

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Menurut (Rahmadia,2021) dalam penelitiannya yang berjudul “ **Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) Berbasis Web Pada SMK Pertiwi** ” menjelaskan bahwa Penerimaan Peserta Didik Baru (PPDB) merupakan langkah awal dalam perjalanan pengembangan sistem Penerimaan Peserta Didik Baru (PPDB) secara online di sekolah. Menegaskan bahwa adopsi teknologi dalam proses Penerimaan Peserta Didik Baru (PPDB) menjadi kunci penting untuk meningkatkan efisiensi, transparansi, dan akuntabilitas. Pengamatan mereka terhadap proses Penerimaan Peserta Didik Baru (PPDB) di sekolah swasta yang masih dilakukan secara manual menunjukkan berbagai kendala, termasuk administrasi yang lambat dan risiko kecurangan. Ini menunjukkan urgensi untuk mengimplementasikan solusi berbasis teknologi guna mengatasi permasalahan tersebut dan meningkatkan kualitas serta profesionalisme dalam manajemen pendidikan.

Menurut (Rahayu,2022) dalam penelitiannya yang berjudul “ **Perancangan Sistem Informasi Pendaftaran Online Berbasis Web Pada Paud Assibyan Serang Banten** “ menjelaskan bahwa pengembangan teknologi informasi yang mempengaruhi kebutuhan informasi cepat dan mudah, terutama dalam bidang Pendidikan, pembangunan sistem informasi berbasis website menjadi suatu keharusan. Pada kasus Paud Assibyan di Serang Banten, proses pendaftaran yang masih manual menyebabkan ketidakakuratan dan ketidakefisien data, terutama dalam situasi seperti pandemic yang membatasi mobilitas. Oleh karena itu, penelitian yang mengusulkan sistem informasi pendaftaran online berbasis web pada Lembaga tersebut menjadi sangat relevan. Dengan metode waterfall dan menggunakan framework Laravel, sistem ini dirancang untuk memenuhi kebutuhan orang tua calon peserta didik dan admin Paud, dengan harapan dapat meningkatkan efisiensi, akurasi, dan kemudahan dalam pengelolaan data pendaftaran. Melalui implementasi yang baik, diharapkan sistem ini dapat memberikan solusi yang

sesuai dengan kebutuhan dan memberikan manfaat maksimal bagi semua pihak yang terlibat.

Menurut (Ernawati,2022) dalam penelitiannya yang berjudul “**Perancangan Sistem Informasi Pendaftaran Murid Baru Pada Paud Kangguru Kecil Berbasis Web**” menjelaskan bahwa dalam era digital yang terus berkembang, teknologi canggih telah menjadi bagian integral dari kehidupan sehari-hari dan kegiatan Pendidikan. Penggunaan teknologi dalam mengelola administrasi sekolah dapat mempermudah proses kegiatan dan memberikan informasi yang bermanfaat, cepat, akurat, dan efektif. Namun masih banyak masyarakat yang belum memanfaatkan inovasi dan teknologi yang ada, terutama dalam mengelola penerimaan murid baru setiap tahunnya. Masalah pendaftaran masih menggunakan proses manual dan penyebaran informasi yang belum efisien, seperti media cetak brosur dan flyer. Untuk mengatasi masalah ini, dirancang sebuah sistem berbasis web menggunakan metode SDLC Waterfall. Sistem ini bertujuan untuk mempermudah proses pendaftaran dan mengurangi kehilangan data. Melalui ini diharapkan sekolah dapat lebih efisien dalam menyebarkan informasi dan mengelola proses pendaftaran murid baru dengan lebih baik.

Menurut (Febriansyah,2022) dalam penelitiannya yang berjudul“**Implementasi Metode Agile Pada Aplikasi Sistem Pendaftaran Siswa Baru Berbasis Web Dengan Notifikasi Whatsapp (Studi Kasus : Paud Nurussyifa)**” menjelaskan bahwa peran teknologi dalam kehidupan sehari-hari semakin meningkat pesat, terutama dalam bidang informasi dan komunikasi. Berkembangnya teknologi informasi, terutama dalam pengembangan aplikasi web, telah memberikan dampak positif dalam memfasilitasi berbagai aspek kehidupan, termasuk dalam bidang Pendidikan. Dalam Pendidikan Anak Usia Dini memiliki kebutuhan akan sistem informasi yang terkomputerisasi dalam proses pendaftaran yang masih menggunakan sistem konvensional, yang memakan waktu dan dapat menyebabkan redundansi data serta keterlambatan dalam pengolahan informasi. Oleh karena itu, pengembangan sebuah aplikasi yang memudahkan proses pendaftaran siswa baru menjadi sangat penting untuk meningkatkan efisiensi akan akurasi dalam pengelolaan data pendidikan.

