BAB II
TINJAUAN PUSTAKA

2.1. Kerjasama

2.1.1 Asal Usul Kerjasama

Pada Pada dasarnya, kerja sama menandakan kehadiran banyak entitas terlibat untuk mendapatkan keuntungan bersama. Secara teoritis, konsep kerja sama telah dikenal sejak lama dan telah dikonseptualisasikan sebagai sumber efisiensi dan keunggulan layanan. Kolaborasi telah mendapatkan pengakuan sebagai sarana ampuh untuk meningkatkan skala ekonomi (Nurdelila, 2022). Pembelanjaan Terlibat dalam berbelanja atau melakukan pembelian secara kolektif, misalnya, telah menunjukkan manfaat ini, di mana skala yang lebih besar atau mencapai "titik ambang" dalam pembelian menjadi lebih menguntungkan daripada skala yang lebih kecil. Pendekatan kolaboratif ini membantu mengatasi biaya overhead bahkan skala kecil. Berbagi dalam saham.

Selain itu, kolaborasi memiliki potensi untuk meningkatkan kualitas layanan. Pertimbangkan contoh pengadaan fasilitas, di mana pihak individu mungkin tidak mampu membelinya secara mandiri. Melalui kolaborasi, fasilitas layanan yang tadi harga umum bisa di pakai dengan harga yang miring. Kolaborasi melibatkan partisipasi bersama dari banyak individu, mengambil berbagai bentuk, semuanya diarahkan demi keuntungan bersama. Pelaksanaan dilakukan dalam kolaborasi dapat ditentukan kesepakatan bersama.

Kolaborasi mencakup upaya terkoordinasi yang dilakukan oleh dua atau lebih individu, bekerja sama secara terpadu menuju target atau tujuan tertentu (Shalahuddin, 2018).

Kerja sama didefinisikan sebagai "pengaturan sukarela di mana dua atau lebih entitas atau individu terlibat dalam pertukaran yang saling menguntungkan alih-alih bersaing." Dalam istilah yang lebih sederhana, kerjasama menggabungkan fitur koordinasi sumber daya (Shalahuddin, 2018).

(Shalahuddin, 2018).

2.1.2 Derajat Kerjasama

Kerjasama memiliki tingkatan yang berbeda dari koordinasi dan kerjasama ke tingkat yang lebih tinggi yaitu kerjasama. Para sarjana umumnya sepakat bahwa perbedaannya terletak pada kedalaman, integrasi, keterlibatan dan kompleksitas komunikasi, dengan kolaborasi pada level terendah dan kolaborasi pada level tertinggi. (Thomson dan Perry dalam (Muh Caesar Kurniaji, 2016). Satu pihak bekerja sama dan merencanakan, sementara kedua pihak bekerja sama dan merencanakan.

2.2 *E-dokumen*

Dokumen elektronik mencakup informasi elektronik yang dihasilkan, ditransmisikan, dipertukarkan, diterima, atau disimpan dalam berbagai format seperti analog, digital, elektromagnetik, optik, atau media serupa. Dokumen-dokumen ini dapat diakses melalui sistem komputer, memungkinkan keterlibatan visual atau pendengaran. Dengan demikian, klasifikasi langsung dokumen elektronik termasuk dalam kategori informasi elektronik yang lebih luas. Semua dokumen elektronik secara inheren memenuhi syarat sebagai informasi elektronik,

tetapi kebalikannya tidak berlaku secara universal. Ada kemungkinan informasi elektronik yang tidak memenuhi kriteria diklasifikasikan sebagai dokumen elektronik, meskipun skenario ini jarang terjadi. (Shirdata : 2018)

Dokumen elektronik merupakan konten elektronik yang disajikan sebagai program komputer atau file, yang memerlukan media elektronik atau teknologi tampilan untuk digunakan, dibaca, atau dilihat. (Muhammad Fadhil Kusuma Wardana, 2020).

