

**RANCANG BANGUN SISTEM INFORMASI PENDAFTARAN PESERTA  
DIDIK BARU (PPDB) BERBASIS WEB DI MADRASAH ALIYAH  
SWASTA (MAS) AL-MUHAJIRIN TUGUMULYO**

**(Skripsi)**

**Oleh:**

**RODHIYATI MARDHIYYAH  
NPM 2115061059**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2026**

**RANCANG BANGUN SISTEM INFORMASI PENDAFTARAN PESERTA  
DIDIK BARU (PPDB) BERBASIS WEB DI MADRASAH ALIYAH  
SWASTA (MAS) AL-MUHAJIRIN TUGUMULYO**

Oleh  
**RODHIYATI MARDHIYYAH**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar  
SARJANA TEKNIK**

**Pada**

**Jurusan Teknik Elektro  
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2026**

## ABSTRAK

### RANCANG BANGUN SISTEM INFORMASI PENDAFTARAN PESERTA DIDIK BARU (PPDB) BERBASIS WEB DI MADRASAH ALIYAH SWASTA (MAS) AL-MUHAJIRIN TUGUMULYO

Oleh

**RODHIYATI MARDHIYYAH**

Setiap tahun ajaran baru, sekolah dihadapkan pada proses Penerimaan Peserta Didik Baru (PPDB) yang membutuhkan pengelolaan data secara cepat dan efisien. Madrasah Aliyah Swasta (MAS) Al-Muhajirin Tugumulyo selama ini masih melakukan rekapitulasi data formulir dan berkas calon siswa pada proses PPDB secara manual yang menyebabkan kendala seperti kesalahan pencatatan data dan antrean pendaftaran. Untuk mengatasi masalah tersebut, penelitian ini melakukan pengembangan sistem informasi berbasis web menggunakan *framework* Laravel. Sistem dikembangkan dengan menerapkan metodologi *Personal Extreme Programming* (PXP) yang dilakukan dalam lima iterasi selama 14,5 minggu. Pengujian sistem dilakukan melalui tiga tahap yaitu *Black-Box Testing*, *System Usability Scale* (SUS), dan *load testing* menggunakan Apache JMeter. Hasil pengujian *black box* pada 20 *user stories* dengan 53 skenario dalam lima iterasi menunjukkan seluruh fungsionalitas sistem berjalan dengan baik. Pengujian SUS yang melibatkan 12 responden menghasilkan skor rata-rata 84,79 yang termasuk dalam kategori *acceptable*. Pengujian beban menggunakan Apache JMeter dengan mensimulasikan maksimal 100 *virtual user* menunjukkan sistem mampu memproses seluruh permintaan tanpa *error* dengan rata-rata waktu respons di bawah 3 detik. Sistem telah diimplementasikan pada tahap produksi dan dapat diakses secara publik. Dengan demikian sistem ini telah berhasil dikembangkan dan siap mendukung proses PPDB secara digital di MAS Al-Muhajirin Tugumulyo.

**Kata Kunci:** Sistem Informasi, Penerimaan Peserta Didik Baru, *Personal Extreme Programming*, *Black Box Testing*, *System Usability Scale*

## **ABSTRACT**

### **DESIGN AND DEVELOPMENT OF A WEB-BASED NEW STUDENT ADMISSION INFORMATION SYSTEM AT MADRASAH ALIYAH SWASTA (MAS) AL-MUHAJIRIN TUGUMULYO**

**By**

**RODHIYATI MARDHIYYAH**

Each academic year, schools face the challenge of managing the New Student Admission (PPDB) process efficiently and accurately. At MAS Al-Muhajirin Tugumulyo, the PPDB process was previously conducted manually, causing issues such as data entry errors and registration queues. To address these problems, this study developed a web-based information system using the Laravel framework. The system was developed using the Personal Extreme Programming (PXP) methodology through five iterations over 14.5 weeks. System testing was conducted in three stages: Black-Box Testing, System Usability Scale (SUS), and load testing using Apache JMeter. The black-box testing results on 20 user stories with 53 scenarios across five iterations showed that all system functionalities operated successfully. SUS testing involving 12 respondents produced an average score of 84.79, categorized as acceptable. Load testing using Apache JMeter with up to 100 virtual users demonstrated that the system processed all requests without errors, with an average response time below 3 seconds. The system has been deployed to the production stage and is publicly accessible. Therefore, the system was successfully developed and is ready to support the digital PPDB process at MAS Al-Muhajirin Tugumulyo.

**Keywords:** Information Systems, New Student Admission, Personal Extreme Programming, Black Box Testing, System Usability Scale

Judul Skripsi

: RANCANG BANGUN SISTEM  
INFORMASI PENDAFTARAN PESERTA  
DIDIK BARU (PPDB) BERBASIS WEB DI  
MADRASAH ALIYAH SWASTA (MAS)  
AL-MUHAJIRIN TUGUMULYO

Nama Mahasiswa

: *Rodhiyati Mardhiyyah*

Nomor Pokok Mahasiswa

: 2115061059

Program Studi

: Teknik Informatika

Fakultas

: Teknik

**MENYETUJUI**

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

**Ir. Resty Annisa S, S.T., M.T.**  
NIP. 199008302019032019

**Puput Budi Wintoro, S.Kom., M.T.I.**  
NIP. 198410312019031004

2. Mengetahui

Ketua Jurusan Teknik Elektro

Ketua Program Studi Teknik Informatika

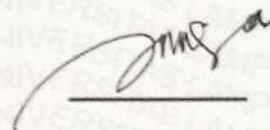
**Herlinawati, S.T., M.T.**  
NIP. 197103141999032001

**Yessi Mulyani, S.T., M.T.**  
NIP. 197312262000122001

**MENGESAHKAN**

1. Tim Penguji

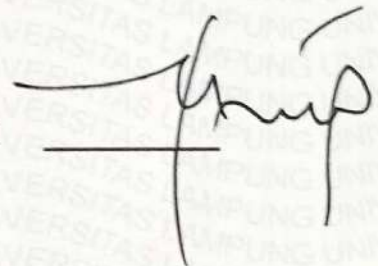
Ketua : **Ir. Resty Annisa S, S.T., M.T.**



Sekretaris : **Puput Budi Wintoro, S.Kom., M.T.I.**



Penguji : **Dr. Eng. Ir. Mardiana, S.T., M.T., I.P.M**



Dekan Fakultas Teknik

**Dr. H. Ahmad Herison, S.T., M.T.**

NIP. 196910302000031001

Tanggal lulus ujian skripsi: **7 Mei 2026**

## SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan dengan sebenarnya bahwa skripsi saya yang berjudul " Rancang Bangun Sistem Informasi Pendaftaran Peserta Didik Baru (PPDB) Berbasis Web Di Madrasah Aliyah Swasta (MAS) Al-Muhajirin Tugumulyo" merupakan hasil karya saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi saya ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 12 Mei 2026

Pembuat Pernyataan



Rodhiyati Mardhiyyah  
2115061059

## RIWAYAT HIDUP



Penulis dilahirkan di Mardiharjo pada tanggal 27 Januari 2003 sebagai anak ketiga dari enam bersaudara, dari pasangan Bapak Mukalil dan Ibu Nurliana. Penulis menyelesaikan jenjang pendidikan formal di SDN Mardiharjo pada tahun 2015, kemudian melanjutkan pendidikan ke SMP Aziziah Palembang dan lulus pada tahun 2018, serta menamatkan pendidikan menengah atas di SMA Aziziah Palembang pada tahun 2021. Pada tahun 2021, penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur SBMPTN. Selama masa perkuliahan, penulis aktif berpartisipasi dalam berbagai kegiatan, antara lain:

1. Menjadi anggota Divisi Kerohanian, Departemen Pendidikan dan Pengembangan Diri, Himpunan Mahasiswa Teknik Elektro (HIMATRO) Universitas Lampung Periode 2022 dan 2023
2. Mengikuti kegiatan MSIB Dicoding X Kampus Merdeka Batch 5 sebagai Pengembang *Front-end* dan *Back-end* pada tahun 2023
3. Mengikuti Program MSIB Batch 6 di Sekretariat Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi pada Proyek SISTER sebagai *Fullstack Developer* pada tahun 2024
4. Melaksanakan Kuliah Kerja Nyata di Desa Trisnomulyo, Kecamatan Batang Hari Nuban, Kabupaten Lampung Timur pada bulan Juni hingga Juli tahun 2024

## MOTTO

***“Start where you are, use what you have, do what you can”***

(Arthur Ashe)

***“And the only way to do great work is to love what you do. If you haven't found it yet, keep looking and don't settle. As with all matters of the heart, you'll know when you've found it.”***

(Steve Jobs)

***“You'll get over it”***

(Lil Uzi)

## **PERSEMBAHAN**

Puji syukur ke hadirat Allah SWT atas limpahan rahmat, karunia, dan hidayah-Nya, sehingga skripsi ini dapat terselesaikan dengan baik. Dengan penuh rasa syukur, saya persembahkan karya ini kepada:

### **Kedua Orang Tua**

Sebagai bentuk ungkapan terima kasih yang mendalam atas doa, dukungan, kasih sayang, serta pengorbanan tiada henti dalam setiap langkah perjalanan hidup saya.

Terima kasih atas segala nasihat, kesabaran, dan kepercayaan yang telah diberikan, sehingga saya dapat menyelesaikan pendidikan ini dengan baik.

Semoga segala kebaikan dan pengorbanan yang telah diberikan mendapatkan balasan yang berlipat ganda dari Allah SWT

## SANCAWANA

*Bismillah*, Segala puji dan syukur ke hadirat Allah SWT atas rahmat dan karunia-Nya yang luar biasa, sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul “Rancang Bangun Sistem Informasi Pendaftaran Peserta Didik Baru (PPDB) Berbasis Web Di Madrasah Aliyah Swasta (MAS) Al-Muhajirin Tugumulyo” yang menjadi salah satu syarat untuk memperoleh gelar Sarjana Teknik Universitas Lampung. Selama penyusunan skripsi ini penulis mendapat banyak bantuan, dukungan, bimbingan, serta doa dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada semua pihak yang telah berkontribusi dalam penyelesaian skripsi ini, baik secara langsung maupun tidak langsung, diantaranya:

1. Tuhan YME, yang telah memberikan nikmat, rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini hingga akhir.
2. Ibu Herlinawati, S.T., M.T., selaku ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung
3. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika Fakultas Teknik Universitas Lampung dan selaku dosen pembimbing utama bagi penulis.
4. Ibu Ir.Resty Annisa S, S.T., M.T. selaku Pembimbing Utama yang selalu meluangkan waktunya untuk membimbing, memberikan motivasi, dukungan, serta saran kepada penulis.
5. Bapak Puput Budi Wintoro, S.Kom., M.T.I., selaku Pembimbing Pendamping yang selalu memberikan dukungan serta bimbingan agar menjadi lebih baik.

6. Ibu Dr. Eng. Ir.Mardiana, S.T., M.T.,I.P.M, selaku Penguji skripsi yang telah memberikan kritik, saran, dan masukan kepada penulis dalam penelitian ini.
7. Staf Administrasi Jurusan yang telah membantu proses administrasi penulis selama masa awal perkuliahan penulis hingga akhir.
8. Ketua Tata Usaha Al-Muhajirin yang telah membantu penulis selama proses penelitian berlangsung.
9. Keluarga besar Jurusan Teknik Informatika angkatan 2021 yang telah menjadi bagian penting dalam perjalanan studi penulis melalui kebersamaan, kenangan yang berharga, serta dukungan dan motivasi yang diberikan selama masa perkuliahan.
10. Diri sendiri, yang telah mampu bertahan, berjuang, dan terus melangkah melewati berbagai tantangan selama proses perkuliahan hingga tahap akhir penyusunan penelitian ini. Terima kasih atas segala usaha serta keteguhan untuk tidak menyerah dalam keadaan apa pun. Terima kasih karena telah percaya bahwa setiap proses yang dijalani akan membawakan hasil pada waktunya.

Penulis menyadari masih terdapat kekurangan dalam penyusunan skripsi ini. Segala kritik dan saran sangat diharapkan untuk perbaikan ke depan. Semoga skripsi ini dapat bermanfaat bagi kita semua. Amin.

Bandar Lampung, 7 Mei 2026  
Penulis,

Rodhiyati Mardhiyyah  
NPM. 2115061059

## DAFTAR ISI

	Halaman
<b>ABSTRAK</b> .....	<b>xi</b>
<b>ABSTRACT</b> .....	<b>xii</b>
<b>DAFTAR ISI</b> .....	<b>i</b>
<b>DAFTAR TABEL</b> .....	<b>v</b>
<b>DAFTAR GAMBAR</b> .....	<b>vii</b>
<b>I. PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	4
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	5
1.5 Batasan Masalah.....	5
1.6 Penulisan Sistematika Skripsi/Tugas Akhir.....	6
<b>II. TINJAUAN PUSTAKA</b> .....	<b>7</b>
2.1 Penerimaan Peserta Didik Baru.....	7
2.1.1 Pengertian Penerimaan Peserta Didik Baru .....	7
2.1.2 Tujuan Penerimaan Peserta Didik Baru .....	7
2.1.3 Asas pelaksanaan Penerimaan Peserta Didik Baru .....	8
2.2 Sistem Informasi Manajemen.....	8
2.3 UML (Unified Modeling Language).....	9
2.3.1 Use Case Diagram.....	9
2.3.2 Activity Diagram .....	11

2.3.3	Class Diagram .....	12
2.4	PHP .....	12
2.5	Laravel .....	13
2.6	MySQL .....	14
2.7	Simple Mail Transfer Protocol (SMTP) .....	15
2.8	AdminLTE .....	16
2.9	PXP (Personal Extreme Programming) .....	16
2.10	MoSCoW Prioritazion .....	21
2.11	Low Fidelity Prototype .....	21
2.12	Black-Box Testing .....	22
2.13	System Usability Scale (SUS) .....	23
2.14	Apache JMeter .....	24
2.15	Penelitian Terkait .....	25
<b>III.</b>	<b>METODOLOGI PENELITIAN .....</b>	<b>30</b>
3.1	Waktu dan Tempat Penelitian .....	30
3.2	Alat dan Bahan Penelitian .....	33
3.3	Tahapan Penelitian .....	34
3.3.1	Requirements .....	34
3.3.2	Planning .....	35
3.3.3	Iterations Initialization .....	35
3.3.4	Design .....	36
3.3.5	Implementation .....	36
3.3.6	System Testing .....	36
3.3.7	Retrospective .....	37
3.3.8	Pengujian System Usability Scale (SUS) .....	37
3.3.9	Pengujian Beban (Load Testing) .....	38

3.3.10	Deployment.....	38
<b>IV.</b>	<b>PEMBAHASAN.....</b>	<b>39</b>
4.1	Requirements.....	39
4.1.1	Studi Literatur .....	39
4.1.2	Wawancara .....	39
4.2	Planning.....	44
4.3	Iterasi Pertama.....	47
4.3.1	Iteration initialization.....	47
4.3.2	Design .....	57
4.3.3	Implementation .....	64
4.3.4	System Testing .....	83
4.3.5	Retrospective.....	85
4.4	Iterasi Kedua .....	86
4.4.1	Iteration initalization.....	86
4.4.2	Design .....	90
4.4.3	Implementation .....	93
4.4.4	System Testing .....	103
4.4.5	Retrospective.....	105
4.5	Iterasi Ketiga .....	106
4.5.1	Iteration Initialization.....	106
4.5.2	Design .....	112
4.5.3	Implementation .....	115
4.5.4	System Testing .....	126
4.5.5	Retrospective.....	127
4.6	Iterasi Keempat .....	128
4.6.1	Iteration initalization.....	128