2.2 Aplikasi

Aplikasi adalah sebuah perangkat lunak yang berisi sebuah coding atau perintah yang dimana bisa diubah sesuai dengan keinginan. Aplikasi adalah sebuah perangkat lunak yang dimana tujuannya adalah agar bisa melayani setiap aktivitas komputerisasi yang dilakukan oleh pengguna (Werstantia, 2018).

Aplikasi merupakan suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Aplikasi dibuat untuk memudahkan pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan, dan penambahan data yang dibutuhkan. (Munari, 2020).

2.3 Pendaftaran

Menurut (Suhendar,2020) Pendaftaran adalah pencatatan hal atau identitas seperti nama, alamat dan sebagainya dalam suatu lembaga pendidikan, pendaftaran merupakan suatu hal yang sangat penting. Pengertian pendaftaran disini pada dasarnya hanya untuk memperlancar dan mempermudah dalam pendaftaran sehingga terorganisir, teratur dengan cepat atau tepat.

2.4 Siswa Baru

Menurut (Kuncoro,2010) Siswa Baru adalah individu yang baru saja mulai mengikuti proses pendidikan di suatu lembaga pendidikan tertentu. Mereka biasanya mengalami masa transisi dari lingkungan sebelumnya ke lingkungan baru.

2.5 Website

Website adalah kumpulan halaman *web* yang dipublikasikan di Internet dengan nama domain/alamat URL (*Uniform Resource Locator*) yang dapat diakses oleh setiap pengguna Internet dengan mengetikkan alamatnya. Halaman *web* biasanya adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*) dan diakses melalui HTTP. HTTPS adalah protokol yang mentransmisikan berbagai informasi dari *server* suatu *website* yang ditampilkan

kepada pengguna atau pengguna melalui *browser* (Nofyat et al, 2018).

Website atau sering disingkat *web*, adalah kumpulan halaman *web* dengan topik terkait, terkadang disertai dengan *file* gambar, *video*, atau jenis *file* lainnya. Situs *web* biasanya *dihosting* di *server web* yang dapat diakses melalui jaringan seperti *Internet* atau jaringan area lokal (LAN) di alamat *Internet* yang dikenal sebagai URL. Kombinasi dari semua halaman yang dapat diakses publik di Internet juga disebut sebagai WWW(Larno et al, 2019).

Website adalah kumpulan dari halaman–halaman situs, yang terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web* (WWW) di dalam Internet. Sebuah halaman *web* biasanya berupa dokumen yang ditulis dalam format *Hyper Text Markup Language* (HTML), yang selalu bisa diakses melalui HTTP, yaitu sebuah protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser* (Nevy,2022).

2.6 XAMPP

XAMPP ialah software yang di dalamnya terdapat server *MySQL* dan didukung oleh *PHP* sebagai bahasa pemrograman untuk membuat *website* dinamis serta terdapat web server apache yang dapat dijalankan di beberapa platform seperti *OS X*, *Windows*, *Linux*, *Mac*, dan *Solaris*. XAMPP merupakan software server apache dimana dalam XAMPP yang telah tersedia *database server* seperti *MySQL* dan *PHP programming*. XAMPP memiliki keunggulan yaitu cukup mudah dioperasikan, tidak memerlukan biaya serta mendukung instalasi pada *Windows* dan *linux* (Fauzi , 2018).



Gambar 2.1 Logo XAMPP.

2.7 PHP MyAdmin

Dengan menggunakan *MySQL*, pengeolaan database dilakukan dengan mengetikkan baris-baris perintah (misalnya menggunakan *MySQL console*) yang sesuai untuk keinginan tertentu, jika kita ingin membuat database dan memanipulasi table, maka kita harus mengetikkan baris perintah yang sesuai. Hal diatas tersebut cukup sulit dilakukan karena itu harus menghafal semua sintak dan harus mengetikkan satu persatu baris sintak tersebut. Oleh karena itu, diperlukan perangkat lunak tambahan yang disebut dengan PHPMyAdmin. Dengan PHPMyAdmin, pengelolaan atau memanipulasi database jadi lebih mudah karena tidak perlu menghafal sintaknya (Wahyudi, 2015).



