



**SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI**

**RANCANG BANGUN SISTEM INFORMASI AKADEMIK  
UNTUK OPTIMALISASI DAN PENINGKATAN  
KOLABORASI ANTAR *DEVELOPER* MENGGUNAKAN  
*FRAMEWORK SCRUM***

**TUGAS AKHIR**

**Fadhlorrohman Zuhdi**

**0110220037**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**DEPOK**

**AGUSTUS 2024**



**STT TERPADU  
NURUL FIKRI**

**SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI**

**RANCANG BANGUN SISTEM INFORMASI AKADEMIK UNTUK  
OPTIMALISASI DAN PENINGKATAN KOLABORASI ANTAR  
DEVELOPER MENGGUNAKAN FRAMEWORK SCRUM**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana komputer**

**STT - NF**

**Fadhlorrohman Zuhdi**

**0110220037**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**DEPOK**

**AGUSTUS 2024**

**HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi/Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.**

**Nama : Fadhlurrohman Zuhdi**  
**NIM : 0110220037**

Depok, 13 Agustus 2024



Fadhlurrohman Zuhdi

**STT - NF**

## HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Fadhlurrohman Zuhdi

NIM : 0110220037

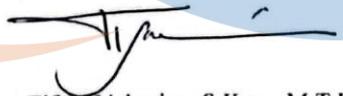
Program Studi : Teknik Informatika

Judul Skripsi : Rancang Bangun Sistem Informasi Akademik Untuk Optimalisasi Dan Peningkatan Kolaborasi Antar *Developer* Menggunakan *Framework Scrum*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

### DEWAN PENGUJI

Pembimbing



Tifani Nabarian, S.Kom, M.T.I

Penguji



Nasrul, S.Pd.I, S.Kom, M.Kom

Ditetapkan di : Depok

Tanggal : 29 Juli 2024

STT - NF

## KATA PENGANTAR

Dengan penuh rasa syukur dan hormat, penulis ingin mengucapkan puji syukur kepada Allah SWT, atas segala berkat dan rahmat-Nya yang telah melimpahkan kepada penulis dalam menyelesaikan tugas akhir ini. Penulisan tugas akhir ini merupakan bagian dari perjalanan menuju gelar Sarjana Komputer, Program Studi Teknik Informatika, di Sekolah Tinggi Teknologi Terpadu Nurul Fikri. Penulis sadar bahwa tanpa dukungan dan bimbingan dari berbagai pihak, dari masa perkuliahan hingga penyusunan tugas akhir ini, penyelesaian tugas akhir ini akan sangat sulit bagi penulis. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang tak terhingga kepada:

1. Allah SWT.
2. Orang tua dan semua anggota keluarga yang telah memberikan dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
3. Bapak Dr. Lukman Rosyidi selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Ibu Tifani Nabarian, S.Kom, M.T.I selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri juga sebagai Dosen Pembimbing Tugas Akhir.
5. Bapak Nasrul, S.Pd.I, S.Kom, M.Kom selaku Dosen Penguji Tugas Akhir
6. Bapak Zaki Imanuddin selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama berkuliah di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
7. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.
8. Keluarga kobon yang selalu menemani keseharian dan memberikan support untuk lulus tepat waktu.

Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila terdapat

kekurangan di dalam penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 13 Agustus 2024



Fadhlurrohman Zuhdi



STT - NF

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Fadhlurrohman Zuhdi

NIM : 0110220037

Program Studi : Teknik Informatika

Jenis karya : Skripsi / Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*)** atas karya ilmiah saya yang berjudul :

Rancang Bangun Sistem Informasi Akademik Untuk Optimalisasi Dan Peningkatan Kolaborasi Antar *Developer* Menggunakan *Framework Scrum*

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 13 Agustus 2024

Yang Menyatakan



(Fadhlurrohman Zuhdi)

