ada *file cache*, maka akan dikirim langsung ke *browser*, melalui eksekusi sistem secara normal.
4. *Security* berfungsi untuk mengontrol aplikasi yang dimuat, permintaan HTTP dan data yang dikirimkan pengguna difilter untuk keamanan.
5. *Application controller* berfungsi sebagai pengendali untuk memuat model, *library*, dan sumber daya lainnya yang diperlukan untuk memproses permintaan spesifik.
6. *View* berfungsi untuk menampilkan tampilan yang telah selesai di-*render* kemudian dikirim ke browser *web* untuk dilihat. Jika *caching* diaktifkan, tampilan di-*cache* terlebih dahulu sehingga pada permintaan selanjutnya dapat dilayani.

CodeIgniter diperlihara dan dikembangkan oleh komunitas yang dikelola oleh Institut Teknologi British Columbia (BCIT) di Kanada dan sejak versi 3.0, CodeIgniter menerapkan lisensi MIT sehingga dapat digunakan secara bebas baik untuk keperluan komersil maupun non-komersil [16].

### 2.2.9 HTML

HTML adalah singkatan dari *Hypertext Markup Language* merupakan komponen pembentuk halaman *web* paling dasar. HTML menentukan struktur dan konten *web*. Teknologi pembentuk *web* di samping HTML ialah CSS (*Cascading Style Sheet*), dan *JavaScript*. CSS digunakan untuk menentukan tampilan laman *web* dan *Javascript* digunakan untuk menentukan fungsionalitas atau perilaku (*behavior*) laman *web* [17].

Kata “*Hypertext*” merujuk pada hubungan atau *link* yang jika diklik memindahkan laman ke laman yang lainnya, baik dalam satu situs web maupun ke situs *web* lainnya.

HTML menggunakan “*markup*” untuk menandai teks, gambar dan konten lainnya untuk ditampilkan di laman *web*. Setiap elemen pada HTML ditandai dengan *tag* yang terdiri dari nama elemen yang dibuka dengan “<” dan “>”, contohnya: “<section>”, “<p>”, “<div>”.

### 2.2.10 CSS

CSS (*Cascading Syle Sheet*) adalah bahasa *sylesheet* untuk mendeskripsikan presentasi/tampilan sebuah dokumen yang ditulis dengan HTML atau XML. CSS mendeskripsikan bagaimana elemen HTML ditampilkan di layar,

di kertas jika dicetak, dibacakan (menggunakan text-to-speech), atau di media lainnya [18].

### **2.2.11 Javascript**

Javascript merupakan bahasa *scripting* yang sangat populer untuk membangun laman *web* yang interaktif. Laman web yang menggunakan Javascript dapat merespon *event-event* dari pengguna dan dapat melakukan proses atau eksekusi program di sisi client (*client side*) berbeda dengan PHP yang memproses program di sisi server (*server side*) [19].

### **2.2.12 JQuery**

JQuery merupakan sebuah Javascript Library yang populer yang banyak digunakan untuk membantu pengembangan aplikasi berbasis *web* menjadi sangat cepat. JQuery memudahkan pemilihan elemen pada HTML dan melakukan tindakan pada elemen HTML yang dipilih, misalnya untuk memanipulasi elemen, menangani event, memulai animasi pada elemen, dan berkomunikasi (melakukan pertukaran data) dengan *server* menggunakan Ajax [20].

### **2.2.13 JSON**

JSON adalah kependekan dari *Javascript Object Notation* [21] merupakan notasi objek Javascript dalam bentuk teks yang digunakan untuk menyimpan atau mengirim data yang kompatibel dengan *object* di Javascript. Objek, *array* dan tipe data lainnya di Javascript dapat dengan mudah diubah menjadi JSON dan sebaliknya.



Meskipun JSON berasal dari Javascript, JSON menjadi format pertukaran data antar *platform*, antar sistem, atau antar aplikasi menggunakan API (*Application Programming Interface*). Saat ini format JSON lebih dipilih oleh pengembang aplikasi dibandingkan format lainnya seperti XML dan PlainText karena kepraktisannya.