Gambar 2.2 Logo *phpMyAdmin*

2.8 Basis Data (*Database*)

Database adalah sebuah *system* yang dibuat untuk mengorganisasi, menyimpan dan menarik data dengan mudah. *Database* terdiri dari kumpulan data yang terorganisir untuk 1 atau lebih penggunaan, dalam bentuk digital. *Database* digital di-*manage* menggunakan *Database Management System* (DBMS), yang menyimpan isi *database*, mengizinkan pembuatan dan *maintenance data* dan pencarian dan akses yang lain. Beberapa *Database* yang ada saat ini adalah : *MySQL*, *SQL Server*, *Ms.Access*, *Oracle*, dan *PostgreSQL* (Istilah yang sering digunakan pada pembahasan *database*, yaitu:

1. DBMS, yaitu sistem perangkat lunak yang menyediakan layanan bagi *user* untuk membuat, mengontrol, dan mengakses *database*.
2. *Table*, merupakan kumpulan data yang diorganisasikan dalam baris

(*record*) dan kolom (*field*). Dalam satu *database* biasanya terdiri dari beberapa tabel.

3. *Field*, merupakan kolom pada tabel yang memiliki ukuran dan tipe data tertentu.
4. *Record*, merupakan baris data yang memiliki nilai untuk masing-masing *field* pada tabel. Satu tabel biasanya terdiri dari banyak *record*.
5. *Key*, merupakan suatu *field* (kolom) yang dapat dijadikan kunci dalam operasi tabel.
6. *SQL*, merupakan singkatan dari *Structured Query Language*, yaitu bahasa yang digunakan untuk mengakses dan memanipulasi *database* (Abdulloh, 2018).

2.9 Maria DB

(Purbo, 2015) menyatakan bahwa MariaDB adalah pilihan yang baik bagi pengguna yang menginginkan stabilitas dan keandalan dalam manajemen basis data. Ia menyoroti bahwa MariaDB merupakan turunan dari MySQL yang dirancang untuk memberikan kinerja yang lebih baik dan keamanan yang ditingkatkan.

(Yozer, 2018) menyatakan bahwa MariaDB adalah pilihan yang tepat untuk perusahaan yang membutuhkan manajemen basis data dengan tingkat keamanan yang tinggi dan kemampuan untuk beroperasi pada berbagai sistem operasi. Ia juga menyebutkan bahwa MariaDB terus mengembangkan fitur-fiturnya untuk memenuhi kebutuhan pengguna di berbagai industri.



Gambar 2.3 Logo *MariaDB*

2.10 *Visual Studio Code*

(Drasner, 2019) menggambarkan *Visual Studio Code* sebagai editor yang menawarkan pengalaman pengembangan yang terpadu dan lancar. Dia menyoroti kemampuan *Visual Studio Code* untuk mendukung berbagai alat pengembangan modern dan ekosistem plugin yang luas.

Seorang pengembang di komunitas *open-source*, Mads Kristensen menyatakan bahwa *Visual Studio Code* adalah salah satu *editor* kode terbaik yang tersedia karena kemampuannya untuk dikustomisasi dan dukungan ekstensinya. Menurutnya, komunitas yang aktif dan dukungan yang luas menjadikan *Visual Studio Code* pilihan utama bagi banyak pengembang (Kristensen, 2017).

2.11 *HTML (Hyper Text Markup Language)*

HTML (Hypertext Markup Language) merupakan bahasa standar yang dipakai untuk menampilkan konten-konten seperti gambar, suara, video dan lainnya pada halaman *website*. Fungsi-fungsi yang dapat dilakukan dengan bahasa programan *HTML* adalah:

1. Mengatur serta mendesain tampilan isi halaman *website*,
2. Membuat tabel pada halaman *website*,
3. Membuat rancangan formulir yang dapat menjadi masukkan sertamenangani registrasi dan transaksi via *website*,
4. Menampilkan area gambar pada *browser* (Mariko, 2019).

Hypertext Markup Language adalah bahasa *markup* yang digunakan untuk membuat aspek *online*, berbagai informasi saat menjelajah Internet dan format *hypertext* sederhana yang ditulis dalam *file ASCII* untuk menghasilkan layar yang terintegrasi. *File* dimuat ke perangkat lunak komputer dan disimpan dalam format *ASCII* sehingga dapat diubah menjadi halaman *web* menggunakan perintah *HTML* (Putra et al, 2019).