## ABSTRAK

Nama : Fadhlurrohman Zuhdi  
NIM : 0110220037  
Program Studi : Teknik Informatika  
Judul : Rancang Bangun Sistem Informasi Akademik Untuk Optimalisasi Dan Peningkatan Kolaborasi Antar *Developer* Menggunakan *Framework Scrum*

Kemajuan teknologi mempermudah pengelolaan data sekolah, namun tantangan seperti optimasi sistem dan kolaborasi tim perlu diatasi. Penelitian ini bertujuan untuk mengembangkan sistem informasi akademik di SMK Bina Husada Mandiri dengan menggunakan *framework Laravel* dan metode pengembangan *Scrum*, serta melakukan pengujian agar sesuai dengan kebutuhan pengguna. Metode yang diterapkan adalah *research and development (R&D)* yang melibatkan desain, implementasi, dan evaluasi sistem. Dalam pengembangan, *framework Laravel* dipilih karena fitur-fitur unggulannya, seperti *Eloquent ORM* dan struktur *MVC*, yang meningkatkan efisiensi. Metode *Scrum* digunakan untuk memperbaiki transparansi dan kolaborasi dalam tim. Hasil evaluasi menunjukkan bahwa sistem yang dikembangkan memenuhi kebutuhan pengguna dengan tingkat keberhasilan 100% berdasarkan uji *blackbox*. *Laravel* terbukti lebih efisien dan berkualitas dalam kode dibandingkan *PHP native*, meskipun *PHP native* menawarkan kecepatan *web* lebih baik. Penelitian ini mengkonfirmasi bahwa penggunaan *Laravel* dan *Scrum* berhasil meningkatkan kualitas sistem informasi akademik di SMK Bina Husada Mandiri.

Kata kunci : *Laravel*, *PHP Native*, *Research and development*, *Scrum*, Sistem informasi akademik

## **ABSTRACT**

*Name : Fadhlorrohman Zuhdi*  
*NIM : 0110220037*  
*Study Program : Informatics*  
*Title : Design and Development of Academic Information System for Optimization and Improved Developer Collaboration Using Scrum Framework*

*Technological advancements have simplified school data management, but challenges such as system optimization and team collaboration need to be addressed. This study aims to develop an academic information system for SMK Bina Husada Mandiri using the Laravel framework and the Scrum development method, and to conduct testing to ensure it meets user needs. The applied method is research and development (R&D), which involves system design, implementation, and evaluation. Laravel was chosen for its superior features, such as Eloquent ORM and MVC structure, which enhance efficiency. The Scrum method was employed to improve transparency and collaboration within the team. Evaluation results indicate that the developed system meets user needs with a 100% success rate based on black-box testing. Laravel has proven to be more efficient and higher in code quality compared to native PHP, although native PHP offers better web speed. This research confirms that using Laravel and Scrum successfully improved the quality of the academic information system at SMK Bina Husada Mandiri.*

*Key words : Academic information system, Laravel, PHP Native, Research and development, Scrum*

## DAFTAR ISI

|  |      |
|--|------|
| HALAMAN PERNYATAAN ORISINALITAS.....   | iii  |
| HALAMAN PENGESAHAN.....  | iv   |
| KATA PENGANTAR .....   | v    |
| HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR<br>UNTUK KEPENTINGAN AKADEMIS ..... | vii  |
| ABSTRAK.....   | viii |
| <i>ABSTRACT</i> .....  | ix   |
| DAFTAR ISI.....  | x    |
| DAFTAR GAMBAR .....  | xiv  |
| DAFTAR TABEL.....  | xvi  |
| BAB I PENDAHULUAN .....  | 17   |
| 1.1 Latar belakang .....   | 17   |
| 1.2 Rumusan Masalah .....  | 18   |
| 1.3 Tujuan dan Manfaat Penelitian.....   | 18   |
| 1.4 Batasan Masalah.....   | 19   |
| 1.5 Sistematika Penulisan.....   | 19   |
| BAB II KAJIAN LITERATUR.....   | 21   |
| 2.1 <i>Framework</i> .....   | 21   |
| 2.2 <i>Laravel</i> .....   | 21   |
| 2.3 Sistem Informasi Akademik .....  | 22   |
| 2.4 <i>Tailwind CSS</i> .....  | 23   |
| 2.5 <i>Scrum</i> .....   | 23   |
| 2.6 <i>Research and Development(R&amp;D)</i> .....                                       | 24   |