4.6.2	Design .....	132
4.6.3	Implementation .....	135
4.6.4	System Testing .....	149
4.6.5	Retrospective.....	150
4.7	Iterasi Kelima .....	151
4.7.1	Iteration initalization.....	151
4.7.2	Design .....	153
4.7.3	Implementation .....	154
4.7.4	System Testing .....	169
4.7.5	Retrospective.....	170
4.8	Pengujian Beban (Load Testing) .....	171
4.8.1	Hasil Pengujian Beban (Load Testing) .....	171
4.9	Pengujian System Usability Scale (SUS).....	175
4.9.1	Daftar Nama Responden .....	175
4.9.2	Rekapitulasi Jawaban Responden .....	176
4.10	Deployment .....	179
4.10.1	Identifikasi Keamanan Sistem .....	180
<b>V.</b>	<b>PENUTUP .....</b>	<b>183</b>
5.1	Kesimpulan.....	183
5.2	Saran.....	184
	<b>DAFTAR PUSTAKA .....</b>	<b>185</b>
	<b>LAMPIRAN.....</b>	<b>191</b>

## DAFTAR TABEL

	Halaman
Tabel 1. Notasi <i>Use Case</i> [16]. .....	10
Tabel 2. Daftar Pertanyaan Pada Pengujian System Usability Scale (SUS).....	23
Tabel 3. Penelitian Terkait.....	25
Tabel 4. Jadwal Penelitian.....	31
Tabel 5. Alat ( <i>Hardware</i> dan <i>Software</i> ) pengembangan sistem .....	33
Tabel 6. Bahan Pengembangan Sistem .....	33
Tabel 7. <i>User Stories</i> .....	41
Tabel 8. Kebutuhan Non-Fungsional .....	44
Tabel 9. Penentuan <i>Story Point</i> dan prioritas kebutuhan menggunakan metode <i>MoSCoW</i> .....	45
Tabel 10. Pembagian <i>User Stories</i> Kedalam Beberapa Iterasi.....	46
Tabel 11. Penentuan Estimasi Durasi Waktu Normal Pengerjaan Per-Iterasi Menggunakan Metode <i>Forecasting</i> .....	47
Tabel 12. Kebutuhan Sistem Iterasi Pertama .....	47
Tabel 13. Pendefinisian <i>Use Case</i> .....	49
Tabel 14. Hasil Pengujian <i>Black Box Testing</i> Iterasi Pertama .....	83
Tabel 15. Verifikasi Estimasi Waktu Pengerjaan Iterasi Pertama .....	86
Tabel 16. Kebutuhan Sistem Iterasi Kedua.....	87
Tabel 17. Hasil Pengujian <i>Black Box</i> Iterasi Kedua.....	104
Tabel 18. Verifikasi Estimasi Waktu Pengerjaan Iterasi Kedua .....	105
Tabel 19. Kebutuhan Sistem Iterasi Ketiga.....	106
Tabel 20. Pendefinisian <i>Use Case</i> .....	107
Tabel 21. Hasil Pengujian <i>Black Box</i> Iterasi Ketiga .....	126
Tabel 22. Verifikasi Estimasi Waktu Pengerjaan <i>User Stories</i> Ketiga.....	127
Tabel 23. Kebutuhan Sistem Iterasi Keempat.....	128

Tabel 24. Hasil Pengujian <i>Black Box</i> Iterasi Ketiga .....	149
Tabel 25. Verifikasi Estimasi Waktu Pengerjaan Iterasi Ketiga.....	150
Tabel 26. Kebutuhan Sistem Iterasi Kelima.....	151
Tabel 27. Hasil Pengujian <i>Black Box</i> Pada Iterasi Kelima .....	169
Tabel 28. Verifikasi Estimasi Waktu Pengerjaan Iterasi Kelima.....	170
Tabel 29. Spesifikasi Perangkat <i>Server Staging</i> Lokal .....	171
Tabel 30. Hasil Pengujian Beban 50 <i>Virtual User</i> .....	173
Tabel 31. Hasil Pengujian Beban 100 <i>Virtual User</i> .....	173
Tabel 32. Perbandingan Waktu Respons Sistem pada Proses Login dan Pengisian Formulir untuk 50 dan 100 User .....	174
Tabel 33. Daftar Nama Responden .....	175
Tabel 34. Rekapitulasi Skor Penilaian Responden.....	176
Tabel 35. Tabel Konversi Skor SUS .....	177
Tabel 36. Hasil Konversi Skor SUS.....	178
Tabel 37. Hasil Akhir Perhitungan Skor SUS.....	178
Tabel 38. Spesifikasi Hosting.....	179
Tabel 39. Identifikasi Mekanisme Keamanan Sistem.....	181

## DAFTAR GAMBAR

	Halaman
Gambar 1. Simbol Activity Diagram (sumber: <a href="https://www.dicoding.com">https://www.dicoding.com</a> ) .....	12
Gambar 2 . Arsitektur Laravel ( <i>Sumber: E-book Laravel Up and Running: A Framework for Building Modern Apps</i> ) [19]. .....	14
Gambar 3. Tahapan Metode PXP [6]. .....	19
Gambar 4. Alur Proses Black Box Testing [33]. .....	22
Gambar 5. Penilaian SUS dalam 3 Perspektif [35]. .....	24
Gambar 6. Tahapan Penelitian [6]. .....	34
Gambar 7. Prosedur Pendaftaran Peserta Didik Baru (PPDB) secara manual .....	39
Gambar 8. Prosedur Pendaftaran Siswa Baru (PPDB) Menggunakan Sistem Berbasis Website .....	41
Gambar 9. Diagram <i>Use Case</i> .....	48
Gambar 10. Activity Diagram Melakukan Registrasi .....	51
Gambar 11. <i>Activity Diagram</i> Melakukan Login.....	52
Gambar 12. <i>Activity Diagram</i> Membuat Jadwal PPDB.....	53
Gambar 13. <i>Activity Diagram</i> Mengedit Jadwal PPDB.....	54
Gambar 14. <i>Activity Diagram</i> Mengisi Formulir Pendaftaran.....	55
Gambar 15. <i>Activity Diagram</i> Memverifikasi pendaftaran.....	56
Gambar 16. <i>Activity Diagram</i> Mengecek Status Pendaftaran .....	57
Gambar 17. <i>Class Diagram</i> . .....	58
Gambar 18. <i>Wireframe</i> Form Daftar Akun. ....	59
Gambar 19. <i>Wireframe</i> Form Login.....	59
Gambar 20. <i>Wireframe</i> Manajemen Jadwal PPDB .....	60
Gambar 21. <i>Wireframe</i> Halaman Isi formulir pendaftaran online .....	61
Gambar 22. <i>Wireframe</i> Halaman Data Pendaftar.....	62
Gambar 23. <i>Wireframe</i> Halaman Verifikasi data pendaftaran .....	63

Gambar 24. <i>Wireframe</i> Halaman Cek Status Pendaftaran .....	64
Gambar 25. <i>Source Code Middleware</i> .....	65
Gambar 26. <i>Source Code Alias Middleware</i> .....	65
Gambar 27. Tampilan Halaman Registrasi Akun.....	66
Gambar 28. Tampilan Halaman Login Akun .....	66
Gambar 29. Tampilan Halaman Ubah <i>Password</i> .....	66
Gambar 30. <i>Model</i> Jadwal PPDB .....	67
Gambar 31. <i>Source Code Controller</i> Manajemen Jadwal PPDB.....	68
Gambar 32. Tampilan Form Membuat Jadwal PPDB Baru Di Halaman Manajemen Jadwal PPDB.....	69
Gambar 33. Tampilan Tab Riwayat Jadwal PPDB Di Halaman Manajemen Jadwal PPDB .....	70
Gambar 34. <i>Source Code Model</i> Pendaftaran .....	70
Gambar 35. <i>Source Code Model</i> Siswa.....	71
Gambar 36. <i>Source Code Model</i> Orang Tua .....	71
Gambar 37. <i>Source Code Controller</i> Pendaftaran Dengan Fungsi <i>Index</i> .....	72
Gambar 38. <i>Source Code Controller</i> Pendaftaran Dengan Fungsi <i>Store</i> .....	73
Gambar 39. Tampilan Halaman Formulir Siswa.....	74
Gambar 40. Tampilan Pesan <i>Error</i> validasi Formulir Siswa .....	75
Gambar 41. <i>Source Code Controller</i> Menetapkan Data <i>Default</i> Halaman Daftar Pendaftar .....	76
Gambar 42. Tampilan <i>Default</i> Halaman Data Pendaftar Pada Dashboard Admin	76
Gambar 43. <i>Source Code</i> Fungsi Menampilkan Detail Data Formulir Pendaftaran .....	77
Gambar 44. <i>Source Code</i> Mengubah Status Pendaftaran Siswa.....	78
Gambar 45. <i>Source Code</i> Status Perbaikan Pada Formulir Pendaftaran Siswa...	79
Gambar 46. Tampilan Halaman Detail Data Pendaftaran .....	80
Gambar 47. Tampilan Form Catatan Perbaikan .....	80
Gambar 48. <i>Method</i> Untuk Mengatur Tampilan Status Pendaftaran .....	81
Gambar 49. <i>Source Code</i> Menampilkan Status Pendaftaran Di Halaman Dashboard Siswa.....	82
Gambar 50. Tampilan Halaman Status Pendaftaran Siswa.....	82

Gambar 51. <i>Activity Diagram</i> Mengedit formulir pendaftaran .....	88
Gambar 52. <i>Activity Diagram</i> Mencetak Bukti Pendaftaran .....	89
Gambar 53. <i>Activity Diagram</i> Mencetak Data Pendaftaran.....	89
Gambar 54. <i>Wireframe</i> Edit formulir pendaftaran .....	91
Gambar 55. <i>Wireframe</i> Cetak Bukti Pendaftaran.....	92
Gambar 56. <i>Wireframe</i> Kelola data jurusan.....	92
Gambar 57. <i>Wireframe</i> Kelola Kategori Prestasi.....	93
Gambar 58. <i>Source Code</i> Edit Formulir Pendaftaran .....	94
Gambar 59. Tampilan Halaman Edit Formulir Pendaftaran .....	95
Gambar 60. <i>Source Code</i> Pratinjau Cetak Bukti Pendaftaran dan Cetak Bukti Pendaftaran .....	96
Gambar 61. Tampilan Halaman Pratinjau Cetak Bukti Pendaftaran.....	97
Gambar 62. Tampilan Dokumen <i>PDF</i> Hasil Cetak Bukti Pendaftaran .....	97
Gambar 63. <i>Source Code</i> Memfilter Data Pendaftar Pada Halaman Dashboard Admin .....	98
Gambar 64. Tampilan Hasil Filter Data Pada halaman Data Seluruh Pendaftar Di Dashboard Admin .....	99
Gambar 65. <i>Source Code</i> Menghapus Data Pendaftar Yang Tidak Valid .....	100
Gambar 66. Tampilan <i>Modal</i> Berhasil Hapus Data Pendaftar Tidak Valid .....	100
Gambar 67. <i>Source Code</i> Mencetak Data Pendaftar Kedalam Format <i>CSV</i> .....	101
Gambar 68. <i>Source Code</i> Mencetak Data Pendaftar Kedalam Format <i>PDF</i> .....	102
Gambar 69. Tampilan Hasil Unduh Data Pendaftar Dalam Bentuk <i>CSV</i> .....	103
Gambar 70. Tampilan Hasil Unduh Data Pendaftar Dalam Bentuk <i>PDF</i> .....	103
Gambar 71. <i>Diagram Use Case</i> .....	107
Gambar 72. <i>Activity Diagram</i> Menambah Jurusan.....	110
Gambar 73. <i>Activity Diagram</i> Menambah Kategori Prestasi.....	111
Gambar 74. <i>Activity Diagram</i> Mengedit Kategori Prestasi .....	112
Gambar 75. <i>Class Diagram</i> . .....	113
Gambar 76. <i>Wireframe</i> Kelola data jurusan.....	114
Gambar 77. <i>Wireframe</i> Kelola Kategori Prestasi .....	115
Gambar 78. <i>Source Code Model</i> Jurusan.....	116
Gambar 79. <i>Source Code</i> Menampilkan Data Jurusan .....	116

Gambar 80. <i>Source Code</i> Menyimpan Data Jurusan Ke <i>Database</i> .....	117
Gambar 81. <i>Source Code</i> Memperbarui Data Jurusan.....	117
Gambar 82. <i>Source Code</i> Menghapus Data Jurusan.....	118
Gambar 83. Tampilan Halaman Manajemen Jurusan .....	119
Gambar 84. Tampilan Form Menambah Jurusan Baru .....	119
Gambar 85. Tampilan Form Edit Data Jurusan.....	120
Gambar 86. <i>Source Code Model</i> Kategori Prestasi.....	121
Gambar 87. <i>Source Code</i> Menampilkan Data Kategori Prestasi .....	121
Gambar 88. <i>Source Code</i> Menyimpan Data Kategori Prestasi Baru .....	122
Gambar 89. <i>Source Code</i> Memperbarui Data Kategori Prestasi.....	123
Gambar 90. <i>Source Code</i> Menghapus Data Kategori Prestasi.....	124
Gambar 91. Tampilan Halaman Manajemen Jalur Prestasi .....	124
Gambar 92. Tampilan Form Menambah Kategori Prestasi Baru.....	125
Gambar 93. Tampilan Form Memperbarui Data Kategori Prestasi .....	125
Gambar 94. <i>Activity Diagram</i> Menghapus Jadwal Pendaftaran .....	129
Gambar 95. <i>Activity Diagram</i> Diagram Tambah User.....	130
Gambar 96. <i>Activity Diagram</i> Mengedit User .....	131
Gambar 97. <i>Activity Diagram</i> Hapus User .....	132
Gambar 98. <i>Wireframe</i> Halaman Manajemen User .....	133
Gambar 99. <i>Wireframe</i> Riwayat Jadwal.....	133
Gambar 100. <i>Wireframe Landing Page</i> . .....	134
Gambar 101. <i>Wireframe</i> Riwayat Jadwal.....	135
Gambar 102. <i>Source Code</i> Mengelola Status Aktif Jadwal PPDB .....	136
Gambar 103. <i>Source Code</i> Penentuan <i>Default</i> Status Jadwal Pendaftaran.....	136
Gambar 104. <i>Source Code</i> Pengaturan Pembukaan dan Penutupan Formulir Pendaftaran .....	137
Gambar 105. Tampilan Tab Riwayat Jadwal PPDB Pada Halaman Manajemen Jadwal PPDB .....	138
Gambar 106. Tampilan Penutupan Halaman Formulir Pendaftaran .....	138
Gambar 107. <i>Source Code</i> Menghapus Jadwal Tidak Valid .....	139
Gambar 108. Tampilan <i>Modal</i> Berhasil Hapus Jadwal.....	140
Gambar 109. <i>Source Code</i> Menampilkan Halaman Manajemen User .....	140