Contoh Skript HTML

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Halaman Sederhana</title>
</head>
<body>
  <h1>Selamat Datang!</h1>
  <p>Ini adalah contoh halaman HTML yang sangat
sederhana.</p>
  <a href="https://www.example.com">Kunjungi Situs Kami</a>
</body>
</html>
```

2.12 PHP (*Personal Hypertext Preprocessor*)

PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis, karena *PHP* merupakan *server-side scripting* maka sintaks dan perintah-perintah *PHP* akan dieksekusi di *server* kemudian hasilnya akan dikirimkan ke *browser* dengan format *HTML*. Dengan demikian kode program yang ditulis dalam *PHP* tidak akan terlihat oleh *user* sehingga keamanan halaman *web* lebih terjamin. *PHP* dirancang untuk membuat halaman *web* yang dinamis, yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web* (Asnawi et al, 2013).

Keunggulan dari *script* PHP adalah :

- a. Cara koneksi dan *query database* yang sederhana
- b. Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemrograman *web* lainnya berorientasi pada *server-side scripting*.
- c. Relatif aman, mengingat *script* atau *source code* PHP yang terdapat pada

web browser tidak dapat dilihat dengan fasilitas *view HTML source*, seperti pada Internet Explorer atau sejenisnya karena *script* PHP diletakkan pada *tag HTML*.

- d. Fleksibel, instalasi sistem operasi mendukung semua jenis *Linux*, *Windows*, *MacOS*, dan jenis *UNIX*. Selain itu, jenis *web server* yang dapat diaplikasikan selain *Apache* adalah *Microsoft Personal Web Server*(PWS), *Netscape*, *PlanetServer*, *Xitami*, *OmniHTTPd*, dan *Microsoft Internet Information Services* (IIS).
- e. Dalam hal keamanan data, *PHP* dapat menggunakan berbagai macam *database*, seperti *MySQL*, *Microsoft Access*, *InterBase*, *mSQL*, *Sybase*, *Dbase*, *IBM Informix*, *Microsoft SQL Server*, dan lain-lain.

Contoh Skript PHP

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Contoh Halaman PHP</title>
</head>
<body>
  <h1>Selamat Datang di Halaman PHP</h1>
  <p>
    <?php
      // Skrip PHP untuk menampilkan pesan selamat
datang dengan tanggal dan waktu saat ini
      echo "Halo! Sekarang tanggal " . date("d-m-Y")
. " dan jam " . date("H:i:s") . ".";
    ?>
  </p>
</body>
</html>
```

2.13 CSS (*Cascading Style Sheets*)

CSS atau akronim dari *Cascading Style Sheets* merupakan satuan atau gabungan kode yang tersinkron dalam HTML dan bertujuan untuk menggantikan gaya atau tampilan dari sebuah halaman *website*. CSS bisa juga digunakan untuk mengatur warna bagian tubuh pada ukuran *border*, *teks*, warna *hyperlink*, warna *border*, spasi antar *paragraf*, warna mouse over, ukuran gambar, warna tebal spasi antara *teks*, *margin* kanan, margin kiri, margin bawah, margin atas dan bagian tubuh lainnya,(Alfian et, al., 2020).

CSS adalah bahasa desain *web* (*Style Sheet Language*) yang mengontrol format tampilan halaman *web* yang ditulis dalam bahasa *markup*. Selain itu, CSS juga dapat menyesuaikan posisi, warna, dekorasi atau apapun yang berhubungan dengan tampilan dokumen HTML (Tampubolon, 2018).

CSS kepanjangan dari *Cascading Style Sheet* adalah bahasa-bahasa yang merepresentasikan halaman web seperti warna, *layout*, dan *font*. CSS dapat mengontrol ukuran gambar, warna, bagian tubuh teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, *spasi* antar *teks*, margin kiri, kanan, atas, bawah, dan parameter lainnya Menurut (Pahlevi et al, 2018).

Contoh Skript CSS

```
body {  
    font-family: Arial, sans-serif;  
    margin: 0;  
    padding: 0;  
    background-color: #f4f4f4;  
}  
h1 {  
    color: #333;  
    text-align: center;  
    margin-top: 20px;  
}  
p {  
    color: #666;  
    padding: 15px;  
    line-height: 1.6;  
}
```

```
a {  
    color: #4CAF50;  
    text-decoration: none;  
}  
a:hover {  
    text-decoration: underline;  
}
```

2.14 Bootstrap

Bootstrap adalah *framework* CSS untuk membuat tampilan *web*. *Bootstrap* menyediakan *class* dan komponen yang sudah siap pakai. *Bootstrap* berisi HTML dan CSS desain berbasis *template* untuk tipografi, bentuk, tombol, navigasi dan komponen antarmuka lainnya, serta opsional *JavaScript* ekstensi. Hal ini bertujuan untuk memudahkan pengembangan dari *website* dinamis dan aplikasi *web* (Setiyawati, 2019).