2.3 *Sistem Informasi*

Menggambarkan konsep sistem informasi, ini melibatkan penggabungan individu, sumber daya atau teknologi, media, protokol, dan pemeriksaan, semuanya ditujukan untuk mengatur jaringan komunikasi penting, proses spesifik, dan transaksi reguler. Komposisi ini melayani tujuan membantu manajemen dan pengguna internal dan eksternal, melengkapi mereka dengan landasan untuk membuat keputusan yang tepat (Hadion Wijoyo, 2021).

Sistem informasi merupakan kumpulan komponen yang saling berhubungan yang tugas utamanya mengolah, menyusun, menyimpan dan mendistribusikan informasi. Semua bertujuan untuk mendukung pengambilan keputusan dan kontrol dalam organisasi. Sistem ini terdiri dari informasi yang bermakna tentang individu, lokasi, dan elemen dalam organisasi dan konteks sekitarnya (Hadion Wijoyo, 2021).

Mendefinisikan sistem informasi melibatkan serangkaian tindakan terorganisir yang, ketika dieksekusi, menghasilkan informasi untuk mendukung pengambilan keputusan dan mekanisme kontrol internal (Garuda Ginting, 2022).

2.4 *Database phpMyAdmin*

Database terdiri dari kompilasi data terstruktur dan disimpan dengan cara yang menghilangkan redundansi. Repositori digital ini memungkinkan pemrosesan yang cepat dan mudah, memfasilitasi pembuatan informasi yang diperlukan. *Database* adalah penyimpanan terstruktur yang terintegrasi pada suatu komputer yang mengumpulkan beberapa hal seperti berikut (Tri Rachmadi, 2020).

Aplikasi web yang dibuat oleh phpMyAdmin.net berfungsi sebagai database MySQL. Dalam menjalankan berbasis pemograman PHP yang dioperasionalkan melalui browser sebagai pengelolahan database nya (Sa'ad, 2020).

2.5 *Application berbasis web*

2.6.1. *Definisi Application* berbasis *web*

Application berbasis *Web* merupakan perangkat lunak komputer yang diakses dengan web browser. Dalam pembuatannya, aplikasi web dibuat menggunakan bahasa pemograman yang mendukung perangkat lunak berbasis web seperti HTML, JavaScript, CSS, Ruby, Php, Java dan bahasa pemograman lainnya. Keunggulan menggunakan aplikasi berbasis web yaitu kinerja komputer lebih ringan karena diakses melalui browser tidak seperti aplikasi destop dimana pengguna harus menginstal perangkat lunak tersebut (Ariandi Nugroho, 2021).

2.6.2. Elemen *Application* berbasis *web*

Application berbasis *Web* dibangun dari berbagai elemen yang saling mendukung, berikut ini beberapa elemen elemen pendung dalam pembuatan *application* berbasis web :

2.5.3.1. HTML

HTML *(HyperText Markup Language)* adalah bahasa pemograman yang digunakan untuk membuat halaman web. HTML disebut dengan tag, yang memiliki fungsi yang berbeda-beda dan aturan penulisan yang sudah ditetapkan. Penulisan HTML sifat *case insensitive* yaitu besar kecilnya huruf tidak akan berpengaruh (Sa'ad, 2020). Dalam pembuatannya, HTML memiliki struktur dasar yang terdiri dari :

- Tag *Doctype*
- Tag HTML
- Tag *Head*
- Tag *Body*

Struktur di atas merupakan struktur HTML paling dasar. Berikut ini contoh *script* HTML((Enterprise, 2016):

```
<!DOCTYPE HTML>
<Html>
        <Head>
            <Title> </Title>
        </Head>
    <Body>
    </Body>
</Html>
```

2.5.3.2. PHP

PHP (*Hypertext Preprosesing)* merupakan bahasa *script* yang biasa digunakan dalam pembuatan aplikasi berbasis web. Dalam penggunaannya, PHP menjadi salah satu bahasa *script* ter- populer dalam dalam penggunaannya karena PHP dapat disisipkan ke dalam HTML dengan menggunakan tag-tag PHP.