Gambar 110. <i>Source Code</i> Membuat dan Menyimpan Data User Baru .....	141
Gambar 111. <i>Source Code</i> Memperbarui Data User.....	142
Gambar 112. <i>Source Code</i> Menghapus Data User.....	143
Gambar 113. Tampilan Halaman Manajemen User .....	144
Gambar 114. Tampilan Form Tambah User Baru .....	145
Gambar 115. Tampilan Form Memperbarui Data User .....	145
Gambar 116. <i>Source Code</i> Menampilkan Informasi Jadwal Aktif, Kategori Prestasi, dan Jurusan Pada Halaman <i>Landing Page</i> .....	146
Gambar 117. <i>Source Code</i> Menampilkan Syarat Pendaftaran Pada Halaman <i>Landing Page</i> .....	147
Gambar 118. Tampilan Halaman <i>Landing Page</i> .....	148
Gambar 119. <i>Activity Diagram</i> Melihat Data Statistik Calon Siswa.....	151
Gambar 120. <i>Activity Diagram</i> Mencetak Grafik Statistik.....	152
Gambar 121. <i>Activity Diagram</i> Memfilter Data Pendaftaran .....	153
Gambar 122. <i>Wireframe</i> Halaman Dashboard .....	154
Gambar 123. <i>Source Code</i> Konfigurasi <i>SMTP Gmail Server</i> .....	155
Gambar 124. <i>Source Code Trait</i> Token Login .....	155
Gambar 125. <i>Source Code Laravel Mailable</i> .....	156
Gambar 126. <i>Source Code</i> Pengiriman Email Notifikasi Pendaftaran Menggunakan <i>Queue</i> .....	156
Gambar 127. <i>Source Code Artisan Command</i> Mengirim Email Pengumuman Pendaftaran .....	157
Gambar 128. Tampilan Email Notifikasi Status Pendaftaran .....	158
Gambar 129. <i>Source Code</i> Menampilkan Daftar Tahun Ajaran Di halaman Dashboard Admin .....	158
Gambar 130. <i>Source Code</i> Mengambil Data Statistik Total Pendaftar .....	159
Gambar 131. <i>Source Code</i> Mengambil Data Jumlah Siswa Yang Mendaftar Melalui Jalur Prestasi .....	160
Gambar 132. <i>Source Code</i> Mengambil Data Jumlah Siswa Pendaftar Berdasarkan Jurusan .....	161
Gambar 133. <i>Source Code</i> Mengambil Data Jumlah Pendaftar Berdasarkan Tahun Ajaran Dan Gelombang Pendaftaran .....	162

Gambar 134. <i>Source Code</i> Menampilkan Visualisasi Data Dalam Grafik Pie..	163
Gambar 135. <i>Source Code</i> Menampilkan Visualisasi Data Dalam Grafik Batang .....	164
Gambar 136. Tampilan Halaman Utama Dashboard Admin .....	165
Gambar 137. <i>Source Code Dropdown</i> Filter Tahun.....	166
Gambar 138. <i>Source Code Event Listener</i> Filter Tahun.....	167
Gambar 139. <i>Source Code</i> Pengiriman Parameter Filter Tahun ke <i>Backend</i> dan Pembaruan Grafik .....	168
Gambar 140. <i>Source Code</i> Filter Tahun Ajaran .....	168
Gambar 141. Tampilan <i>Dropdown</i> Filter Tahun .....	169
Gambar 142. Grafik Perbandingan Waktu Respons Sistem pada Proses Login dan Pengisian Formulir untuk 50 dan 100 User .....	175

## I. PENDAHULUAN

### 1.1 Latar Belakang

Pendidikan merupakan aspek penting dalam pembangunan suatu bangsa karena berperan dalam membentuk karakter, meningkatkan kualitas sumber daya manusia, serta mendorong daya saing bangsa [1]. Salah satu jalur pendidikan yang umumnya ditempuh adalah pendidikan formal. Menurut Undang-Undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional, pendidikan formal merupakan jalur pendidikan yang terstruktur dan berjenjang yang terdiri atas pendidikan dasar, pendidikan menengah, dan pendidikan tinggi. Untuk menjamin pemerataan pendidikan bagi setiap warga, pemerintah pusat maupun daerah wajib menyediakan fasilitas, layanan, kemudahan serta menjamin terselenggaranya pendidikan yang berkualitas bagi setiap warga tanpa diskriminasi.

Sumber utama dan terpenting dalam pendidikan formal adalah peserta didik. Tanpa adanya peserta didik, maka proses pendidikan formal tidak bisa berjalan dengan semestinya [2]. Oleh karena itu, setiap institusi pendidikan perlu menyelenggarakan proses penerimaan peserta didik baru setiap tahun sebagai bagian dari operasional pendidikan. Penerimaan peserta didik baru atau yang biasa disingkat menjadi PPDB merupakan proses seleksi yang dilakukan oleh sekolah untuk menerima calon peserta didik baru dengan menyeleksi peserta yang memenuhi syarat tertentu untuk memperoleh jenjang satuan pendidikan yang lebih tinggi. Proses ini bertujuan untuk mendapatkan calon siswa yang memenuhi kriteria sekolah, mengelola jumlah siswa sesuai daya tampung dan memberikan kesempatan yang sama bagi siswa untuk mendapatkan pendidikan yang layak [3].

Seiring dengan perkembangan teknologi informasi, proses administrasi Penerimaan Peserta Didik Baru (PPDB) mulai beralih menuju sistem digital.

Perubahan ini terjadi karena administrasi manual atau konvensional sering memunculkan sejumlah kendala seperti risiko kesalahan pencatatan data, proses rekapitulasi data yang memakan waktu, antrean pendaftaran serta ketergantungan pada dokumen fisik yang mudah hilang atau rusak.

Berdasarkan penelitian sebelumnya yang dilakukan oleh Warta dan Janaenah (2024) dengan judul “Analisis Perbandingan Efektivitas Sistem PPDB Konvensional dan Online : Kajian terhadap Aspek Waktu, Biaya, dan Akurasi Data” menjelaskan bahwasanya layanan PPDB yang menggunakan sistem *online* dinilai lebih unggul dari pada yang menggunakan metode konvensional, karena dapat menghemat waktu pendaftaran 15-20 menit per pendaftar dibandingkan dengan pendaftaran secara manual yang bisa memakan waktu 45-60 menit per pendaftar. Selain itu dari tingkat akurasi data, PPDB online lebih unggul karena persentase jumlah kesalahan lebih kecil yaitu sebesar 3-5% dibanding PPDB manual dengan tingkat kesalahan sebanyak 8-12% [4].

Meskipun digitalisasi layanan PPDB memiliki berbagai keunggulan, namun masih terdapat sekolah yang belum menerapkan sistem PPDB online, termasuk sekolah dengan fasilitas dan sumber daya yang sebenarnya sudah memadai. Salah satunya adalah Madrasah Aliyah Swasta (MAS) Al-Muhajirin, yaitu sebuah sekolah swasta yang beralamat di Jl. Jend. Sudirman, Desa Trikoyo, Kecamatan Tugu Mulyo, Kabupaten Musi Rawas, Provinsi Sumatera Selatan. Sekolah ini didirikan tahun 1983 dan telah meraih akreditasi A berdasarkan sertifikat 036/BAN-PDM/SK/2025. MAS Al-Muhajirin dikenal dengan sekolah yang mampu menghasilkan siswa-siswi berprestasi. Setiap tahun siswa-siswi di sekolah ini berhasil meraih berbagai kejuaraan dalam perlombaan akademik maupun non akademik di tingkat lokal dan regional sehingga menjadikan sekolah ini sebagai salah satu Madrasah Aliyah unggulan di Kabupaten Musi Rawas.

Selama tiga tahun terakhir, MAS Al-Muhajirin telah menerima rata-rata lebih dari 200 calon siswa baru per tahun. Selama pelaksanaan PPDB, panitia masih melakukan rekapitulasi data formulir dan berkas calon siswa secara manual, sehingga prosesnya memerlukan waktu yang cukup lama. Selain itu, metode

manual tersebut meningkatkan risiko kesalahan pencatatan data yang pada akhirnya memerlukan pemeriksaan dan perbaikan ulang. Akibatnya, proses penerimaan siswa baru menjadi lebih lambat dan kurang efisien. Kemudian saat melakukan pendaftaran, calon siswa atau wali calon siswa harus datang langsung ke sekolah untuk mengisi formulir dan menyerahkan berkas fisik. Ketergantungan pada dokumen fisik tersebut berpotensi menimbulkan risiko kehilangan atau kerusakan berkas. Selain itu, proses ini juga dapat menyebabkan antrean pendaftaran apabila terdapat banyak calon siswa atau wali calon siswa yang datang untuk melakukan pendaftaran di waktu yang bersamaan.

Berdasarkan uraian permasalahan diatas, penelitian ini bertujuan untuk merancang dan membangun sistem informasi Pendaftaran Peserta Didik Baru Berbasis Website di MAS Al-Muhajirin. Sistem ini dibangun menggunakan *framework* Laravel. *Framework Laravel* dipilih karena merupakan salah satu *framework PHP* yang bersifat *open source*. *Laravel* memiliki arsitektur MVC (*Model-View-Controller*) yang dapat memisahkan antara logika, *model* dan *view* [5]. Adapun metodologi penelitian yang akan digunakan untuk mengembangkan sistem ini adalah *Personal Extreme Programming* (PXP) yaitu metode pengembangan perangkat lunak yang diadaptasi dari metode *agile* (*Extreme Programming*) untuk pengembang yang bekerja secara individu. Metode ini dipilih karena pengembangannya bersifat iteratif sehingga memungkinkan evaluasi dan perbaikan secara bertahap di setiap iterasi, serta mengurangi proses dokumentasi yang berlebihan yang dapat memperlambat proses kerja [6].

Hasil perancangan sistem ini akan diuji menggunakan pengujian *black box testing* dan *System Usability Scale* (SUS). *Black box testing* adalah teknik pengujian yang berfokus pada fungsionalitas aplikasi tanpa melihat implementasi kode didalamnya. Alasan dipilihnya metode *black box testing* karena pengujian ini dilakukan berdasarkan sudut pandang klien. Penguji tidak perlu memahami kode program yang ada dibalik sistem sehingga proses pengujian menggunakan *black box testing* bisa lebih cepat namun tetap mendapatkan hasil yang akurat [7]. Sedangkan metode pengujian *System Usability Scale* (SUS) merupakan metode yang digunakan untuk menguji kebergunaan suatu sistem menggunakan skala

*likert* dengan 5 poin yang terdiri dari 10 pertanyaan dimana pertanyaan negatif dan pertanyaan positif masing-masing disusun secara bergantian dengan tujuan supaya responden membaca keseluruhan pertanyaan. Pengujian ini terbukti efisien untuk menilai *usability* atau kegunaan suatu sistem dengan menggunakan ukuran sample yang kecil. Hasil perhitungan skor *SUS* dapat memberikan wawasan dalam mengukur persepsi pengguna terhadap kemudahan penggunaan sistem [8].

Selain itu pada sistem ini juga dilakukan pengujian beban (*load testing*) yaitu sebuah tahap pengujian sistem yang bertujuan untuk mengetahui kinerja dan waktu respons sistem ketika diakses oleh banyak pengguna secara simultan. Pengujian ini menggunakan Apache JMeter untuk mensimulasikan pengguna menggunakan *virtual user* dengan jumlah yang bervariasi. Hasil pengujian ini akan menjadi bahan evaluasi stabilitas, kecepatan, dan ketahanan sistem PPDB terhadap peningkatan jumlah pengguna. [9].

## **1.2 Perumusan Masalah**

1. Bagaimana merancang dan mengembangkan sistem informasi PPDB berbasis website untuk mendukung dan mempermudah pengelolaan proses pendaftaran siswa baru di MAS Al-Muhajirin Tugumulyo?
2. Apakah sistem PPDB berbasis website yang dikembangkan telah berfungsi dengan baik berdasarkan hasil pengujian menggunakan metode *Black Box Testing*?
3. Apakah sistem PPDB berbasis website yang dikembangkan telah memenuhi aspek kebergunaan (*usability*) dari perspektif pengguna?
4. Apakah sistem PPDB berbasis website yang dikembangkan tetap memiliki kinerja yang baik saat diakses oleh banyak pengguna secara bersamaan?

## **1.3 Tujuan Penelitian**

Adapun tujuan dari penelitian ini di antaranya adalah:

1. Merancang dan mengembangkan sistem informasi PPDB berbasis website untuk mendukung dan mempermudah pengelolaan proses pendaftaran siswa baru di MAS Al-Muhajirin Tugumulyo

2. Menguji tingkat fungsionalitas sistem melalui metode pengujian *black box*.
3. Menguji tingkat kebergunaan (*usability*) sistem PPDB berbasis website berdasarkan hasil evaluasi menggunakan metode *System Usability Scale (SUS)*.
4. Menguji ketahanan sistem PPDB terhadap peningkatan jumlah pengguna menggunakan metode pengujian beban (*load testing*)

#### **1.4 Manfaat Penelitian**

Penelitian ini diharapkan memberi manfaat sebagai berikut:

- a. Bagi Panitia PPDB  
Meningkatkan efisiensi dalam pengelolaan pendaftaran dengan mengurangi pekerjaan manual dan meminimalkan risiko kesalahan data.
- b. Bagi Calon Peserta Didik  
Mempermudah proses pendaftaran karena dapat dilakukan secara lebih cepat tanpa harus datang ke sekolah dan menyerahkan berkas fisik secara langsung.
- c. Bagi Sekolah  
Mendukung digitalisasi administrasi sekolah sehingga proses PPDB menjadi lebih akurat dan terdokumentasi dengan baik.

#### **1.5 Batasan Masalah**

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini tidak membahas aspek keamanan tingkat lanjut, seperti enkripsi data maupun autentikasi berlapis.
2. Sistem yang dikembangkan berlaku untuk satu institusi yaitu MAS Al-Muhajirin dan belum terhubung dengan sistem eksternal atau sistem pendidikan nasional lainnya.
3. Perancangan sistem difokuskan untuk membuat sistem penerimaan peserta didik baru tanpa membuat sistem ujian seleksi berbasis online.
4. Pengujian sistem dibatasi pada *Black Box Testing*, *System Usability Testing (SUS)* dan pengujian beban (*Load Testing*), dan tidak mencakup pengujian performa yang lebih mendalam seperti *Stress Testing* (uji ketahanan beban ekstrem).

## 1.6 Penulisan Sistematika Skripsi/Tugas Akhir

Secara garis besar penelitian ini terbagi dalam beberapa bab yang tersusun sebagai berikut:

### **BAB I : PENDAHULUAN**

Bab ini berisi latar belakang yang menjelaskan alasan pengembangan Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) Berbasis Web Di Madrasah Aliyah Swasta (MAS) Al-Muhajirin Tugumulyo, perumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan laporan.

### **BAB II : TINJAUAN PUSTAKA DAN LANDASAN TEORI**

Membahas tentang penelitian-penelitian terkait pada tinjauan pustaka dan dasar-dasar teori dari Penerimaan Peserta Didik Baru, Sistem Informasi Manajemen, *UML (Unified Modeling Language, PHP, Laravel, MySQL, SMTP, AdminLTE, Apache Jmeter, Low Fidelity Design, Personal Extreme Programming (PXP)* dan metode pengujian sistem yang terdiri dari *System Usability Scale (SUS)* dan *Black-Box Testing*

### **BAB III : METODE PENELITIAN**

Bab ini berisi pembahasan tentang waktu, tempat, alat dan bahan, serta tahapan penelitian yang akan dikerjakan.

### **BAB IV : HASIL DAN PEMBAHASAN**

Berisi tentang proses pengambilan data, hasil yang didapatkan saat penelitian dan analisis data dari hasil penelitian.

### **BAB V : KESIMPULAN DAN SARAN**

Berisi tentang kesimpulan dari penelitian yang dilakukan dan saran yang didasarkan pada hasil data mengenai perbaikan dan pengembangan lebih lanjut agar didapatkan hasil lebih baik.