Bootstrap adalah *framework frontend* yang intuitif dan kuat untuk pengembangan aplikasi *web* yang lebih cepat dan lebih mudah, *Bootstrap* menggunakan bahasa pemrograman berupa HTML, CSS, dan *Javascript* (Hanafri,2018).

Bootstrap adalah *library framework* CSS yang dibuat khusus untuk bagian pengembangan *front-end website*. *Bootstrap* juga merupakan salah satu *framework* HTML, CSS dan javascript yang paling populer di kalangan *web developer* yang digunakan untuk mengembangkan sebuah *website* yang *responsive* (Tampubolon,2018).



Gambar 2.4 Logo *Bootstrap*

Contoh Skript *Bootstrap*

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Contoh Bootstrap</title>
  <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/boo
tstrap.min.css" rel="stylesheet">
</head>
<body>
  <div class="container text-center mt-5">
    <h1>Selamat Datang</h1>
    <p>Ini adalah contoh halaman dengan Bootstrap.</p>
    <a href="#" class="btn btn-primary">Tombol</a>
  </div>
  <script src="https://code.jquery.com/jquery-
3.5.1.slim.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/u
md/popper.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/boots
trap.min.js"></script>
</body>
</html>
```

2.15 Javascript

JavaScript adalah bahasa yang tidak memerlukan kompiler, cukup dengan *interpreter* (Hermiati et al, 2021).

JavaScript ialah suatu bahasa scripting yang digunakan sebagai

fungsionalitas dalam membuat suatu web (Permatasari, 2018).

Javascript adalah bahasa pemrograman *web* yang bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe Bahasa programan yang pemrosesannya dilakukan oleh *Client*. *JavaScript* pada awal perkembangannya berfungsi untuk membuat interaksi antar *user* dengan situs *web* menjadi lebih cepat tanpa harus menunggu pemrosesan di *web server*. Berbagai animasi untuk mempercantik halaman *web*, fitur *chatting*, efek-efek *modern*, *games*, semuanya bias dibuat menggunakan *JavaScript* (Sianipar,2017).

Contoh Skript *Javascript*

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Contoh JavaScript</title>
</head>
<body>
  <div class="container text-center mt-5">
    <h1>Selamat Datang</h1>
    <p id="message">Klik tombol di bawah untuk melihat
pesan.</p>
    <button onclick="showMessage()" class="btn btn-
primary">Klik Saya</button>
  </div>
  <script>
    function showMessage() {
      document.getElementById('message').textContent =
'Halo, ini pesan dari JavaScript!';
    }
  </script>
</body>
</html>
```

2.16 Flowchart

Flowchart atau diagram alir merupakan *chart* (bagan) yang menunjukkan hasil *flow* dalam program atau prosedur sistem secara logika. Digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. *Diagram* ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian.



Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan *algoritma* atau langkah-langkah instruksi yang berurutan dalam sistem. *Flowchart* digambarkan dengan menggunakan simbol-simbol setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung. *Flowchart* dibuat dengan tujuan untuk menunjukkan setiap proses yang harus dilalui dalam suatu sistem (Rosaly et al,2019).

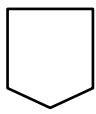
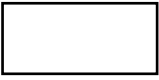
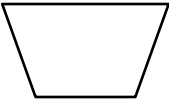
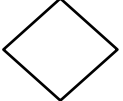
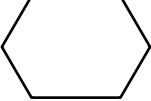
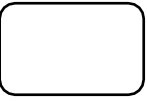
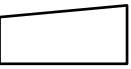
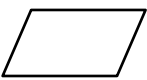

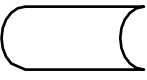


Simbol-simbol yang di pakai dalam *flowchart* dibagi menjadi 3 kelompok:


- 1) *Flow direction symbol*. Digunakan untuk menghubungkan simbol satudengan yang lain. Disebut juga *connecting line*.
- 2) *Processing symbols*. Menunjukkan jenis operasi pengolahan dalam suatuproses / prosedur.
- 3) *Input / Output symbol*. Menampilkan jenis peralatan yang digunakan.

Berikut dibawah ini adalah tabel yang berisi simbol-simbol yang digunakan dalam *flowchart* disertai dengan keterangan fungsinya (Sitorus, 2016).