#### **2.2.14 AJAX**

AJAX merupakan kependekan dari *Asynchronous Javascript and XML* adalah praktik pemrograman untuk membangun *web* dinamis kompleks menggunakan teknologi XMLHttpRequest [22]. Laman *web* yang telah dimuat di peramban web, dapat berkomunikasi dengan server dan melakukan pertukaran data tanpa harus memuat ulang dengan menggunakan AJAX.

#### **2.2.15 API**

API atau Application Programming Interface adalah penghubung antar komputer atau antar program. API merupakan sebuah antarmuka perangkat lunak yang menawarkan layanan untuk perangkat lunak atau program lain. API biasanya disertai dengan dokumen yang disebut spesifikasi API yang memberikan informasi bagaimana menerapkan dan menggunakan API tersebut. Kata “API” bisa merujuk kepada spesifikasi atau kepada penerapannya [23].

#### **2.2.16 Framework Bootstrap**

Bootstrap adalah sebuah *framework* CSS untuk membangun *front-end* berbasis *web*. *Framework* Bootstrap diarahkan untuk pengembangan *front-end* web yang *responsive* dan *mobile-first*. *Framework* ini pertama kali dikembangkan oleh

Mark Otto dan Jacob Thornton di Twitter sebagai framework yang mengedepankan konsistensi untuk setiap perangkat-perangkat internal mereka [24].

Saat ini *framework* Bootstrap sudah mencapai versi 5 pada bulan Mei 2021 dengan menerapkan lisensi MIT yang artinya bebas digunakan baik untuk keperluan komersil maupun non-komersil.

### 2.2.17 SQL

SQL (*Structured Query Language*) bahasa standar untuk sistem pengelolaan basis data relasional atau *Relational Database Management System (RDBMS)*. SQL digunakan untuk menyelesaikan banyak tugas yang berkaitan dengan basis data seperti: mengambil data (*select*), merubah data (*update*), memasukan data baru (*insert*), atau menghapus data (*delete*) [25].

### 2.2.18 RDBMS

RDBMS (*Relational Database Management System*) adalah sebuah sistem pengelolaan basis data relasional menggunakan bahasa SQL [26]. Beberapa contoh RDBMS populer adalah sebagai berikut:

1. MySQL
2. Microsoft SQL Server
3. Oracle
4. Microsoft Access
5. IBM DB2
6. PostgreSQL
7. MariaDB

### 2.2.19 MySQL

MySQL adalah sebuah sistem pengelolaan basis data relasional (RDBMS) dengan kode sumber terbuka (*open source*). MySQL dapat berjalan di banyak *platform* seperti Windows, Linux, macOS, Novell Netware, Oracle Solaris, Symbian, UnixWare, dan sebagainya. MySQL menerapkan dua jenis lisensi yaitu lisensi GPL v2 dan lisensi komersial [27].

### 2.2.20 Nama Domain

Sebuah nama domain adalah identifikasi berupa *string* yang menentukan otonomi administratif, otoritas atau kontrol di *internet*. Nama *domain* digunakan di banyak konteks dalam jaringan komputer untuk alamat spesifik pada aplikasi dan pengalamatan. Secara umum, sebuah nama domain adalah identitas sebuah domain jaringan yang merepresentasikan sebuah alamat IP (*Internet Protocol*) untuk mengakses komputer atau server di internet [28].

Sebuah nama *domain* didapatkan dengan melakukan pendaftaran di pendaftar nama *domain internet* yang disebut *Registrar*. Nama *domain* yang telah didaftarkan, dapat digunakan selama minimal 1 tahun dan dapat diperpanjang di tahun berikutnya. Setelah didaftarkan, nama *domain* dapat diatur untuk menunjuk ke *server* nama (*Name Server*) tertentu yang selanjutnya server nama dapat diatur untuk menunjukan alamat ke IP tertentu sesuai dengan alamat IP *server* yang kita miliki agar *server* dapat diakses menggunakan nama *domain*.

### 2.2.21 Web Server

*Web server* atau *server web* merujuk pada perangkat keras (*hardware*) atau perangkat lunak (*software*), atau keduanya berjalan secara bersama-sama untuk melayani permintaan (*request*) dari peramban *web* (*web browser*) menggunakan protokol HTTP (*Hypertext Transfer Protocol*) [29]. Peramban *web* yang mengirimkan *request* ke *server web*, misalnya meminta *resource* (*file*, gambar, dan sumber lainnya), jika sesuai dan *resource* ditemukan, maka akan direspon oleh *server web* dengan mengirimkan *response* berupa *resource* tersebut. Jika *resource* yang diminta tidak ditemukan, *server web* kan memberikan *status* 404 (*file* tidak ditemukan) ke peramban *web*.