**DAFTAR PUSTAKA** : Bab ini memuat daftar sumber kutipan teori-teori yang dijadikan acuan dalam menulis laporan.

## **II. TINJAUAN PUSTAKA**

### **2.1 Penerimaan Peserta Didik Baru**

#### **2.1.1 Pengertian Penerimaan Peserta Didik Baru**

Penerimaan peserta didik baru merupakan proses seleksi yang dilakukan oleh sekolah untuk menerima calon peserta didik baru dengan menyeleksi peserta yang memenuhi syarat tertentu untuk memperoleh jenjang satuan pendidikan yang lebih tinggi [10]. Proses rekrutmen peserta didik baru merupakan bagian dari manajemen peserta didik, yaitu segala sesuatu yang berkaitan dengan perencanaan, pengorganisasian, pelaksana, dan pengawasan terhadap kegiatan peserta didik mulai dari penerimaan siswa hingga siswa menyelesaikan pendidikannya di sekolah.. Proses rekrutmen peserta didik baru memiliki beberapa tahapan. Yang pertama melakukan pembentukan panitia penerimaan peserta didik baru yang akan bertugas menyelenggarakan seluruh kegiatan seleksi. Panitia ini bisa terdiri dari berbagai pihak di sekolah seperti staff administrasi, komite sekolah dan guru. Kedua memasang pengumuman penerimaan siswa baru secara terbuka. Kegiatan ini bisa dilakukan dengan berbagai media seperti menggunakan media sosial atau pamflet. Adapun informasi yang dicantumkan didalamnya wajib memuat profil singkat sekolah, prosedur pendaftaran, syarat pendaftaran, jadwal pendaftaran, jumlah ketersediaan daya tampung, ketentuan pendaftaran tidak dipungut biaya. serta penetapan pengumuman hasil proses seleksi [11].

#### **2.1.2 Tujuan Penerimaan Peserta Didik Baru**

Tujuan pelaksanaan Penerimaan peserta didik baru yang pertama adalah untuk memberikan kesempatan yang setara bagi seluruh peserta didik untuk memperoleh layanan pendidikan bermutu yang dekat dengan domisili. Kedua

memperluas akses pendidikan bagi murid yang berasal dari keluarga dengan ekonomi tidak mampu dan penyandang disabilitas. Ketiga meningkatkan motivasi dan capaian prestasi peserta didik. Dan yang keempat adalah memaksimalkan peran masyarakat untuk ikut terlibat dalam proses penerimaan murid [12].

### **2.1.3 Asas pelaksanaan Penerimaan Peserta Didik Baru**

Didalam Permendikdasmen no 3 tahun 2025 bahwasanya asas penerimaan peserta didik baru dilaksanakan berdasarkan asas berikut: (a) objektif, yaitu pelaksanaan PPDB harus dilakukan berdasarkan ketentuan perundang-undangan yang berlaku tanpa adanya pengaruh dari kepentingan pribadi atau golongan tertentu; (b) transparan, yaitu informasi terkait PPDB dapat diakses oleh masyarakat luas secara terbuka; (c) akuntabel, yaitu terkait dengan proses pelaksanaan PPDB mulai dari tahap awal hingga selesai dapat dipertanggung jawabkan; (d) berkeadilan, yaitu setiap warga negara memiliki hak yang sama untuk mendapatkan kesempatan menempuh pendidikan; dan (e) tanpa diskriminasi, yaitu pelaksanaan PPDB dilakukan tanpa membedakan latar belakang suku, ras, agama atau status sosial kepada calon siswa baru.

## **2.2 Sistem Informasi Manajemen**

Sistem Informasi Manajemen (SIM) adalah sistem yang menggabungkan teknologi, manusia, dan proses untuk mengumpulkan, mengelola, menyimpan, dan mendistribusikan informasi guna mendukung pengambilan keputusan dalam sebuah organisasi. SIM tidak hanya mencakup perangkat keras dan perangkat lunak, tetap juga bagaimana teknologi digunakan oleh manajer dan karyawan untuk meningkatkan efisiensi operasional dan mencapai tujuan bisnis. Sistem informasi terdiri dari tiga komponen utama yaitu:

1. Input, yaitu proses mengumpulkan data mentah dari lingkungan internal maupun eksternal
2. Proses, yaitu mengubah data mentah menjadi informasi yang bermakna.
3. Output, yaitu penyampaian informasi kepada pihak yang membutuhkannya.

Selain itu, Sistem informasi juga memerlukan umpan balik untuk memastikan bahwa informasi dapat membantu suatu organisasi untuk memastikan bahwa informasi yang dihasilkan tetap akurat dan relevan. Dengan penerapan yang baik, sistem informasi dapat membantu meningkatkan efisiensi, mengoptimalkan berbagai proses, serta mendukung pengambilan keputusan yang lebih efektif dan strategis di berbagai bidang [13].

### **2.3 UML (*Unified Modeling Language*)**

UML (*Unified Modeling Language*) merupakan sebuah pemodelan standar untuk software dan pengembangan sistem berbasis OO (*Object-Oriented*). UML digunakan untuk mendeskripsikan, memvisualisasikan, membangun, dan mendokumentasikan bagian-bagian dari suatu sistem yang kompleks. UML dikembangkan sekitar pertengahan tahun 1990 oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Versi pertama UML diciptakan tahun 1997 (UML 1.0). Saat ini UML yang digunakan sudah tersedia dalam versi 2.0 yang menjadi standar internasional (ISO) [14]. Menurut draw.io (2021) sudah ada 14 diagram yang ditambahkan dalam UML 2.5. Beberapa diantaranya adalah *use case*, *class*, dan *activity diagram* [15].

#### **2.3.1 Use Case Diagram**

*Use case* merupakan sebuah diagram yang menggambarkan bagaimana pengguna atau user berinteraksi dengan sistem. *Use case* dapat memperlihatkan proses aktivitas secara urut dalam sistem, mengidentifikasi pihak-pihak yang dapat berinteraksi dengan sistem serta tindakan yang dapat dilakukan oleh sistem. Adapun komponen-komponen di dalam diagram *use case* terdiri dari *use case*, aktor, asosiasi, dan relasi antar *use case*. *Use case* merupakan komponen berbentuk oval horizontal yang menggambarkan fungsi atau layanan pada sistem. Aktor merupakan entitas eksternal yang berinteraksi dengan sistem. Aktor tidak hanya berupa pengguna manusia tetapi bisa juga berupa perangkat keras atau sistem lain seperti *E-mail server* [16].

Berdasarkan penjelasan Cockburn, *use case* dibagi ke dalam beberapa level tujuan dimana setiap level menunjukkan tingkat kedalaman detail dari tujuan yang dituliskan. Dengan adanya pembagian level ini dapat membantu pengembang untuk menentukan seberapa luas atau rinci sebuah *use case* sebaiknya dibuat. Level-level *use case* tersebut dapat diklasifikasikan sebagai berikut:

1. *Summary level*

Level ini digunakan untuk menjelaskan konteks besar atau kumpulan dari banyak tujuan pengguna yang saling terkait dalam satu siklus bisnis. Level ini bersifat umum dan digunakan untuk memberikan konteks keseluruhan sistem.

2. *User-goal level*


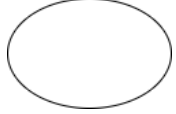

Level ini digunakan untuk menjelaskan tugas yang ingin diselesaikan oleh satu aktor dalam satu sesi. Level ini dianggap yang paling direkomendasikan karena berfokus pada nilai bisnis dan kebutuhan pengguna dengan jumlah langkah yang ringkas (sekitar 3–10 langkah).

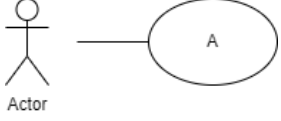
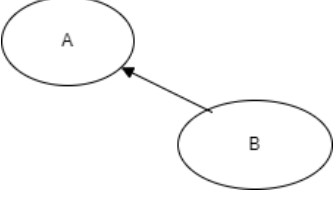
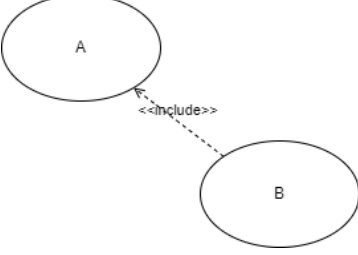
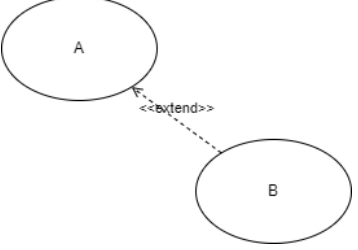
3. *Subfunction level*

Level ini digunakan untuk menjelaskan langkah-langkah yang sangat rinci dan teknis. Level ini tidak disarankan jika terlalu detail, karena dapat membuat *use case* terlalu panjang dan sulit dipahami [17].

Berikut ini merupakan notasi diagram use case yang sering digunakan:



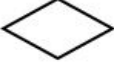



Tabel 1. Notasi *Use Case* [16].

Nama	Notasi	Deskripsi
Sistem		Menunjukkan batas antara sistem dan pengguna
Use case		Unit fungsionalitas yang disediakan oleh sistem
Aktor		Peran pengguna atau entitas eksternal yang berinteraksi dengan sistem

Nama	Notasi	Deskripsi
Association		Menunjukkan bahwa aktor berpartisipasi dalam eksekusi <i>use case</i>
Generalization		Aktor anak mewarisi peran dan partisipasi aktor induk dalam <i>use case</i>
Include		<i>Use case</i> utama selalu menyertakan <i>use case</i> lain sebagai bagian wajib dalam alur proses.
Extends		<i>Use case</i> tambahan memperluas <i>use case</i> utama secara opsional, hanya dijalankan pada kondisi tertentu.

### 2.3.2 Activity Diagram

*Activity diagram* merupakan diagram yang menggambarkan proses prosedural dalam suatu sistem. Diagram ini menggambarkan alur kontrol (*control flow*) dan alur data (*data flow*) diantara berbagai langkah atau aksi yang diperlukan untuk menyelesaikan suatu aktivitas. *Activity diagram* terlihat mirip dengan *flowchart* tetapi *activity diagram* dapat memperlihatkan *parallel behaviour* karena menggambarkan interaksi antara beberapa *use case*. Beberapa element penting dalam *activity diagram* diantaranya terdapat *nodes* dan *action* yang merepresentasikan langkah-langkah atau tindakan yang terjadi pada proses. *Edges* digunakan untuk menghubungkan *node-node* dan aliran data. *Initial node* dan *final node* menandai titik awal dan titik akhir aktivitas atau proses. *Decision nodes* untuk membuat percabangan dan *merge nodes* menggabungkan alur aktivitas [16].

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 1. Simbol *Activity Diagram* (sumber: <https://www.dicoding.com>)

### 2.3.3 Class Diagram

*Class diagram* merupakan diagram yang merepresentasikan elemen-elemen utama sistem seperti *class*, atribut dan operasi serta relasi antar *class*. Dalam pemrograman berorientasi objek (OOP) *class* merupakan sebuah konstruksi yang mendefinisikan tipe data abstrak. *Class* berfungsi dalam membuat *blueprint* atau cetakan untuk menciptakan objek-objek (*instances*) yang memiliki karakteristik dan perilaku serupa. Atribut merepresentasikan data atau properti yang dimiliki objek-objek dari *class* tersebut. Sedangkan operasi merupakan metode yang dapat dilakukan oleh objek-objek didalam *class* tersebut. Sedangkan relasi merupakan hubungan antar *class* seperti *multiplicity* [16].

## 2.4 PHP

PHP merupakan bahasa yang dapat dijalankan di berbagai sistem operasi seperti Linux, Windows, dan Mac OS, serta kompatibel dengan *web server* seperti Apache dan Microsoft IIS. Bahasa ini pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994 untuk keperluan pembuatan halaman web pribadi. Pada

awal pengembangannya, PHP merupakan singkatan dari *Personal Home Page*. Seiring berjalannya waktu, PHP mengalami berbagai pembaruan dan peningkatan fitur yang menjadikannya salah satu bahasa pemrograman yang banyak digunakan dalam pengembangan aplikasi berbasis web. Saat ini PHP lebih dikenal sebagai *PHP: Hypertext Preprocessor* [5].

PHP merupakan bahasa yang *powerful* yang didesain untuk membuat konten website bersama dengan HTML. PHP merupakan bahasa *server-side scripting* yang mampu membuat tampilan website yang dinamis. Selain itu kelebihan dari PHP adalah bisa dijalankan melalui *command-line scripting* untuk berbagai tugas sistem administrasi seperti *backup*. Bahasa ini mendukung berbagai database seperti MySQL, PostgreSQL, Oracle dan lain-lain. Saat ini sudah banyak *library* yang dikembangkan di PHP untuk memudahkan pengerjaan tugas-tugas umum seperti *error handling* dan lain-lain [18].

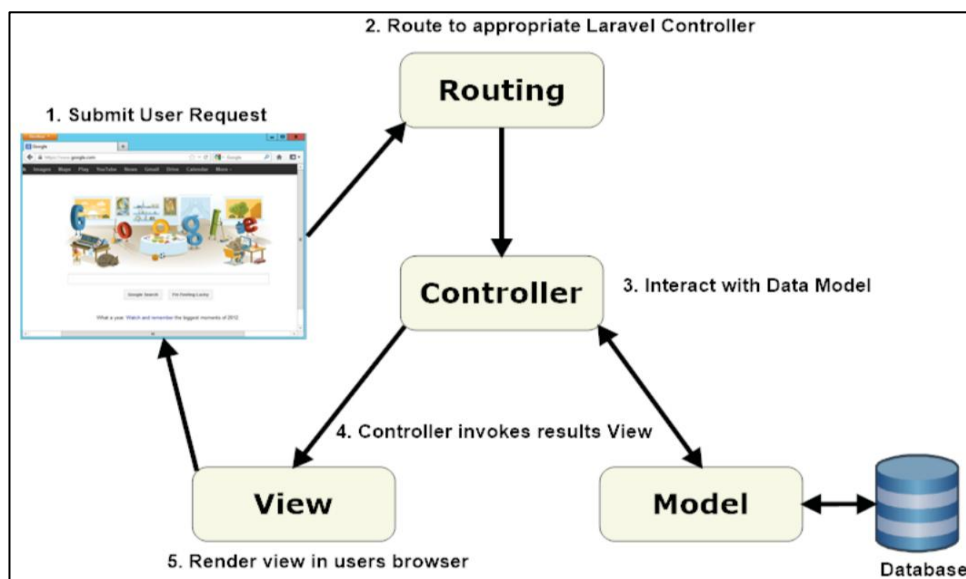
## 2.5 Laravel

Laravel adalah *framework* pengembangan web yang dikembangkan oleh Taylor Otwell. Laravel memungkinkan pengembang untuk membuat aplikasi *web* yang berjalan di berbagai platform dengan menggunakan bahasa pemrograman PHP. Sejak versi ke dua laravel, *framework* ini telah menyediakan fitur ORM (*Object Relational Mapping*) yang dinamakan dengan *Eloquent* [19]. ORM merupakan teknik yang digunakan untuk berinteraksi dengan database menggunakan objek yang memungkinkan pengelolaan data tanpa perlu menulis banyak kode SQL secara manual sehingga membuat proses pengembangan lebih cepat dan kode lebih mudah dibaca [20].

*Framework* ini menggunakan bahasa PHP dan menerapkan arsitektur berbasis MVC (*Model-View-Controller*) yaitu arsitektur yang memisahkan bagian *Model*, *View* dan *Controller*. *Model* ini digunakan untuk mengelola data dan logika aplikasi. *View* digunakan untuk menampilkan tampilan kepada pengguna, sedangkan *Controller* berfungsi untuk sebagai penghubung antara *Model* dan *View*. Dengan adanya sistem ini, perubahan pada suatu bagian tidak akan

mempengaruhi bagian lainnya, sehingga mempermudah pengembangan dan pemeliharaan aplikasi, serta memudahkan koneksi ke *database* [5].

Laravel bekerja dengan memproses permintaan pengguna yang dikirim melalui browser dan diterima oleh *web server*. Permintaan tersebut kemudian diteruskan ke *routing* untuk menentukan ke mana harus diarahkan. Selanjutnya *Controller* menerima permintaan dari *routing* lalu meminta bantuan *Model* untuk mengambil atau menyimpan data di *database*. Setelah data di proses, maka *controller* mengirimkan hasilnya ke *view* atau bagian yang mengatur tampilan halaman. *View* akan ditampilkan ke *browser* pengguna sehingga pengguna bisa melihat hasilnya yang memudahkan pengembangan aplikasi web. Berikut ilustrasi dari arsitektur Laravel [19].