Tabel 2.1 Simbol-Simbol Flowchart

NO	SIMBOL	KETERANGAN
1		Simbol arus/ <i>flow</i> , berfungsi untuk menyatakan jalannya arus suatu proses
2		Simbol <i>connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama

3		Simbol <i>off-page connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
4		Simbol <i>process</i> , berfungsi untuk menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
5		Simbol <i>manual</i> , berfungsi untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
6		Simbol <i>decision</i> , berfungsi untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya/tidak
7		Simbol <i>predefined process</i> , berfungsi untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
8		Simbol <i>keying operation</i> , berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>
9		Simbol <i>manual input</i> , berfungsi untuk memasukkan data secara manual dengan menggunakan online <i>keyboard</i>
10		Simbol <i>input/output</i> , berfungsi untuk menyatakan proses input atau output tanpa tergantung jenis peralatannya
11		Simbol <i>magnetic tape</i> , berfungsi untuk menyatakan input berasal dari pita magnetis atau output disimpan ke pita magnetis
12		Simbol <i>disk storage</i> , berfungsi untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>
13		Simbol <i>document</i> , berfungsi untuk mencetak keluaran dalam bentuk dokumen (melalui printer)
14		Simbol <i>Start/End</i> , 13system ini mewakili titik awal, titik akhir, dan hasil potensial dari suatu jalur. Sering berisi “Mulai” atau “Akhir” dalam bentuk

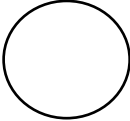
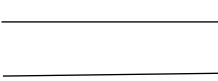
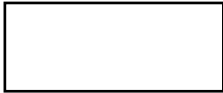

15		Simbol <i>Display</i> , berguna untuk menunjukkan di mana informasi akan ditampilkan dalam aliran proses.
----	-----------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------

2.17 *Diagram Flow Diagram (DFD)*

DFD adalah representasi dari notasi simbol yang menggambarkan aliran data sistem dan informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) hingga keluaran (*output*) (Salahudin, 2016).

Berdasarkan definisi diatas, maka dapat disimpulkan bahwa *DFD* adalah aliran informasi yang disusun dalam bentuk kumpulan komponen yang saling berhubungan simbol-simbol *Data Flow Diagram* dapat dilihat pada table 2.2.


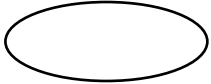

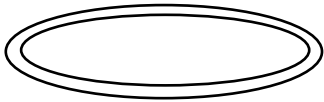
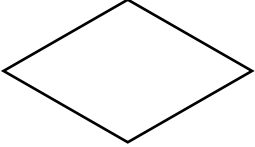

Tabel 2.2 Tabel *Diagram Flow Diagram (DFD)*

No.	Simbol	Nama	Keterangan
1.		<i>Proses</i>	Proses atau fungsi atau prosedur, pada permodelan perangkat lunak yang akan diimplementasikan.
2.		<i>Data Store</i>	File atau basis data atau penyimpanan
3.		<i>Entity</i>	Entitas luar atau masukan atau keluaran atau orang yang memakai.
4.		<i>Data Flow</i>	Aliran data merupakan data yang dikirim antar proses, dari penyimpanan ke proses

2.18 *Entity Relationship Diagram (ERD)*

(Salahuddin, 2016) *Entity Relationship Diagram (ERD)* adalah pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *ERD*. *ERD* dikembangkan berdasarkan teori himpunan dalam bidang matematika. *ERD* digunakan untuk pemodelan basis data *relasional*. Sehingga jika penyimpanan basis data menggunakan *OODBMS* maka perancangan basis data tidak perlu menggunakan *ERD*. Beberapa komponen yang terdapat didalam *Entity Relationship Diagram* model adalah dapat dilihat pada tabel 2.3.