*Server web* yang biasa digunakan saat ini antara lain: IIS (*Internet Information System*), Apache dan Nginx. Di sistem operasi Windows biasanya sudah disertakan IIS yang siap digunakan, namun pada sistem operasi Linux atau macOS, *server web* yang dapat digunakan adalah Apache atau Nginx dengan mengunduhnya lalu melakukan instalasi terlebih dahulu. Setelah melakukan instalasi *server web*, pengguna harus melakukan konfigurasi pada sistem servernya, setidaknya menentukan lokasi folder tempat *file-file resource* disimpan dan *port* yang digunakan untuk mengakses *server web* tersebut.

*Server web* secara *default* menggunakan *port* 80 untuk akses HTTP dan *port* 443 untuk akses HTTPS (versi aman dari HTTP menggunakan enkripsi sertifikat pihak ketiga/SSL).

### 2.2.22 Komputasi awan (*cloud computing*)

Komputasi awan merupakan gabungan pendayagunaan teknologi komputer atau komputasi dan pengembangan berbasis internet (awan). Awan (*cloud*) adalah metafora dari internet dan awan sering dijadikan diagram untuk merepresentasikan sebuah jaringan komputer atau *internet*. Komputasi awan adalah suatu metode dimana kapabilitas komputasi terkait teknologi informasi disajikan dalam bentuk layanan (*as a service*), sehingga pengguna dapat mengaksesnya melalui *internet*. Komputasi awan adalah suatu konsep umum yang mencakup perangkat lunak sebagai layanan (*Software as a Service – SaaS*), Web 2.0, dan tren teknologi terbaru lain [30]. Komputasi awan merupakan salah satu tren teknologi informasi saat ini menurut Gartner [31].

Dengan memanfaatkan komputasi awan, kebutuhan komputasi misalnya untuk *server web* dapat dengan mudah ditingkatkan kemampuannya tanpa harus merubah perangkat keras (*hardware*) dan pengembang hanya perlu mempertimbangkan biayanya saja.

Contoh kasusnya adalah jika pengembang mempublikasikan aplikasi berbasis *web*, tentu membutuhkan *server web*. Pada awal peluncuran, *server web* yang dibutuhkan tidak harus menggunakan perangkat keras dengan kemampuan tinggi misalnya cukup dengan 1 prosesor dan 2 Gb RAM. Namun, seiring dengan peningkatan jumlah pengguna, *server web* tersebut akan menjadi tidak memadai dan harus ditingkatkan (dilakukan *upgrade*). Dengan menggunakan komputasi awan, maka peningkatan tersebut hanya perlu dilakukan melalui *dashboard* layanan komputasi awan saja, tanpa harus benar-benar meningkatkan perangkat kerasnya.

Dengan demikian untuk pengembangan sebuah aplikasi yang menargetkan jumlah pengguna yang akan terus meningkat, maka komputasi awan adalah pilihan yang paling efisien.

Beberapa penyedia layanan komputasi awan antara lain: Google (Google Cloud Platform), Amazon (Amazon EZ-II Web Services), dan Microsoft (Microsoft Azure).

### **2.2.23 *Firestore Realtime Database***

*Firestore Realtime Database* adalah *database* NoSQL (tanpa menggunakan bahasa SQL) yang di-*hosting* di *Cloud* dan dapat digunakan untuk menyimpan dan menyinkronkan data antar-pengguna secara *real-time* [32]. *Firestore Realtime Database* berguna untuk pertukaran data *real-time* antar-pengguna pada aplikasi berbasis *web* dan *mobile*. Contoh penggunaan *Firestore Realtime Database* adalah untuk membangun aplikasi permainan dengan banyak pemain (*multi-player game*), aplikasi untuk kolaborasi *real-time* dan aplikasi *chatting* dengan banyak pengguna sekaligus.