Gambar 2 . Arsitektur Laravel (Sumber: E-book *Laravel Up and Running: A Framework for Building Modern Apps*) [19].

## 2.6 MySQL

Dalam membangun aplikasi seperti website, seringkali dibutuhkan tempat yang efisien untuk menyimpan informasi yang diperlukan. Oleh karena itu, diperlukan suatu tempat penyimpanan bernama *database* untuk menyimpan data tersebut. Database terbagi menjadi dua yaitu database *flat* dan *database* relasional. Database *flat* ini terdiri dari kumpulan data yang disimpan dalam bentuk teks

biasa seperti CSV, TXT dan sebagainya. Sedangkan *database* relasional terdiri dari tabel yang merupakan objek dasar untuk menyimpan data. Setiap tabel bisa memiliki hubungan atau relasi antar tabel lain yang dihubungkan dengan *Foreign Key*. *Database* relasional memiliki lebih banyak keunggulan dari pada *database flat* dikarenakan memudahkan manajemen data dengan struktur tabel yang mempermudah pelacakan informasi. Selain itu *database* relasional juga mampu mengurangi redundansi sehingga menghemat ruang penyimpanan di *hard disk* [21].

Salah satu *Relational Database Management System* (RDBMS) yang populer dan paling banyak digunakan adalah MySQL. MySQL dikembangkan oleh perusahaan bernama MySQL AB dari Swedia di tahun 1995. Beberapa kelebihan MySQL dibandingkan dengan RDBMS lain yaitu terletak pada kecepatan servernya yang lebih baik dibanding RDBMS yang lain. MySQL juga merupakan *software* yang *open-source* sehingga bebas digunakan tanpa harus membayar lisensinya. Selain dapat dijalankan di berbagai sistem operasi seperti *Linux* maupun *Windows*, ukuran memori databasenya juga tidak terbatas. Sehingga MySQL sangat cocok diterapkan untuk berbagai aplikasi mulai dari sistem yang kecil hingga besar [21].

## **2.7 Simple Mail Transfer Protocol (SMTP)**

SMTP atau *Simple Mail Transfer Protocol* merupakan sebuah protokol yang digunakan untuk mengirimkan email antar komputer atau *host* yang berbeda dalam jaringan TCP/IP (*Transmission Control Protocol/Internet Protocol*), yaitu jaringan yang bekerja pada lapisan aplikasi. Didalam SMTP terdapat dua *role* dengan peran yang berbeda yaitu *sender* dan *receiver*. *Sender* berperan sebagai klien yang memulai koneksi dua arah (TCP) dengan *receiver* untuk mengirim email. Sedangkan *receiver* bisa berupa penerima sebenarnya atau server perantara yang akan mengirimkan pesan sebelum sampai ke tujuan akhir. Setelah pesan tiba di server penerima, pesan tersebut dikirim ke kotak masuk penerima menggunakan protokol yang berbeda seperti POP3 (*Post Office Protocol 3*) atau IMAP (*Internet Message Access Protocol*) untuk mengambil dan mengakses email dari *server* penerima [22].

SMTP menggunakan *port* untuk mengirim email antar *server*. Setiap *port* memiliki kelebihan masing-masing. *Port 25* merupakan *port* yang tidak memiliki enkripsi sehingga tingkat keamanannya rendah. *Port 465* menggunakan enkripsi SSL (*Secure Socket Layer*). Namun *port 465* bukanlah standar *port* untuk SMTP. *Port 587* merupakan *port* yang paling direkomendasikan karena sudah menggunakan STARTTLS yang sudah di dukung oleh berbagai *email service*. Terakhir adalah *port 2525* yang bisa digunakan jika beberapa *port* sebelumnya diblokir. Sama seperti *port 465*, *port 2525* bukanlah *port* standar SMTP [23].

## 2.8 AdminLTE

AdminLTE merupakan template dashboard berbasis komponen UI (*User Interface*) untuk membuat antarmuka *dashboard* dengan tampilan yang responsif. Admin LTE cukup populer digunakan oleh pengembang karena bersifat *open source*. Template ini dibangun dengan pustaka dan plugin pendukung seperti jQuery, Popper.js, Font Awesome untuk ikon dan Chart.js. Beberapa kelebihan yang dimiliki AdminLTE adalah menyediakan komponen UI yang sudah sepenuhnya responsif untuk berbagai perangkat dan *browser* serta mudah untuk di kustomisasi misalnya dengan merubah warna atau tata letak. Selain menyediakan versi gratis, AdminLTE juga tersedia dalam versi berbayar yang menyediakan akses penuh untuk semua komponen dan template yang ada didalamnya. Dengan menggunakan AdminLTE, maka pengembang dapat menghemat waktu dalam membangun sebuah antarmuka *dashboard* [24].

## 2.9 PXP (*Personal Extreme Programming*)

*Personal Extreme Programming* merupakan sebuah metode pengembangan perangkat lunak berbasis *agile* yang dirancang untuk developer yang bekerja tanpa tim (*autonomous developer*). Metode ini merupakan gabungan dari konsep XP (*Extreme Programming*) dan PSP (*Personal Software Process*) [6]. *Extreme Programming* merupakan salah satu prinsip *agile* yang diterapkan dalam pengembangan software dengan melibatkan interaksi dan kolaborasi bersama klien. Proses atau tahapan XP terdiri dari empat aktivitas kerangka kerja yaitu:

1. *Planning* (Perencanaan) : mengidentifikasi kebutuhan pengguna berdasarkan *user stories* kemudian dilakukan perencanaan estimasi waktu atau jadwal untuk mengerjakan setiap kebutuhan tersebut.
2. *Design* (Perencanaan) : Perancangan dilakukan dengan prinsip kesederhanaan. Jika terdapat kendala, dapat dilakukan *spike solution* (prototipe sederhana).
3. *Coding* (Pengodean): Pengembangan kode dilakukan dengan pendekatan *Test-Driven Development* (TDD), yaitu membuat *unit test* sebelum menulis kode, serta menerapkan *pair programming* untuk meningkatkan kualitas perangkat lunak.
4. *Testing* (Pengujian): *Unit test* dijalankan secara rutin untuk mendeteksi kesalahan sejak dini, serta dilakukan *acceptance testing* untuk memastikan sistem sesuai dengan kebutuhan pengguna [25].

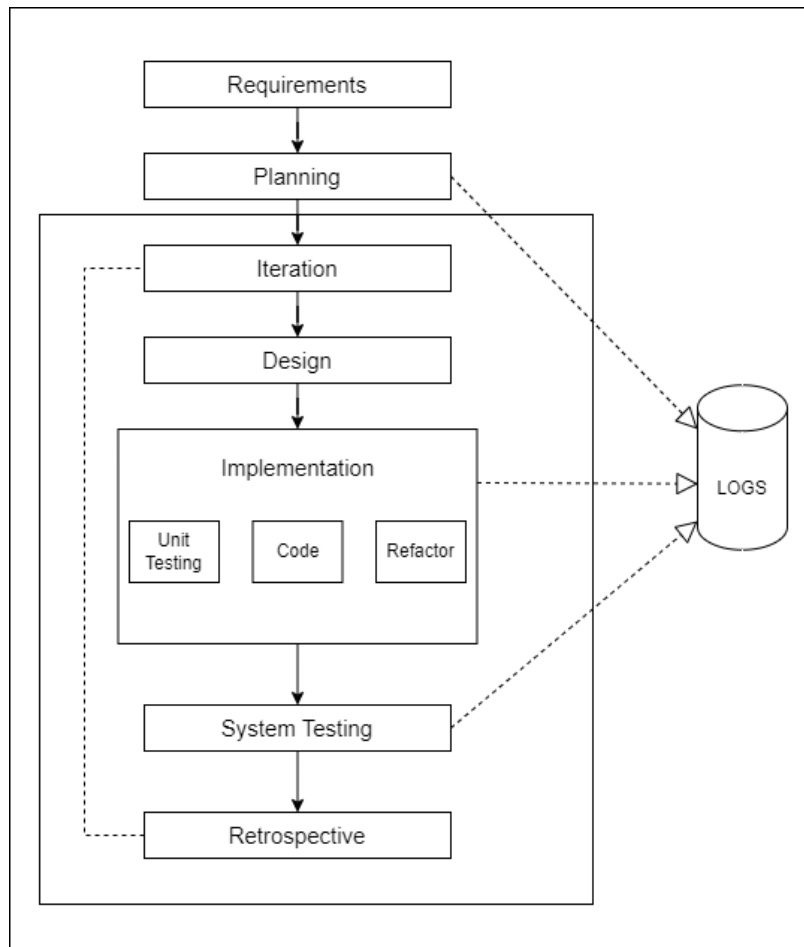
Sedangkan PSP (*Personal Software Process*) adalah metode pengembangan perangkat lunak yang ditunjukkan untuk individu. Metode ini menuntut pengembang untuk mengikuti proses yang terstruktur, mencatat setiap aktivitas, dan melakukan evaluasi berkala, sehingga dapat menghasilkan perangkat lunak yang berkualitas dan profesional. PSP terdiri dari 4 element utama yaitu yang pertama *script* yang merupakan panduan langkah-langkah pengerjaan untuk setiap fase yang terdiri dari tujuan, kriteria input output serta langkah yang harus dilakukan. Yang kedua adalah form yaitu template untuk mencatat dan menyimpan data secara konsisten. Form ini berisi *checklist* untuk meninjau desain, kode, estimasi waktu kerja dan *defect* (kekurangan) yang ditemukan. Yang ketiga adalah *measures* atau pengukuran. Ada 4 data utama yang diukur dalam PSP yaitu *time* (waktu kerja per fase), *size* (ukuran produk atau kode), *quality* (jumlah *bug*) dan *schedule* (rencana dan realisasi jadwal). Element yang keempat adalah *standards* yaitu membuat definisi yang konsisten agar data dapat dibandingkan misalnya seperti standar penulisan kode [26].

PXP menggabungkan kedua metode XP dan PSP dengan menghilangkan unsur *pair programming* dari XP menjadi pendekatan mandiri yang disiplin bagi *autonomous developer* atau pengembang individu serta menyederhanakan

dokumentasi berat PSP menjadi catatan personal yang lebih praktis sehingga memungkinkan pengembang individu menghasilkan perangkat lunak berkualitas tinggi secara efisien tanpa kehilangan fleksibilitas agile [6].

Penelitian sebelumnya pernah dilakukan oleh Iyawa dengan melakukan wawancara terhadap 10 *autonomous software developer* dari Namibia yang pernah menggunakan metode PXP dengan tujuan untuk memberikan gambaran manfaat dan tantangan yang dihadapi developer saat menggunakan PXP. Penelitian ini menggunakan dua kerangka teori yaitu *Diffusion of Innovation* (DoI) dari Rogers untuk mengukur *relative advantage*, *compatibility*, *complexity* dan *observability*. Serta *Technology Acceptance Model* (TAM) untuk mengukur *perceived usefulness* dan *perceived ease of use*. Hasilnya dapat disimpulkan bahwa metode PXP cocok diterapkan pada proyek dengan skala kecil hingga menengah. Namun keberhasilan proyek sangat bergantung pada keterampilan developer, karena kecepatan pengembangan ditentukan oleh kemampuan mereka [27].

*Personal Extreme Programming* memiliki beberapa fase dalam proses pengembangannya yang bersifat iteratif. Selama proses pengembangan, *developer* perlu mencatat aktivitas kedalam *logs files* untuk *planning*, durasi pengerjaan yang sesungguhnya, perbaikan kode hingga saran perbaikan. Hal ini berguna untuk menjadi prediksi yang lebih baik di proyek selanjutnya. Adapun fase pada *PXP* dapat dilihat pada Gambar 3.



Gambar 3. Tahapan Metode PXP [6].

### 1. *Requirements*

Ini merupakan tahap mengumpulkan dokumen kebutuhan fungsional dan non-fungsional. Tahap ini bersifat opsional jika kebutuhan sudah tersedia melalui *meeting* atau dokumen pengembangan sebelumnya. Dalam pengembangan sistem berbasis *Agile*, *user story* merupakan metode yang efektif untuk mendapatkan kebutuhan pengguna (*requirements*). *User story* berfungsi menggambarkan fitur yang diinginkan dari sudut pandang pengguna. Dalam metode *agile* setiap iterasi dilakukan dalam satu siklus yang pendek dan cepat sehingga diperlukan pembuatan dokumen kebutuhan yang singkat dan mudah dipahami tanpa perlu dokumentasi yang rumit. Format umum yang digunakan *user story* dapat ditulis dengan sebagai [siapa], saya ingin [tujuan atau fitur yang diinginkan], sehingga [manfaat atau alasan tujuan tersebut] [28].

## 2. *Planning*

Dalam tahap *planning*, dilakukan penyusunan tugas-tugas berdasarkan daftar kebutuhan serta menyusun estimasi waktu yang dibutuhkan untuk mengerjakan masing-masing tugas yang diukur menggunakan satuan *story point*. *Story point* merupakan satuan yang digunakan untuk mengukur kompleksitas atau ukuran dari suatu *user story* atau fitur. Setiap *user story* akan diberikan sejumlah poin secara relatif. Misalnya, jika sebuah *story* bernilai dua poin, maka tingkat usaha yang dibutuhkan diperkirakan dua kali lebih besar dibandingkan *story* yang bernilai satu poin. *Story point* juga digunakan untuk memperkirakan waktu yang dibutuhkan untuk pengerjaan proyek dengan menentukan *velocity*. *Velocity* merupakan total *story point* yang diselesaikan dalam satu iterasi [29].

## 3. *Iterations*

Iteration menandai awal sebuah iterasi kerja. Menurut Dzurov estimasi durasi iterasi dapat bervariasi mulai dari 1 – 3 minggu tergantung besarnya proyek. Satu iterasi merupakan gabungan dari beberapa *user story*. Satu iterasi memiliki jumlah *velocity* yang ditentukan berdasarkan kemampuan pengembang untuk menentukan berapa lama proses iterasi akan berlangsung.

## 4. *Design*

Tahap ini pengembang membuat model sistem sederhana dari modul atau kelas sistem berdasarkan kebutuhan saat ini tanpa memperhatikan kebutuhan atau fitur-fitur yang kemungkinan akan muncul nantinya.

## 5. *Implementation*

Tahap implementasi merupakan tahap dimana kode yang sebenarnya di buat. Tahap ini meliputi tiga proses. Yang pertama adalah *unit testing* yaitu pengujian unit kecil untuk memastikan kode bekerja dengan semestinya. *Unit testing* bisa dilakukan secara manual atau otomatis. Namun penerapan unit testing manual memakan waktu yang lama serta sering terjadi kesalahan sehingga pengujian otomatis lebih sering diterapkan. Proses kedua yaitu koding atau pembuatan kode dan tahap ketiga adalah *refactoring* yaitu menyempurnakan atau merapikan kode.

## 6. *System Testing*

*System testing* merupakan tahap dimana semua fitur sistem yang telah dibangun akan di uji. Apabila ditemukan fitur yang kurang, salah implementasi atau tidak sesuai, maka harus segera diperbaiki sebelum masuk ke tahap selanjutnya.

## 7. *Retrospective*

*Restrospective* merupakan tahap akhir dimana segala kesalahan seperti ketidaksesuaian waktu pengerjaan, cacat program atau *bug* dikumpulkan untuk dijadikan evaluasi dan perbaikan untuk proyek pengembangan selanjutnya.