Dalam pembuatannya, PHP memiliki struktur penulisan yaitu *script* harus di awali dengan tanda <? dan diakhiri tanda ?> , berikut ini contoh penulisan PHP (Ariandi Nugroho, 2021):

<?php **echo " ini adalah script php";** ?>

PHP dapat membuat keterangan atau komentar di dalam *script* PHP dan komentar tersebut tidak akan di proses, berikut ini *script* untuk membuat komentar :

1. Untuk membuat komentar lebih dari satu baris dapat menggunakan tag /* dan diakhiri dengan tag */ .
2. Untuk membuat komentar hanya terdiri dari satu baris dapat menggunakan tag //; .
3. Untuk membuat komentar hanya satu baris dapat menggunakan tag.

2.5.3.3. CSS

CSS (*Cascading Style Sheets*) berdiri sebagai bahasa stylesheet yang digunakan untuk mengatur susunan visual halaman *web* yang dihasilkan menggunakan HTML dan XHTML. Untuk diterapkan pada HTML, CSS menggunakan kode-kode terstruktur yang biasa disebut dengan *class* (Sa'ad, 2020).

CSS berfungsi secara langsung untuk menentukan atribut seperti jenis *font*, warna, dan format lainnya, karena memerlukan gaya khusus. Setiap gaya terdiri dari dua komponen dasar seperti yang dijelaskan di bawah ini (Ariandi Nugroho, 2021):

1. *Selector*

Pemilih menunjukkan elemen HTML biasanya dimulai dengan simbol *hash* "#" atau titik..

2. *Declaration*

Declaration mencakup sekumpulan perintah/nilai CSS yang diapit dalam tanda kurung kurawal "{ }" dan dipisahkan dengan titik koma ";" antara perintah.

2.5.3.4. *JavaScript*

JavaScript merupakan bahasa pemograman yang berfungsi memproses halaman web seperti mem-*validari* apa yang diketikan pengguna ke dalam *form* sebelum pengiriman *form* ke *database* dilakukan (Sa'ad, 2020). Berikut ini contoh dari penulisan *JavaScript :*

```
<!DOCTYPE HTML>
<Html>
        <Head>
            <Title> JavaScript </Title>
        </Head>
    <Body>
    <script type=”text/javasript”>
    Document.write(“\”Hello Word \”\n”);
    </script>
    </Body>
</Html>
```

JavaSript memiliki kontruksi – kontruksi dasar dalam pembuatannya yaitu:

- **Tipe Data**

Program dapat melakukan seperti perhitungan, pengurangan, dan lain lainya, namun agar program dapat melakukan hal tersebut dibutuhkannya data. Tipe data dapat menentukan apakah data tersebut sebuah angka atau karakter sehingga data akan disimpan dan diproses di dalam sebuah program (Tri Rachmadi, 2020)

- **Variabel**

Variabel merupakan kontainer yang menampung data seperti angka dan *string*. *Variabel* memiliki tiga tipe data yaitu *numerik, string* , dan *boolean* (Tri Rachmadi, 2020) .

2.6.3. *Tools* pendukung Aplikasi Web

2.5.3.1. *Sublime Text* 3

Sublime Text 3 merupakan aplikasi terbaik untuk menulis kode program yang dapat dipasang dan digunakan pada sistem operasi *Windows*, OS X, dan *Linux*. Aplikasi *sublime* dapat diunduh melalui situs *www.sublimetext.com* (Ica, 2018)

2.5.3.2. *Framework CodeIgniter*

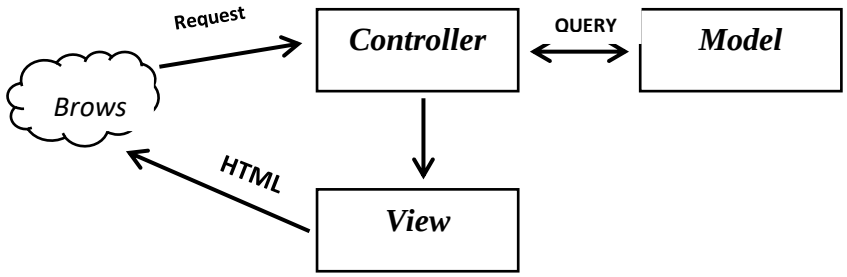
CodeIgniter merupakan aplikasi *open source* yang memudahkan *developer* untuk membuat aplikasi web dengan cepat. Hal ini karena *CodeIgniter* dalam penggunaanya memiliki model MVC (*Model, View, Conttroler*) dalam membangun web dinamis dengan menggunakan PHP (Supono & Putratama, 2018).