Tabel 2.3 Tabel *Entity Relationship Diagram (ERD)*

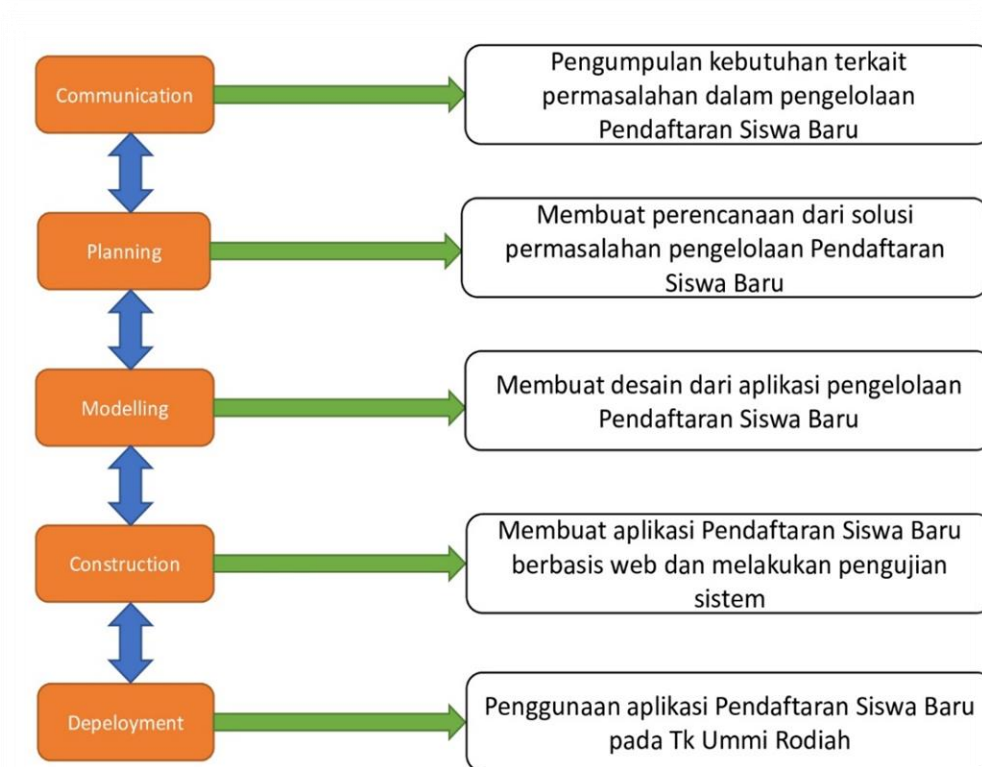
No.	Simbol	Keterangan
1.	Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut bersifat unik (berbeda tanpa ada yang sama).
4.	Atribut Multinilai / <i>multivalue</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6.	Asosiasi / <i>association</i> 	Penghubung antar relasi dan entitas dimana di kedua ujungnya memiliki multiplicity kemungkinan jumlah

		pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.
--	--	--------------------------------------------------------------------------------------------------------------------------------

2.19 Metode *Web Engineering*

Metode *Web Engineering* merupakan pendekatan rekayasa yang menggabungkan teknik pengembangan perangkat lunak dengan prinsip-prinsip desain web untuk menciptakan aplikasi yang dapat diskalakan, aman, dan mudah digunakan. Menggunakan pendekatan sistematis, disiplin, dan terukur untuk pengembangan, operasi dan pemeliharaan aplikasi berbasis *Web* (Simarmata, 2010:1). Sedangkan menurut (Wijaya,2020) mendefinisikan bahwa web engineering sebagai pendekatan metodologis yang mengkombinasikan praktik terbaik dari rekayasa perangkat lunak dan pengembangan *web* untuk menghasilkan aplikasi yang adaptif dan dapat diskalakan. Mereka menekankan pentingnya kolaborasi tim yang efektif dan penggunaan alat manajemen proyek untuk mencapai hasil yang optimal. Model yang digunakan untuk mencakup perencanaan, desain, implementasi, pengujian, dan pemeliharaan aplikasi *web* dengan pendekatan yang terstruktur dan berbasis data. Dengan menyoroti peran analitik dan pemahaman perilaku pengguna dalam memastikan aplikasi *web* yang dikembangkan dapat memberikan pengalaman pengguna yang optimal (Rahmawati, T.,dkk,2022).

Metode penelitian yang digunakan dalam penelitian ini adalah *web engineering*, alur proses *web engineering* yang digunakan mencakup tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Setiap tahap ini dilakukan secara sistematis dan terstruktur untuk memastikan aplikasi *web* yang dikembangkan memiliki kualitas tinggi, aman, dan memenuhi kebutuhan pengguna. Alur lengkap dari proses *web engineering* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Tahapan Proses *Web Engineering*

Web Engineering yang terdiri dari beberapa tahapan, yaitu :

1. *Communication*

Sebelum melakukan pengembangan perangkat lunak, seorang pengembang harus melakukan pengumpulan kebutuhan yang terkait dengan permasalahan yang terjadi pada proses pendaftaran siswa baru di TK Ummi Rodiah. Salah satu permasalahan utama yang diidentifikasi adalah keberadaan sistem pendaftaran yang masih menggunakan metode manual. Hal ini mengakibatkan proses pendaftaran menjadi kurang efisien dan rentan terhadap kesalahan input data serta kesulitan dalam manajemen informasi. Dengan memperkenalkan aplikasi yang tepat, diharapkan dapat meningkatkan efisiensi, akurasi dan keterjangkauan layanan pendaftaran bagi orang tua dan administrasi sekolah.