### 2.10 *MoSCoW Prioritazion*

Teknik *MoSCoW Prioritazion* adalah metode penentuan prioritas kebutuhan dalam suatu proyek terutama pengembangan perangkat lunak atau manajemen produk. Dengan memahami prioritas kebutuhan, maka pengembang dapat mengetahui kebutuhan mana yang harus dikerjakan terlebih dahulu. *MoSCoW* sendiri merupakan akronim dari empat kategori prioritas, yaitu *Must Have*, *Should Have*, *Could Have* dan *Won't Have*. *Must have* adalah kebutuhan utama yang sangat penting dan harus segera diselesaikan. Apabila kebutuhan ini tidak terpenuhi, maka proyek bisa dianggap gagal. *Should have* merupakan kebutuhan yang sangat penting tetapi tidak sepenting *must have* namun sebaiknya tetap harus dipenuhi dalam batas waktu proyek. *Could have* adalah kebutuhan yang apabila di terapkan, maka akan meningkatkan kepuasan pengguna. Sehingga *could have* boleh ada dan boleh ditinggalkan apabila tidak memberikan dampak apapun. *Won't have* merupakan kebutuhan yang tidak akan dimasukan kedalam proyek untuk saat itu juga namun dapat diterapkan di masa depan tergantung pada kebutuhan dan perubahan kondisi [30].

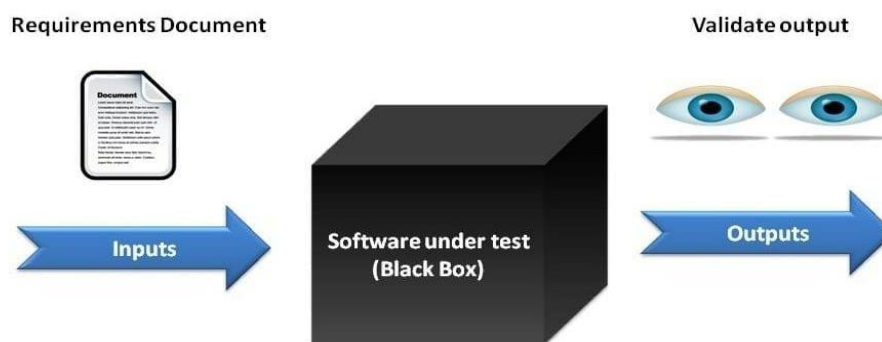
### 2.11 *Low Fidelity Prototype*

*Low Fidelity Prototype* merupakan representasi desain antarmuka sistem sederhana yang berfokus pada tata letak dan fungsi dasar tanpa detail visual atau elemen grafis yang kompleks. Tujuan dari pembuatan antarmuka menggunakan *low fidelity* adalah untuk mengurangi waktu yang digunakan selama tahap desain antarmuka serta membantu pengguna untuk memahami konsep sistem yang

dikembangkan secara cepat dan sederhana. *Low fidelity* dapat dibuat dengan bahan yang sederhana seperti kertas, sketsa atau digital *wireframe* [31].

### 2.12 Black-Box Testing

*Black box testing* adalah teknik pengujian yang berfokus pada fungsionalitas aplikasi tanpa melihat implementasi kode didalamnya. Pengujian *black-box* dilakukan melalui simulasi penggunaan sistem dengan memasukkan data tertentu, kemudian mengamati respons yang dihasilkan. Selanjutnya, output sistem diverifikasi dengan membandingkannya terhadap output yang diharapkan [32]. Beberapa alasan yang menjadikan *black box testing* banyak digunakan adalah seorang *tester* atau penguji tidak perlu memahami kode program karena testing dilakukan berdasarkan sudut pandang *customer*. Sehingga proses pengujian menggunakan *black box testing* bisa lebih cepat. Meskipun demikian *black box testing* juga memiliki beberapa kekurangan. Jika *requirement* tidak jelas, maka penguji akan kesulitan membuat skenario pengujian yang tepat. Selain itu, beberapa *error* yang tersembunyi kemungkinan tidak terdeteksi karena pengujian hanya berfokus pada output tanpa memperhatikan kode internal program [7]. Alur proses pengerjaan *black box testing* dapat dilihat pada Gambar 4.



Gambar 4. Alur Proses *Black Box Testing* [33]

Berdasarkan Gambar 4 dapat dijelaskan bahwa alur pengujian *black box* terdiri dari:

1. Input: Penguji memilih input yang valid berdasarkan *requirement* dan dokumen spesifikasi untuk menentukan skenario pengujian.

2. *Software under test*: Penguji mengeksekusi skenario pengujian yang telah di buat sebelumnya tanpa memedulikan cara kerja internal sistem tersebut.
3. Output: Setelah pengujian selesai dan menghasilkan output yang diinginkan, selanjutnya penguji akan membuat laporan akhir [33].

### 2.13 System Usability Scale (SUS)

*System Usability Scale* (SUS) adalah sebuah metode yang digunakan untuk menguji kebergunaan suatu sistem. Pengukuran SUS menggunakan skala *Likert* dengan 5 poin yang terdiri dari 10 pertanyaan dimana pertanyaan negatif dan pertanyaan positif masing-masing disusun secara bergantian dengan tujuan supaya responden membaca keseluruhan pertanyaan [8]. Adapun pertanyaan negatif terdapat pada nomor genap seperti 2,4,6,8, dan 10. Sedangkan pertanyaan positif terdapat pada nomor ganjil seperti 1,3,5,7, dan 9 [34]. Metode SUS banyak digunakan karena metode ini dikenal efektif untuk mengevaluasi sistem dengan cepat tanpa melakukan analisis *usability* lebih dalam. Tabel 2 menjelaskan 10 daftar pertanyaan yang digunakan pada pengujian *System Usability Scale* (SUS).

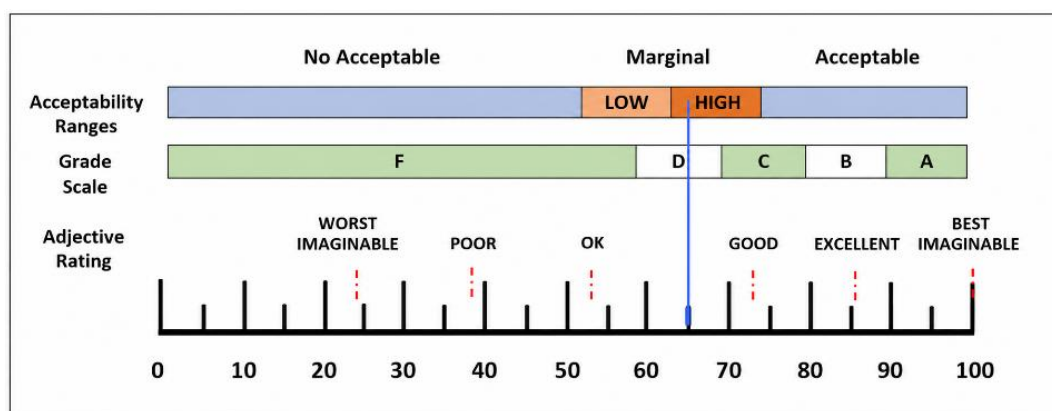
Tabel 2. Daftar Pertanyaan Pada Pengujian *System Usability Scale* (SUS)

No.	Pertanyaan	1	2	3	4	5
1.	Saya rasa saya akan sering menggunakan sistem ini					
2	Saya merasa sistem ini terlalu rumit					
3	Saya rasa sistem ini mudah digunakan					
4	Saya rasa saya membutuhkan bantuan orang teknis untuk dapat menggunakan sistem ini					
5	Saya merasa berbagai fungsi dalam sistem ini terintegrasi dengan baik.					
6	Saya pikir sistem ini terlalu banyak ketidaksesuaian.					
7	Saya membayangkan sebagian besar orang akan belajar menggunakan sistem ini dengan cepat					
8	Saya merasa sistem ini sangat sulit digunakan.					
9	Saya merasa sangat percaya diri saat menggunakan sistem ini.					
10	Saya perlu mempelajari banyak hal sebelum dapat mulai menggunakan sistem ini.					

Skor SUS dihitung dengan menjumlahkan skor dari masing-masing pertanyaan dimana setiap pertanyaan memiliki kontribusi skor antara 1 sampai 5. Adapun

pertanyaan dengan nomor 1, 3, 5, 7, 9 masing-masing skor kontribusi dihitung dengan rumus nilai yang dipilih dikurangi dengan 1. Sedangkan pertanyaan dengan nomor 2, 4, 6, 8, 10 dihitung dengan rumus 5 dikurangi dengan nilai yang dipilih. Setelah menjumlahkan semua skor, maka kalikan jumlah tersebut dengan nilai 2,5 untuk mendapatkan skor akhir SUS [8].

SUS memiliki tiga perspektif penilaian skor akhir yang terdiri dari *acceptability range*, *grade scale*, dan *adjective ratings* seperti yang terlihat pada Gambar 5. *Acceptability* merupakan sebuah penilaian untuk melihat tingkat penerimaan pengguna terhadap sistem. *Acceptability* ini terbagi menjadi tiga yaitu *not acceptable*, *marginal*, dan *acceptable*. Suatu sistem mendapat predikat *not acceptable* apabila skor akhir SUS berada diantara 0 – 50. Kemudian predikat *marginal* memiliki rentang skor 51 - 60 untuk marginal low dan 61 - 70 untuk marginal high. Dan predikat tertinggi yaitu *acceptable* memiliki nilai skor akhir antara 70 -100. Berikut ini merupakan gambar yang menampilkan penilaian SUS dalam 3 perspektif [35].



Gambar 5. Penilaian SUS dalam 3 Perspektif [35].

## 2.14 Apache JMeter

Apache JMeter merupakan sebuah aplikasi *open source* berbasis *Java* yang diciptakan oleh Apache Software Development. Aplikasi ini digunakan untuk melakukan pengujian performa hingga pengujian beban (*load testing*) pada berbagai aplikasi terutama aplikasi website dan layanan jaringan. Dalam pengujian JMeter mensimulasikan berbagai protokol komunikasi seperti

HTTP/HTTPS, FTP dan layanan web sehingga fleksibel untuk digunakan dalam berbagai skenario pengujian [9].

Dibandingkan dengan perangkat lunak pengujian lainnya seperti Soap UI, Apache JMeter memiliki beberapa kelebihan seperti bersifat *open source*, mendukung berbagai *plugin*, serta mampu melakukan berbagai jenis pengujian secara simultan. Namun Apache JMeter juga memiliki beberapa keterbatasan, seperti proses konfigurasi awal yang relatif lebih kompleks dan membutuhkan waktu instalasi yang lebih lama dibandingkan beberapa tools pengujian lainnya [36].

Salah satu komponen penting dari JMeter adalah kemampuannya mensimulasikan perilaku pengguna secara paralel dengan *thread group* dimana setiap *thread* mewakili satu pengguna virtual yang mengirim permintaan sesuai skenario yang ditetapkan. Hasil pengujian digunakan sebagai analisis pengembangan perangkat lunak untuk mengevaluasi stabilitas, kecepatan, respons ketahanan sistem terhadap beban tinggi [37].

### 2.15 Penelitian Terkait

Untuk menunjang penelitian ini, diperlukan teknik pengumpulan data berupa studi literatur dari penelitian terkait. Tujuannya adalah untuk mendukung pelaksanaan penelitian dan menghindari plagiarisme. Tabel 3 berikut merupakan penjabaran beberapa penelitian relevan yang terdahulu.

Tabel 3. Penelitian Terkait

No.	Peneliti, Tahun Terbit	Judul	Hasil
1.	Iskandar A. <i>et al.</i> (2024)	<i>Design of A Web-Based Information System for New Student Registration in Vocational High</i>	Pengembangan sistem pendaftaran siswa baru berbasis web menggunakan bahasa pemrograman PHP dan MySQL sebagai <i>database</i> . Hasil pengujian <i>Black box</i> menunjukkan sistem dinilai mudah dipahami, aman, responsif. Hasil survei dari 400 responden menunjukkan 37,5% merasa sangat puas dan 62,5% menyatakan puas

No.	Peneliti, Tahun Terbit	Judul	Hasil
			terhadap sistem [38].
2.	Ahmed M. <i>et al</i> (2022)	<i>Web Based Student Registration and Exam Form Fill-up Management System for Educational Institute</i>	Pengembangan sistem manajemen mahasiswa yang meliputi pendaftaran calon mahasiswa, pengisian formulir ujian akhir semester secara online dengan fasilitas pembayaran daring. Web dikembangkan dengan bahasa pemrograman PHP, HTML, CSS, JavaScript, Ajax dan MySQL sebagai <i>database</i> . Hasilnya sistem mampu mengurangi beban kerja, mempercepat proses administrasi dan menerapkan digitalisasi dokumen [39].
3.	Ozzi Suria (2024)	<i>A Statistical Analysis of System Usability Scale (SUS) Evaluations in Online Learning Platform</i>	Penelitian ini mengevaluasi kegunaan sistem <i>e-learning</i> berbasis Moodle menggunakan <i>System Usability Scale</i> (SUS) dengan melibatkan 120 responden mahasiswa. Hasilnya menunjukkan rata-rata skor SUS sebesar 71.52, yang menempatkan sistem dalam kategori 'C+' menurut skala Sauro dan Lewis, mengindikasikan bahwa meskipun kegunaan sistem dapat diterima, masih diperlukan peningkatan yang substansial [40].
4	P. Pahrizal. <i>et al.</i> (2022)	<i>The Use of PHPMailer in E-SPT System (Electronic Task Order) Parking Court at The Transportation Office Bengkulu City</i>	Penelitian ini mengembangkan sistem E-SPT (Surat Perintah Tugas Elektronik) berbasis website untuk juru parkir pada Dinas Perhubungan Kota Bengkulu. Latar belakang penelitian ini adalah proses pendaftaran juru parkir yang masih dilakukan secara manual, pembuatan SPT yang masih dibuat satu per satu, dan surat perintah tugas yang tidak dalam format yang terstandar. Sistem ini menggunakan

No.	Peneliti, Tahun Terbit	Judul	Hasil
			library PHPMailer untuk mempermudah pengiriman data E-SPT ke email juru parkir secara otomatis. Hasil penelitian berhasil membangun sistem E-SPT berbasis website untuk juru parkir, di mana sistem mampu mengirimkan pesan pengumuman melalui website ke email pengguna secara otomatis setiap kali pengumuman baru ditambahkan [41].
5.	Palyus Fiqar, T. dkk (2023)	Implementasi Metode <i>Personal Extreme Programming</i> dalam Pengembangan Sistem Informasi Manajemen Hak Paten pada Sentra HKI ITK	Implementasi metode MoSCoW untuk menentukan prioritas kebutuhan pada <i>user story</i> . Hasil dari implementasi ini berhasil menentukan prioritas sehingga membentuk kelompok iterasi pengerjaan. Iterasi dikerjakan secara berurutan mulai dari iterasi yang memiliki prioritas tertinggi hingga iterasi dengan prioritas terendah [42].
6.	Mahadi Y dkk. (2024)	<i>Design Of Student Register Application Using Laravel 8 Framework</i>  (Jurnal Nasional, Sinta 5)	Perancangan aplikasi pendaftaran ulang siswa berbasis web SDN Tegallega. Sistem dibangun menggunakan <i>framework</i> Laravel, dengan bahasa pemrograman PHP dan MySQL sebagai <i>database</i> . Hasil implementasi menunjukkan peningkatan efisiensi dalam proses daftar ulang, pengurangan kesalahan input data, serta digitalisasi dokumen dan mempermudah pengambilan keputusan administratif [43].
7.	Andri Nata dkk.	<i>Workshop</i>	Workshop pembuatan admin panel