*Firestore Realtime Database* merupakan layanan berbasis *cloud* yang gratis dari Google, namun untuk penggunaan dengan skala yang sangat besar, Google memberikan opsi berbayar (*paid service*). Pengembang perlu mendaftarkan diri terlebih dahulu untuk menggunakan layanan *Firestore Realtime Database*.

### **2.2.24 *Firestore Cloud Messaging***

*Firestore Cloud Messaging* adalah solusi pertukaran pesan lintas *platform* yang dapat digunakan untuk mengirim pesan secara aman tanpa biaya. FCM dapat

dimanfaatkan untuk mengirimkan notifikasi ke aplikasi secara *real-time* untuk notifikasi pengingat atau notifikasi apa pun yang akan langsung tampil di sisi pengguna baik pengguna *web* maupun pengguna *mobile* [33].

#### **2.2.25 Jitsi Meet**

Jitsi Meet adalah aplikasi WebRTC Javascript dengan kode sumber terbuka (*open source*) yang dapat digunakan terutama untuk konferensi *video* (*video conferencing*). Selain *audio-video*, Jitsi juga memiliki fasilitas berbagi layar (*screen sharing*). *Server* Jitsi dapat berjalan di sistem operasi linux dan dapat di-*download* dan disesuaikan secara gratis [34].

#### **2.2.26 Ubuntu Server**

Ubuntu adalah sistem operasi distribusi linux berbasis Debian yang terdiri dari perangkat-perangkat lunak kode sumber terbuka (*open source*) yang sangat populer untuk komputasi awan [35]. Ubuntu tersedia dalam beberapa edisi yaitu: *Server*, *Desktop* dan *Core* untuk penerapan *Internet of Things (IoT)* dan robotik. Semua edisi dapat berjalan sendiri pada komputer atau pun pada mesin *virtual* (*virtual machine*).

#### **2.2.27 Webmin**

Webmin adalah alat pengkonfigurasi sistem berbasis web untuk sistem operasi linux dan sejenisnya. Webmin memudahkan pengguna dalam mengkonfigurasi sistem operasi seperti *users*, *quota* penyimpanan, *services* dan juga untuk mengendalikan atau menkonfigurasi aplikasi yang dipasang di sistem operasi misalnya MySQL, Apache HTTP Server, PHP, Samba dan lain sebagainya [36].

### **2.2.28 Android**

Android adalah sistem operasi berbasis linux yang dimodifikasi untuk perangkat bergerak (*mobile*) seperti telepon pintar (*smartphone*), *tablet*, *smart-tv*, *smart-watch* dan perangkat lainnya. Pada awalnya android dikembangkan oleh Android Inc., namun pada tahun 2005, Android diakuisisi dan dikembangkan lebih lanjut oleh Google Inc. [37].

Android adalah sistem operasi yang terus berkembang yang ditandai dengan peningkatan versi dan level API-nya. Level API Android dinyatakan dalam satu bilangan unik yang menunjukkan versi Android yang sedang berjalan pada perangkat pengguna. Artinya pengembang harus memperhatikan level API tersebut dalam membuat aplikasi Android menyesuaikan dengan target pengguna aplikasinya. Jika target pengguna menggunakan level API yang lama maka pengembang juga harus menentukan dalam pengembangan aplikasinya agar dapat mendukung level API yang lama pula (bukan hanya level API terbaru) [38].

### **2.2.29 Android Studio**

*Android Studio* adalah lingkungan pengembangan terpadu atau *IDE (Integrated Development Environment)* resmi dari Google untuk membangun dan mengembangkan aplikasi *Android*. Sebelumnya, aplikasi *Android* dibangun dan dikembangkan menggunakan Eclipse dan ADT (*Android Development Tools*) [39].

### **2.2.30 Visual Studio Code**

*Visual Studio Code* atau biasa disebut VSCode adalah alat penyunting kode program (*code editor*) yang dibuat oleh Microsoft yang dapat digunakan untuk



menyunting banyak bahasa pemrograman dan script diantaranya: HTML, XML, JSON, CSS, Javascript, TypeScript, Java, Python, Go, NodeJS, C++ dan lain sebagainya. *Visual Studio Code* memiliki banyak fitur yang memudahkan *programmer* dalam menulis dan menyunting kode program seperti: *syntax highlighting, bracket matching, code folding, code snippets, intellisense* dan memungkinkan penambahan fitur pihak ketiga [40].