Model MVC pada *framework CodeIgniter* adalah kumpulan fungsi – fungsi dan *class* – *class* yang bertujuan mempermudah dan mempercepat seorang programer dalam membuat web tanpa harus membuat fungsi dan *class* dari awal. MVC bekerja dengan cara memisahkan aplikasi berdasarkan komponen utama sehingga dalam penyusunan struktur program lebih tersusun. Terdapat 3 jenis komponen pada MVC dalam membangun aplikasi yaitu (Supono & Putratama, 2018)):

1. *View*, merupakan bagian yang menangani *presentation logic* yaitu bagian file template HTML, yang diatur oleh *conttroler* sedangkan *View* tidak memiliki akses langsung pada bagian *Model*.
2. *Controller,* merupakan bagian untuk mengatur hubungan antaran *model* dan *view*. *Controller* berfungsi untuk menerima *request* dari data *user* sehingga data *user* akan di proses oleh aplikasi.

3. *Model*, merupakan bagian yang terhubung dengan *database* untuk memanipulasi data seperti *insert, update,* dan *dalete.*

Berikut ini bentuk arsitektur aplikasi web jika dikembangkan menggunakan model MVC :

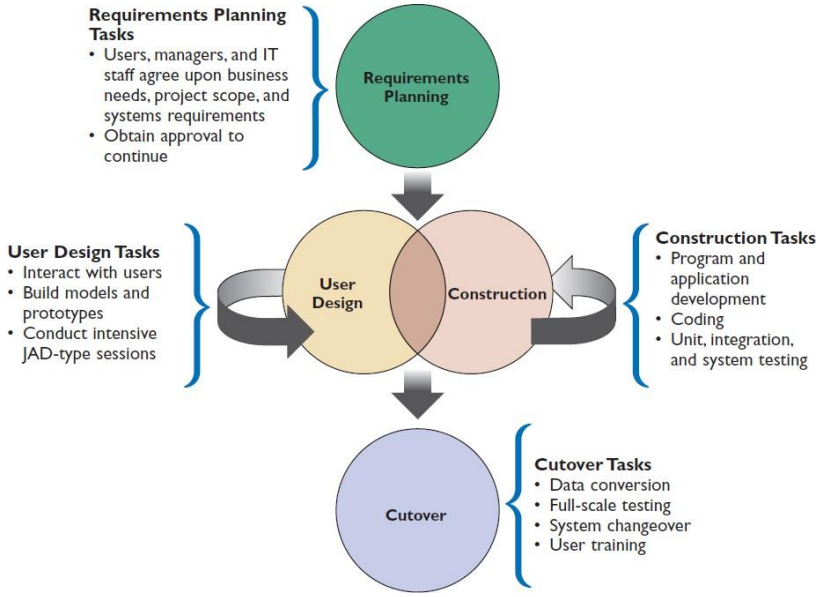


Gambar 2.1 Model MVC (*Model, View, and Conttroler*)

2.6 Metode Pengembangan Sistem

2.6.1. Metode RAD (*Rapid Application Development*)

Metode RAD (Rapid Application Development) merupakan salah satu pendekatan yang digunakan dalam pengembangan sistem atau aplikasi, yang ditandai dengan orientasi objek dan pemrosesan aplikasi yang cepat dan akurat. Keuntungan penting dari metode ini adalah periode penggunaannya yang sangat singkat, yang biasanya bervariasi dari 30 hingga 90 hari. Sebaliknya, metode pengembangan tradisional biasanya membutuhkan setidaknya 180 hari untuk mencapai hasil yang sama. (Rukmana & desiyani, 2017).