No.	Peneliti, Tahun Terbit	Judul	Hasil
	(2023)	Pengenalan <i>Framework</i> CSS Dalam Pembuatan Halaman Admin Aplikasi Web Menggunakan Admin LTE	menggunakan Admin LTE di SMK N 1 Setia Janji Kabupaten Asahan, Sumatera Utara, menunjukkan bahwa <i>framework</i> ini mempercepat pengembangan karena menyediakan komponen antarmuka siap pakai, sehingga pengguna tidak perlu membuat desain dari awal [24].
8.	López-Gorozabeet al, (2021)	<i>Bootstrap as a Tool for Web Development and Graphic Optimization on Mobile Devices</i>	Evaluasi Bootstrap sebagai alat <i>re-engineering</i> sistem web "EducArte Comunidad" yang berinteraksi dengan HTML5, CSS, dan JQuery. Penelitian ini menjelaskan bahwa Bootstrap diciptakan oleh Twitter pada 2011 dengan fungsi menyesuaikan resolusi website agar dapat ditampilkan dengan benar di berbagai resolusi layar perangkat mobile, dan diterapkan untuk pengembangan aplikasi web dalam modalitas multiplatform [44].
9.	Desy Intan Permata-sari. Dkk (2020)	Pengujian Aplikasi Menggunakan Metode <i>Load Testing</i> dengan <i>Apache Jmeter</i> pada Sistem Informasi Pertanian	Penelitian ini membahas pengujian beban ( <i>load testing</i> ) pada sistem informasi pertanian. Alat yang digunakan dalam pengujian ini adalah Apache JMeter, yaitu perangkat lunak <i>open source</i> yang digunakan untuk melakukan simulasi akses pengguna serta mengukur kinerja sistem. Hasil pengujian ini digunakan untuk mengetahui bagaimana sistem berperilaku ketika diakses oleh beberapa pengguna secara bersamaan [37].
10.	N. Susilo dan D. Cahyono (2025)	Perancangan Platform	Penelitian ini membahas perancangan platform digital administrasi desa berbasis

No.	Peneliti, Tahun Terbit	Judul	Hasil
		Digital Administrasi Desa Berbasis Web Melalui Pengujian UML di Balai Desa Waruangering	web untuk mempermudah pengelolaan administrasi umum dan surat-menyurat secara terintegrasi. Sistem dirancang menggunakan model UML ( <i>Unified Modeling Language</i> ) seperti <i>use case</i> , <i>activity</i> , <i>class</i> , dan <i>sequence diagram</i> . Dari hasil uji kelayakan model UML menggunakan kuisisioner menunjukkan bahwa 83,88% responden merasa visualisasi UML dapat membantu memahami alur sistem dan mendukung proses perancangan sistem [45].

### **III. METODOLOGI PENELITIAN**

#### **3.1 Waktu dan Tempat Penelitian**

Waktu dan tempat pelaksanaan penelitian dilakukan pada:

Waktu Penelitian : Oktober 2025 sampai dengan Mei 2026  
Tempat Penelitian : Laboratorium Komputer Universitas Lampung  
dan MAS Al-Muhajirin Tugumulyo

Tabel 4 berikut menunjukkan rincian jadwal kegiatan penelitian yang dilakukan.





### 3.2 Alat dan Bahan Penelitian

Adapun alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 5. Alat (*Hardware* dan *Software*) pengembangan sistem

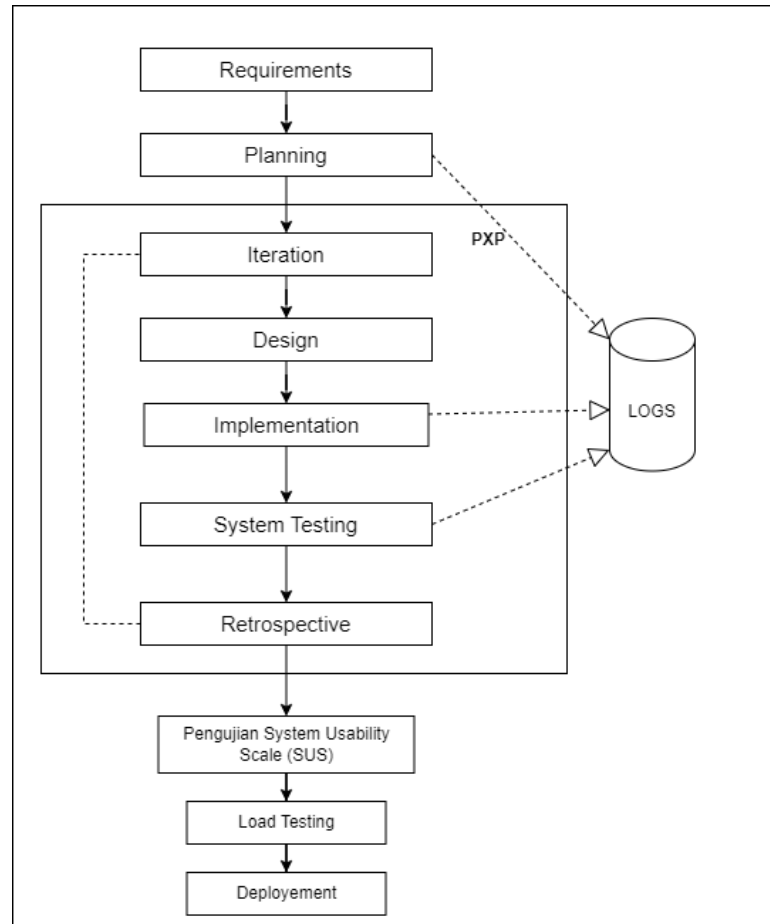
No	Perangkat	Spesifikasi	Kegunaan
1	Laptop	Intel Core-i5-1135G7 @ 2.40GHz, RAM 8 GB	Perangkat Keras yang digunakan dalam proses pengembangan
2.	Visual Studio Code	Version 1.93.1	Perangkat Lunak sebagai <i>text editor</i>
3.	Laravel	Version 12	<i>Framework</i> untuk membangun aplikasi
4.	Laragon	6.0	Perangkat lunak sebagai <i>server</i> lokal
5.	Figma	Version 125.10.5	Perangkat lunak untuk mendesain UI/UX
6.	Admin LTE	Version 3	Template antarmuka dashboard
7.	Draw IO	2.5.1	Perangkat lunak sebagai <i>design requirement</i>
8.	Browser	Google Chrome	Peramban Website
9.	Apache JMeter	Version 5.6.3	Perangkat lunak untuk pengujian beban

Tabel 6. Bahan Pengembangan Sistem

No.	Nama	Kegunaan
1	Petunjuk Teknis Penerimaan Peserta Didik Baru Madrasah	Sebagai data yang akan digunakan dalam aplikasi

### 3.3 Tahapan Penelitian

Beberapa tahapan pada penelitian ini dapat dilihat berdasarkan diagram dibawah ini.



Gambar 6. Tahapan Penelitian [6]

#### 3.3.1 Requirements

Tahap pertama dari penelitian yang dilakukan adalah *requirements* yaitu melakukan analisis kebutuhan pengguna yang akan diterapkan pada sistem informasi pendaftaran peserta didik baru. Dalam penelitian ini, data diperoleh dengan studi literatur dan wawancara.

Penerapan metode studi literatur dilakukan dengan membaca dan memahami literatur-literatur sebelumnya yang berkaitan dengan penelitian ini seperti jurnal, buku, dokumentasi teknologi serta menentukan alat yang akan digunakan untuk pengembangan sistem. Proses studi literatur dilakukan dengan memanfaatkan teknologi internet yaitu *google scholar* dan *google books*.

Proses wawancara dilakukan dengan bertatap muka langsung dengan narasumber dengan tujuan untuk menganalisis dan memahami kebutuhan dan permasalahan yang ada. Hasil dari analisis ini kemudian digambarkan dalam bentuk *user stories* yang menjadi dasar dalam menyusun gambaran umum sistem yang sedang berjalan dan merumuskan kebutuhan sistem baru yang akan dikembangkan. Adapun analisis yang dilakukan pada fase ini mencakup:

- a. Analisis sistem berjalan, yaitu menganalisis proses bisnis yang berlangsung saat ini, alur kerja pendaftaran, serta mengidentifikasi kekurangan dan permasalahan dari sistem manual.
- b. Analisis kebutuhan sistem, yaitu menganalisis permasalahan yang telah diidentifikasi sebelumnya, kemudian disusun beberapa rekomendasi kebutuhan sistem baru berdasarkan hasil wawancara. Rekomendasi ini ditujukan untuk mengatasi kendala yang ada serta mendukung proses administrasi yang lebih efisien.

### **3.3.2 Planning**

Tahap selanjutnya adalah *planning*, yaitu menyusun daftar tugas yang akan dikerjakan berdasarkan hasil *requirements* yang telah dibuat sebelumnya. Dalam tahapan ini, *user stories* dikelompokkan ke dalam beberapa iterasi. Proses pengelompokan ini didasarkan pada dua kriteria utama yaitu tingkat prioritas fitur serta estimasi kompleksitas dan waktu pengerjaan yang diukur secara kuantitatif menggunakan satuan *story point* yang bersifat relatif. Penentuan prioritas kebutuhan sistem dilakukan dengan pendekatan *MoSCoW* untuk mengelompokkan yang mengklasifikasikan kebutuhan sistem ke dalam empat kategori utama (*Must have, Should have, Could have, Won't have*).

### **3.3.3 Iterations Initialization**

*Iteration Initialization* merupakan tahap awal dalam proses pengembangan dari setiap iterasi. Pada tahapan ini, kebutuhan fungsional yang dikumpulkan dari *user stories* dijabarkan kedalam bentuk diagram dengan menggunakan *Unified Modeling Language* (UML) untuk memberikan gambaran visual tentang rancangan struktur dan cara kerja fitur. Beberapa diagram yang digunakan

diantaranya ada *use case* untuk menggambarkan hubungan interaksi antara aktor dengan sistem dan *activity diagram* untuk menggambarkan alur kerja atau proses bisnis dalam sistem.

### **3.3.4 Design**

Tahap desain dilaksanakan melalui pembuatan rancangan struktur basis data yang akan diintegrasikan ke dalam sistem dengan memanfaatkan pemodelan berupa *class diagram*. *Class diagram* tersebut berfungsi untuk menggambarkan hubungan antara berbagai entitas yang terdapat dalam sistem. Di samping itu, pada fase ini juga dilakukan perancangan antarmuka sistem dengan menerapkan teknik *low fidelity* atau dikenal sebagai desain berkualitas rendah. Penerapan desain ini bertujuan agar pembaca dapat dengan mudah memahami konsep tampilan dari program yang sedang dikembangkan. Penggunaan desain *low fidelity* juga memiliki keunggulan karena dapat mengurangi waktu yang diperlukan, sehingga memudahkan pengembangan sistem terutama jika dikerjakan oleh pengembang tunggal.

### **3.3.5 Implementation**

*Implementation* merupakan tahapan dimana pengembang mengimplementasikan desain kedalam bentuk kode program. Tahap *implementation* akan mengembangkan fitur secara bertahap sesuai dengan iterasi yang telah disusun pada tahap *planning*. Pada tahap ini memanfaatkan beberapa *tools* seperti Visual Studio Code sebagai teks editor, *Laravel* sebagai *framework* pengembangan sistem, *AdminLTE* sebagai template *dashboard* admin, *bootstrap* sebagai *framework* CSS serta HTML sebagai bahasa penanda. Selain itu pengembangan sistem PPDB ini juga memanfaatkan SMTP sebagai media untuk mengirimkan notifikasi melalui email kepada wali calon siswa atau calon siswa untuk menyampaikan informasi status pendaftaran dan pengumuman hasil seleksi.

### **3.3.6 System Testing**

Pada tahap ini dilakukan pengujian sistem terhadap seluruh fitur yang telah diimplementasikan. Pengujian dilakukan dengan metode *black box testing*. *Testing* ini cukup efisien untuk diterapkan dalam sistem dengan skala yang besar maupun menengah. Tahap awal pengujian dimulai dengan menyusun skenario pengujian

yang memuat daftar fitur sistem yang akan diuji berdasarkan sudut pandang pengguna. Dari hasil pengujian ini nantinya didapatkan perbandingan jumlah fitur yang lolos pengujian dengan jumlah keseluruhan fitur yang tersedia dalam sistem. Nilai akhir dari pengujian tersebut akan digunakan sebagai dasar untuk menentukan apakah sistem layak, dapat diterima, dan sesuai dengan kebutuhan pengguna.

### **3.3.7 Retrospective**

Pada tahap ini dilakukan evaluasi selama fase iterasi pengembangan sistem yang mencakup kesalahan-kesalahan yang terjadi seperti ketidaksesuaian antara estimasi waktu pengerjaan dengan waktu pengerjaan sesungguhnya. Pada tahap ini juga mengevaluasi penyebab hambatan tersebut supaya kesalahan ini tidak terjadi di iterasi selanjutnya.

### **3.3.8 Pengujian System Usability Scale (SUS)**

Dalam tahap pengujian *System Usability Scale* (SUS), digunakan 10 pertanyaan standar yang dibagikan kepada responden melalui *Google Form* untuk memperoleh skor penilaian terhadap sistem. Skor penilaian ini selanjutnya dihitung berdasarkan aturan pada metode SUS untuk menentukan nilai kebergunaan (*acceptability*) pada aplikasi. Adapun responden yang terlibat terbagi menjadi dua kategori sesuai dengan dua peran yang ada pada sistem yaitu admin dan wali calon siswa atau calon siswa. Pembagian ini bertujuan agar masing-masing peran dapat menilai kebergunaan sistem berdasarkan pengalaman dan fungsi yang mereka gunakan. Adapun jumlah responden SUS yang dibutuhkan sekitar 12 orang. Hal ini berdasarkan penelitian yang dilakukan oleh Tulis dan Stetson yang melakukan pengujian dengan mengumpulkan 123 responden untuk mengevaluasi dua situs web finansial. Hasil dari penelitian ini menunjukkan bahwa pada ukuran sampel 10-12 orang, kuesioner SUS memberikan hasil yang benar dimana skor SUS mencapai titik *asymptote* atau stabil pada  $n = 12$ , artinya, di bawah 12 orang, skor SUS sangat fluktuatif yang mana dapat berubah drastis jika ada 1 orang memberikan penilaian buruk atau baik. Sedangkan di atas 12 orang, nilai rata-rata mulai stabil dan tidak banyak berubah [46], [47].

### **3.3.9 Pengujian Beban (*Load Testing*)**

Pengujian beban (*load testing*) merupakan tahapan pengujian sistem yang bertujuan untuk mengetahui kinerja dan waktu respons sistem ketika diakses oleh banyak pengguna secara simultan. Pengujian ini dilakukan untuk memastikan bahwa sistem PPDB mampu menangani beban akses pengguna, khususnya pada proses-proses utama seperti login dan pengisian formulir pendaftaran. Pada penelitian ini, pengujian beban dilakukan menggunakan aplikasi Apache JMeter yang berfungsi untuk mensimulasikan user secara virtual (*virtual user*) yang mengakses sistem secara bersamaan. Pengujian dilakukan dengan mensimulasikan beberapa variasi jumlah virtual user untuk melihat pengaruh peningkatan beban terhadap performa sistem.

### **3.3.10 Deployment**

Tahap *deployment* merupakan tahap terakhir dalam proses pengembangan sistem dimana aplikasi yang telah selesai dikembangkan akan di implementasikan ke lingkungan produksi. Pada tahap ini dibutuhkan layanan hosting berbayar untuk memindahkan aplikasi ke lingkungan produksi agar aplikasi dapat diakses oleh publik.