2. *Planning*

Dalam tahapan ini, dilakukan perencanaan mendetail untuk mengatasi permasalahan pengelolaan pendaftaran siswa baru di TK Ummi Rodiah dengan

menerapkan metode *web engineering*. Perencanaan meliputi identifikasi kebutuhan sistem yang komprehensif, seperti integrasi antarmuka pengguna yang intuitif, manajemen basis data yang efisien, dan pengaturan keamanan data yang ketat. Selain itu, tahapan ini juga mencakup perancangan arsitektur aplikasi web yang dapat menangani volume data pendaftaran siswa dengan skalabilitas yang baik serta memastikan ketersediaan dan keandalan sistem secara keseluruhan. Langkah ini direncanakan secara sistematis untuk memastikan bahwa solusi yang dihasilkan tidak hanya memenuhi kebutuhan operasional TK Ummi Rodiah saat ini, tetapi juga siap untuk pertumbuhan dan perkembangan masa depan.

3. *Modelling*

Tahapan ini mencakup pembuatan desain sistem untuk pengelolaan pendaftaran siswa baru di TK Ummi Rodiah dengan menggunakan diagram konteks, diagram *flow diagram* (DFD), *entity relationship diagram*(ERD), serta menggunakan *flowchart* yang akan menggambarkan interaksi antara berbagai aktor seperti orang tua siswa, staf administrasi dan mengilustrasikan proses utama dalam sistem, termasuk pendaftaran siswa baru oleh orang tua, verifikasi dokumen dan pembayaran oleh staf administrasi. Setiap usecase akan menjelaskan langkah-langkah yang diperlukan dalam pendaftaran, mulai dari registrasi, pengisian formulir oleh orang tua. Dengan diagram ini, alur kerja menjadi lebih jelas dan terstruktur, memudahkan setiap aktor untuk memahami peran dan tanggung jawab terhadap sistem pendaftaran siswa baru di TK Ummi Rodiah.

4. *Construction*

Proses pengembangan aplikasi penerimaan siswa baru berbasis web untuk TK Ummi Rodiah melibatkan beberapa tahapan konstruksi yang penting. Menggunakan bahasa pemrograman PHP dan *database MySQL*, aplikasi ini dirancang untuk mempermudah dan mempercepat proses penerimaan siswa baru. Dengan adanya aplikasi ini, diharapkan dapat membantu TK Ummi Rodiah dalam mengatasi berbagai permasalahan yang muncul selama proses penerimaan siswa baru, seperti pengelolaan data yang lebih terstruktur dan efisien. Selain itu, aplikasi ini akan diuji menggunakan metode *blackbox testing* untuk memastikan bahwa seluruh fitur

dan fungsionalitasnya berjalan sesuai dengan yang diharapkan, serta untuk mengidentifikasi dan memperbaiki kesalahan yang terjadi.

5. *Deployment*

Pada tahapan terakhir dalam metode web engineering, penerapan aplikasi penerimaan siswa baru pada TK Ummi Rodiah menjadi langkah yang krusial. Implementasi aplikasi ini bertujuan untuk meningkatkan efisiensi dan efektivitas kinerja bagian administrasi dan accounting TK Ummi Rodiah dalam mengelola proses penerimaan siswa baru. Dengan adanya aplikasi ini, seluruh data calon siswa dapat dikelola secara lebih terstruktur sehingga memudahkan dalam pencarian dan pelaporan data. Selain itu, aplikasi ini diharapkan dapat mempercepat proses administrasi, dan meningkatkan kepuasan orang tua dan wali murid dengan layanan yang lebih responsif.

2.20 **Blackbox Testing**

Black box Testing merupakan metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program. Dalam pengujian ini, menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya.

Pada *Black Box Testing* ini dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh customer. *Black box Testing* ini lebih menguji ke tampilan luar (*Interface*) dari suatu aplikasi agar mudah digunakan oleh pengguna. Pengujian ini tidak melihat dan menguji *source code program*. *Black-box testing* bekerja dengan mengabaikan *struktur control* sehingga perhatiannya hanya terfokus pada informasi domain.

Jadi metode ini bertujuan untuk memeriksa, setelah tahap akhir proyek, apakah perangkat lunak atau aplikasi berfungsi dengan baik, dan melayani penggunaannya secara *efisien*. Biasanya, penguji mencari fungsi yang hilang atau salah antarmuka, kinerja, inisialisasi program dan kesalahan keluar struktur data atau kesalahan akses basis data *eksternal* (Aisyah, 2021).