Gambar 2.2 Tahapan Metode RAD

Pemilihan model RAD untuk penelitian ini mempertimbangkan keunggulan model RAD dibandingkan dengan model perangkat lunak lainnya. Keuntungan dari model ini adalah memungkinkan siklus pengembangan sistem yang cepat, kualitas sistem yang lebih baik dibandingkan dengan model perangkat lunak tradisional, dan biaya pengembangan dan pemeliharaan sistem yang lebih rendah. (Ndjurumana, Evangs Mailoa, 2020). Pendekatan RAD terdiri dari empat fase: Perencanaan Kebutuhan, Desain Pengguna, Konstruksi, dan Pemotongan. Berikut ikhtisar dari setiap tahapan dalam metodologi RAD (Rosenblatt & Shelly, 2016):

2.5.3.1. *Requirements planning*

Fase ini melibatkan analisis masalah sistem yang ada untuk mengidentifikasi prasyarat sistem. Proses ini dilakukan melalui wawancara atau observasi, yang dijabarkan sebagai berikut:

a) Wawancara

Wawancara adalah teknik yang dilakukan untuk mengumpulkan informasi dalam

pengambilan data, dengan cara bertanya langsung kepada narasumber, sehingga lebih akurat dan terpercaya (Dr. Muhammad Ilyas Ismail, 2020).

b) Observasi

Observasi merupakan cara untuk mengumpulkan data atau informasi dengan mengamati secara langsung ke tempat yang akan diamati (Dr. Muhammad Ilyas Ismail, 2020).

2.5.3.2. *User design*

Fase ini melibatkan perumusan desain sistem atau aplikasi, disertai dengan proses implementasi selanjutnya. Selama tahap ini, desain dibuat, meliputi aliran data sistem, hubungan basis data, tinjauan sistem, dan antarmuka aplikasi. Komponen selanjutnya dihasilkan selama fase desain sistem:

a. DFD (*Data Flow Diagram)*

DFD (Data Flow Diagram) menggambarkan aliran informasi pada sistem yang ada dan yang baru dikembangkan, menggunakan kerangka aliran logis yang tidak memasukkan elemen lingkungan eksternal. Keuntungan menggunakan DFD terletak pada kemampuannya untuk memfasilitasi pemahaman bagi individu yang tidak memiliki keahlian di bidang komputer, memungkinkan mereka memahami seluk-beluk operasional sistem. (Wahyudi Agustiono, 2019). Proses untuk membuat DFD dibagi menjadi 3 level, yaitu :

1. Diagram konteks

Proses lengkap yang dijalankan dalam sistem atau aplikasi direpresentasikan secara visual dalam bentuk gambar melingkar yang komprehensif. Diagram konteks, terletak di puncak hirarki DFD (level-0), menawarkan ikhtisar dari semua entitas eksternal sambil mencakup aliran data primer yang masuk dan

keluar dari sistem.

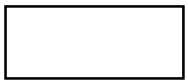
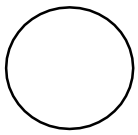
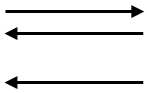
2. Diagram nol (diagram level-1)

Grafik nol adalah bagian bawah grafik konteks yang berisi elemen penyimpanan data. Dalam diagram nol, lingkaran besar tunggal merangkum lingkaran yang lebih kecil di dalamnya, melambangkan hubungan hierarkis.

3. Diagram rinci

Diagram rinci merupakan proses yang berasal dari diagram nol.

Tabel 2.1 Komponen DFD

Yourdon/De Marco	Keterangan
	Entitas eksternal dapat berupa orang/unit terkait yang berinteraksi dengan sistem, tetapi diluar sistem.
	Orang/unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasikan.
	Penyimpanan data atau tempat data di refer oleh proses.
	Aliran arah data dari arah khusus dan dari sumber ke tujuan.

b. ERD (*Entity Relationship Diagram*

ERD berfungsi sebagai representasi visual yang dirancang untuk menggambarkan database dari sudut pandang pengguna. Ini menguraikan konstituen penting dari database, yang meliputi entitas, atribut, dan hubungan, semua diwakili melalui metode notasi tertentu. Beberapa notasi berbeda digunakan untuk

mendeskripsikan ERD, termasuk Notasi Chen asli, Crow's Foot, dan Notasi UML. (Supuwiningsih, 2021).