## V. PENUTUP

### 5.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, terdapat beberapa kesimpulan yang diambil diantaranya:

1. Sistem informasi PPDB berbasis website untuk MAS Al-Muhajirin Tugumulyo berhasil dikembangkan menggunakan *framework* Laravel dengan metode *Personal Extreme Programming* (XP) melalui 5 iterasi selama 14,5 minggu, di mana seluruh 20 *user stories* sebagai kebutuhan fungsional yang telah ditetapkan berhasil diimplementasikan.
2. Berdasarkan hasil pengujian menggunakan metode *Black Box*, sebanyak 20 *user stories* yang diuji melalui 53 skenario pengujian menunjukkan kesesuaian antara hasil pengujian dan hasil yang diharapkan. Sehingga dapat disimpulkan secara fungsional sistem telah berjalan dengan baik.
3. Berdasarkan pengujian tingkat kebergunaan menggunakan *System Usability Scale* (SUS) oleh 12 responden, maka didapatkan skor 84,79. Hal ini menunjukkan bahwa sistem ini telah memenuhi nilai kebergunaan dengan predikat *acceptable* atau dapat diterima.
4. Berdasarkan hasil pengujian beban menggunakan Apache JMeter pada lingkungan staging lokal, sistem mampu menangani 100 pengguna secara simultan tanpa terjadi *error* dan waktu respons rata-rata masih berada di bawah 3 detik, sehingga sistem dapat dikatakan berfungsi dengan baik pada beban maksimal untuk satu periode PPDB.

## 5.2 Saran

1. Sistem ini dapat dikembangkan lebih lanjut agar dapat terintegrasi dengan sistem pendidikan nasional untuk mempermudah sinkronisasi data.
2. Dikarenakan pengujian sistem masih dilakukan pada lingkungan lokal yang belum sepenuhnya merepresentasikan lingkungan produksi dan saat ini sekolah masih menggunakan layanan *shared hosting*, disarankan agar sistem diimplementasikan pada *server* dengan spesifikasi yang lebih tinggi seperti *Virtual Private Server* (VPS) untuk memperoleh performa dan skalabilitas sistem yang lebih baik.
3. Sistem ini sebaiknya dikembangkan lebih lanjut dengan mengintegrasikan fitur keamanan lanjutan seperti pencegahan *brute force* dan filtrasi *file* berbahaya untuk keamanan aplikasi yang lebih baik.

## DAFTAR PUSTAKA

- [1] S. Syamsurijal, "Titik Temu Pendidikan dan Pembangunan Sumber Daya Manusia Berdaya Saing," *Edu Cendikia: Jurnal Ilmiah Kependidikan*, vol. 3, no. 3, pp. 545–553, 2024, doi: <https://doi.org/10.47709/educendikia.v3i03.3398>.
- [2] H.Hujaemah, "Pemberdayaan Walikelas Untuk Meningkatkan Partisipasi Siswa Dalam Melaksanakan PJJ Ramadhan," *Jurnal Perseda (Jurnal Pendidikan Guru Sekolah Dasar)*, vol. 3, no. 2, pp. 88–94, Aug. 2020, doi: <https://doi.org/10.37150/perseda.v3i2.903>.
- [3] R. M. Artikasari, A. S. Rodiyah, F. P. Nabila, A. A. Safitri, dan A. A. Siswoyo, "Strategi penerimaan peserta didik baru dalam upaya meningkatkan jumlah pendaftar di SDN Tanjung Jati 2," *Jurnal Media Akademik (JMA)*, vol. 2, no. 12, Desember 2024. Doi: <https://doi.org/10.62281/v2i12.1289>
- [4] W. Warta dan J. Janaenah, "Analisis Perbandingan Efektivitas Sistem PPDB Konvensional dan Online: Kajian terhadap Aspek Waktu, Biaya, dan Akurasi Data," *Journal of Human And Education*, vol. 4, no. 6, pp. 433–438, 2024.
- [5] S. B. Uzayr, *PHP: The Ultimate Guide*, 1st ed. Florida: CRC Press, 2023.
- [6] Y. Dzhurov, I. Krasteva, and S. Ilieva, "Personal Extreme Programming-An Agile Process for Autonomous Developers". *Faculty of Mathematics and Informatics, Sofia University, Bulgaria*, Tech. Rep., 2009.
- [7] A. Maspupah, "Literature Review: Advantages And Disadvantages Of Black Box And White Box Testing Methods," *Jurnal Techno Nusa Mandiri*, vol. 21, no. 2, pp. 151–162, Sep. 2024. Doi: <https://doi.org/10.33480/techno.v21i2.5776>

- [8] J. Brooke, "SUS: A 'Quickly and Dirty' Usability Scale," in *Usability Evaluation in Industry*, United Kingdom, Taylor & Francis, 1996, pp. 189-194.
- [9] Indrianto, "Performance Testing On Web Information System Using Apache Jmeter And Blazemeter," *Jurnal Ilmiah Ilmu Terapan Universitas Jambi*, vol. 7, no. 2, pp. 138–149, Dec. 2023, doi: <https://doi.org/10.22437/jiituj.v7i2.28440>.
- [10] A. B. Santosa, A. Syamsuddin, Sukirman, and Wardinata, "Manajemen Perubahan Penerimaan Peserta Didik Baru (PPDB)," *Seminar Nasional Hasil Pengabdian kepada Masyarakat*, pp. 1902–1908, Nov. 2022.
- [11] Widya Astuti Permana, "Manajemen Rekrutmen Peserta Didik Dalam Meningkatkan Mutu Lulusan ," *Jurnal Isema Islamic Educational Management*, vol. 5, no. 1, pp. 83–96, Jun. 2020. Doi: <https://doi.org/10.15575/isema.v5i1.5989>
- [12] Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, "*Peraturan Menteri Pendidikan Dasar dan Menengah Nomor 3 Tahun 2025*", 2025. [https://jdih.kemdikbud.go.id/sjdih/siperpu/dokumen/salinan/Permendikdasmen%20Nomor%203%20Tahun%202025%20\(JDIH\).pdf](https://jdih.kemdikbud.go.id/sjdih/siperpu/dokumen/salinan/Permendikdasmen%20Nomor%203%20Tahun%202025%20(JDIH).pdf). [Diakses: 16 April 2025]
- [13] K. C. Laudon and J. P. Laudon, *Management information systems : managing the digital firm*. 13th ed, London: Pearson Education, 2014.
- [14] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th ed. New York: McGraw-Hill, 2010.
- [15] Draw.io. "*UML 2.5 shape library with updated shapes*". Drawio.com, 2021. <https://www.drawio.com/blog/uml-2-5>. [Diakses: 21 April 2025]
- [16] S. Martina, S. Marion, H. Christian, and K. Gerti, *UML @ Classroom An Introduction to Object-Oriented Modeling*. Switzerland: Springer, 2015.
- [17] A. Cockburn, *Writing effective use cases*. Boston Addison-Wesley, 2001.

- [18] R. Lerdorf and K. Tatroe, Eds., *Programming PHP*, 1st ed. United State of America: O'Reilly, 2020.
- [19] M. Stauffer, *Laravel: Up and Running A Framework for Building Modern PHP Apps*, 2nd ed. United State of America: O'Reilly, 2019.
- [20] D. Colley, S. Stanier, and M. Asaduzzaman, "Investigating the Effects of Object-Relational Impedance Mismatch on the Efficiency of Object-Relational Mapping Frameworks," *Journal of Database Management*, vol. 31, no. 4, 2020. doi: 10.4018/JDM.2020100101
- [21] W. Komputer, *Panduan Belajar MySQL Database Server*, 1st ed. Jakarta: mediakita, 2010.
- [22] V. Riabov, "SMTP (Simple Mail Transfer Protocol)," in *Handbook of Computer Networks*, Hoboken, NJ, USA: John Wiley & Sons, 2007, pp. 388–406.
- [23] D. Lepilkina, "SMTP: The Protocol Behind Email Delivery – What Is It and How Does It Work," *Mailtrap Blog*, Jan. 1, 2025. [Online]. Available: <https://mailtrap.io/blog/smtp/#What-is-SMTP>. [Diakses: 15 Desember 2025]
- [24] A. Nata, M. Iqbal, dan N. Marpaung, "Workshop Pengenalan Framework CSS dalam Pembuatan Halaman Admin Aplikasi Web Menggunakan AdminLTE," *Community Development Journal*, vol. 4, no. 2, hal. 3032–3036, Jun. 2023. Doi: <https://doi.org/10.31004/cdj.v4i2.15018>
- [25] R. S. Pressman, *Software Engineering : A Practitioner's Approach*, 7th ed. New York: Mcgraw-Hill Higher Education, 2010.
- [26] M. Pomeroy-Huff, R. Cannon, T. A. Chick, J. Mullaney, and W. Nichols, "The Personal Software Process (PSP) Body of Knowledge, Version 2.0," *Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Special Report CMU/SEI-2009-SR-018*, Aug. 2009.
- [27] G. E. Iyawa, "*Personal Extreme Programming: Exploring Developers' Adoption*", in *Proceedings of the Americas Conference on Information Systems (AMCIS)*, bepress, 2020.

- [28] A. R. Amna and G. Poels, "Systematic Literature Mapping of User Story Research," *IEEE Access*, vol. 10, pp. 51723–51746, 2022, doi: 10.1109/ACCESS.2022.3173745.
- [29] V. Tawosi, R. Moussa, and F. Sarro, "On the Relationship Between Story Points and Development Effort in Agile Open-Source Software," in *Proceedings of the 16th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '22)*, Helsinki, Finland, 2022, pp. 183–194, doi: 10.1145/3544902.3546238.
- [30] R. Dhakad, "Requirement Prioritization Techniques in Agile Development Process: A Systematic Literature Review", Oct. 28, 2020. [Online]. Available: <https://doi.org/10.13140/RG.2.2.16140.39048>
- [31] D. A. Pratama, "Low-Fidelity vs High-Fidelity Prototyping," Dicoding Blog, [Online]. Tersedia: <https://www.dicoding.com/blog/low-fidelity-vs-high-fidelity-prototyping/>. [Diakses: 12 Mei 2025].
- [32] G. Alfarabi and P. B. Wintoro, "Pengembangan Front-End Web Pencarian Koleksi Buku Dinas Perpustakaan Dan Kearsipan Provinsi Lampung Menggunakan Framework Laravel Dan Bootstrap," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 1, Jan. 2025, doi: <https://doi.org/10.23960/jitet.v13i1.5523>.
- [33] Unit Laboratorium Fakultas Ilmu Terapan, "User Acceptance Test (UAT)," Laboratorium Fakultas Ilmu Terapan | Telkom University, 2023. tersedia: <https://fit.labs.telkomuniversity.ac.id/user-acceptance-test-uat/>. [Diakses: 10 Mei 2026].
- [34] Sony Ferbangkara et al., "Usability of Lampung Heritage Virtual Reality Tour," *Journal of Engineering and Scientific Research*, vol. 4, no. 2, Jan. 2023, doi: <https://doi.org/10.23960/jesr.v4i2.107>.
- [35] A. Wulansari, Asif Faruqi, and Tri, "Analisis Tingkat Usabilitas Website Perguruan Tinggi Menggunakan System Usability Scale (SUS)," *Jurnal Teknologi Informasi dan Terapan*, vol. 10, no. 2, pp. 126–131, Dec. 2023, doi: <https://doi.org/10.25047/jtit.v10i2.328>.

- [36] T. Vatsya, U. Sachin, K. G. Jayati, and A. Sanjiv, "Analytical Evaluation of Web Performance Testing Tools: Apache JMeter and SoapUI," in *Proc. 12th Int. Conf. on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, 2023. doi:10.1109/CSNT57126.2023.10134699
- [37] D. I. Permatasari, M. Ardani, A. Y. Ma'ulfa, N. Ilhami, S. G. Pratama, S. R. D. Astuti, and N. W. Naufalita, "Penguujian Aplikasi Menggunakan Metode Load Testing Dengan Apache JMeter Pada Sistem Informasi Pertanian," *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, vol. 8, no. 1, pp. 135–139, Jan. 2020. Doi: <https://doi.org/10.26418/justin.v8i1.34452>
- [38] A. Iskandar, Heri Retnawati, Haryanto, and Sahariani, "Design of a Web-Based Information System for New Student Registration in Vocational High Schools," *Ingénierie des systèmes d'information*, vol. 29, no. 04, pp. 1469–1481, Aug. 2024. Doi: <https://doi.org/10.18280/isi.290420>
- [39] Md. Tofail Ahmed, Md. Humaun Kabir, and S. Roy, "Web Based Student Registration and Exam Form Fill-up Management System for Educational Institute," *International Journal of Information Engineering and Electronic Business*, vol. 14, no. 2, pp. 47–62, Apr. 2022. Doi: [10.5815/ijieeb.2022.02.04](https://doi.org/10.5815/ijieeb.2022.02.04)
- [40] O. Suria, "A Statistical Analysis of System Usability Scale (SUS) Evaluations in Online Learning Platform," *Journal of Information Systems and Informatics*, vol. 6, no. 2, pp. 992–1007, Jun. 2024, doi: <https://doi.org/10.51519/journalisi.v6i2.750>.
- [41] P. Pahrizal, M. Muntahanah, Y. Reswan, and W. Seprianti, "The Use of Phpmailer in E-SPT System (Electronic Task Order) Parking Court at The Transportation Office Bengkulu City," *Jurnal Komputer, Informasi dan Teknologi*, vol. 2, no. 2, Dec. 2022, doi: <https://doi.org/10.53697/jkomitek.v2i2.881>.
- [42] T. P. Fiqar, N. F. Azhar, dan F. Azzahra, "Implementasi Metode Personal Extreme Programming dalam Pengembangan Sistem Informasi Manajemen

- Hak Paten pada Sentra HKI ITK," *Jurnal Specta*, vol. 6, no. 3, pp. 329–341, Jan. 2023. Doi: 10.35718/specta.v6i3.799
- [43] Yashfi Mahadi, Rinda Tiara, A. F. Abdurrohman, D. Rohpandi, and Feri Alpiyasin, "Design Of Student Register Application Using Laravel 8 Framework," *Acman Accounting and Management Journal*, vol. 4, no. 2, pp. 154–163, Jan. 2025. Doi: <https://doi.org/10.55208/7aj.v4i2.162>
- [44] O. López-Gorozabel, E. Cedeño-Palma, J. Pinargote-Ortega, W. Zambrano-Romero, and M. Pazmiño-Campuzano, "Bootstrap as a Tool for Web Development and Graphic Optimization on Mobile Devices," *Artificial Intelligence, Computer and Software Engineering Advances*, pp. 290–302, 2021, doi: [https://doi.org/10.1007/978-3-030-68080-0\\_22](https://doi.org/10.1007/978-3-030-68080-0_22).
- [45] Irvansyah Nuradi Susilo and Dwi Cahyono, "Perancangan Platform Digital Administrasi Desa Berbasis Web Melalui Pengujian Uml Di Balai Desa Warungering," *Jurnal Informatika Teknologi dan Sains (Jinteks)*, vol. 7, no. 2, pp. 995–1004, Jul. 2025, doi: <https://doi.org/10.51401/jinteks.v7i2.6039>.
- [46] None Rozanah and H. Budi, "Usability Evaluation and Alternative Design Recommendations for HSIS Mobile Application: A Case Study of PT HNI," *Indonesian Journal of Computer Science*, vol. 13, no. 4, Aug. 2024, doi: <https://doi.org/10.33022/ijcs.v13i4.4260>.
- [47] P. Vlachogianni and N. Tselios, "Perceived usability evaluation of educational technology using the System Usability Scale (SUS): A systematic review," *Journal of Research on Technology in Education*, vol. 54, no. 3, pp. 1–18, Feb. 2021, doi: <https://doi.org/10.1080/15391523.2020.1867938>.
- [48] S. Das, "How fast should a website load in 2023?," BrowserStack, Apr. 06, 2023. <https://www.browserstack.com/guide/how-fast-should-a-website-load>. [Diakses: 10 Oktober 2025]