Berikut ini adalah penjelasan relasi dari entitas beserta contoh dalam penggunaan *Crow's Foot Notation*.

1. Relasi *One-to-many* (1:M)

Relasi 1:M, menjelaskan relasi suatu entitas yang dikaitkan dengan banyak contoh dari entitas terkait. Contoh relasi 1:M dapat dilihat pada Gambar 2.3.

Gambar 2.3 Relasi 1:M

2. Relasi *Many-to-many* (M:N)

Relasi M:N, menjelaskan relasi antara dua atau lebih entitas di mana satu kejadian dari suatu entitas dikaitkan dengan banyak kejadian dari entitas terkait dan satu kejadian dari entitas terkait dikaitkan dengan banyak kejadian dari entitas pertama. Contoh relasi M:N dapat dilihat pada Gambar 2.4.

Gambar 2.4 Relasi M:N

3. Terhubung *One-to-one* (1:1)

Relasi 1:1, menjelaskan antara dua atau lebih entitas. Dalam relasi 1:1, satu entitas dikaitkan hanya dengan satu entitas terkait. Contoh relasi M:N dapat dilihat pada Gambar 2.5.

Gambar 2.5 Relasi 1:1

c. *Flowchart*

Flowchart terdiri dari serangkaian langkah atau urutan prosedural yang berasal dari proses dan logika aplikasi, yang disajikan secara sistematis melalui citra grafis. Tujuan utama flowchart membantu analis dan pengembang untuk membedakan masalah dan cara kerja aplikasi.(Indra Rianto, 2023). Simbol *flowchart* dapat dilihat pada tabel 2.

Tabel 2.2 Simbol Flowchart

Simbol	Nama	Fungsi
	Terminator	Permulaan/akhir program
	Garis alir (*Flow line*)	Arah aliran program
	Preparation	Proses inisialisasi/pemberian harga awal
	Proses	Proses perhitungan /proses pengolahan data
	Input/output data	Proses *input/output* data, parameter, informasi
	Predefined process (Sub program)	Permulaan sub program/proses menjalankan sub program
	Decision	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	On page connector	Penghubung bangian-bagian *flowchart* yang berada pada satu halaman
	Off page connector	Penghubung bagian-bagian *flowchart* yang berada pada halaman berbeda

2.5.3.3. *Construction*

Selama tahap konstruksi, sistem atau aplikasi dikembangkan secara bertahap berdasarkan cetak biru. *Fase* ini melibatkan *coding* aplikasi, melakukan pengujian, dan menerapkan penyesuaian yang diperlukan.

2.5.3.4. *Cutover*

Selama fase cutover, proses pengujian aplikasi yang komprehensif sedang dilakukan. Aplikasi sistem yang baru dikembangkan ini menjalani pengujian dengan menggunakan metode Black Box Testing di lingkungan operasional. Metodologi ini digunakan untuk mengidentifikasi dan memperbaiki kesalahan perangkat lunak yang mungkin muncul selama pengembangan, memastikan bahwa aplikasi yang diperbaiki dapat digunakan dengan lancar oleh pengguna.

Black Box Testing memfokuskan ulasannya pada spesifikasi fungsional perangkat lunak. Seorang penguji dapat menjelaskan sekumpulan kondisi input dan kemudian mengevaluasi informasi fungsional program. Pengujian Kotak Hitam tidak hanya menggantikan Pengujian Kotak Putih, melainkan melengkapinya untuk mengatasi aspek yang mungkin tidak dicakup oleh Pengujian Kotak Putih. Biasanya, Pengujian Kotak Hitam digunakan untuk mendeteksi masalah. (Agustian, 2022).