|   |   |    |
|---|---|----|
| 2.7   | <i>Black Box Testing</i> .....                    | 25 |
| 2.8   | RDBMS .....                                       | 25 |
| 2.9   | <i>MariaDB</i> .....                              | 26 |
| 2.10  | Penelitian Terkait .....                          | 27 |
| 2.11  | Posisi Penelitian .....                           | 28 |
| <b>BAB III METODOLOGI PENELITIAN</b> .....    |   | 30 |
| 3.1   | Tahapan Penelitian .....                          | 30 |
| 3.1.1   | Studi Pendahuluan( <i>Sprint Planning</i> ) ..... | 31 |
| 3.1.2   | Analisis( <i>Sprint Planning</i> ).....           | 31 |
| 3.1.3   | Perancangan Sistem( <i>Design</i> ) .....         | 31 |
| 3.1.4   | Implementasi( <i>Daily Sprint</i> ).....          | 31 |
| 3.1.5   | <i>Testing(Daily Sprint)</i> .....                | 31 |
| 3.1.6   | Evaluasi( <i>Daily Sprint</i> ).....              | 32 |
| 3.2   | Rancangan Penelitian .....                        | 32 |
| 3.2.1   | Jenis Penelitian.....                             | 32 |
| 3.2.2   | Metode Analisis Data.....                         | 33 |
| 3.2.3   | Metode Pengumpulan Data.....                      | 33 |
| 3.2.4   | Metode Pengujian.....                             | 33 |
| 3.2.5   | Metode Implementasi dan Evaluasi .....            | 35 |
| 3.2.6   | Lingkungan Pengembangan .....                     | 36 |
| <b>BAB IV IMPLEMENTASI DAN EVALUASI</b> ..... |   | 37 |
| 4.1   | Analisis .....                                    | 37 |
| 4.1.1   | Masalah, Analisis dan Solusi .....                | 37 |
| 4.1.2   | <i>User Story</i> .....                           | 38 |
| 4.1.3   | <i>Product Backlog</i> .....                      | 39 |

|                                 |  |    |
|---------------------------------|--|----|
| 4.1.4                           | <i>Fase Sprint</i> .....                     | 40 |
| 4.2                             | Perancangan.....                             | 41 |
| 4.2.1                           | <i>Use Case Diagram</i> .....                | 41 |
| 4.2.2                           | <i>Class Diagram</i> .....                   | 42 |
| 4.2.3                           | <i>Activity Diagram</i> .....                | 44 |
| 4.3                             | Implementasi.....                            | 46 |
| 4.3.1                           | Halaman <i>Login</i> .....                   | 46 |
| 4.3.2                           | <i>Dashboard</i> .....                       | 47 |
| 4.3.3                           | Halaman Manajemen Pegawai.....               | 47 |
| 4.3.4                           | Halaman Manajemen Siswa.....                 | 49 |
| 4.3.5                           | Halaman Manajemen Kelas.....                 | 51 |
| 4.3.6                           | Halaman Manajemen Ruangan.....               | 53 |
| 4.3.7                           | Halaman Manajemen Mata Pelajaran.....        | 55 |
| 4.3.8                           | Halaman Manajemen Tahun Ajaran.....          | 57 |
| 4.3.9                           | Halaman Manajemen Jabatan.....               | 59 |
| 4.3.10                          | Halaman Manajemen Unit.....                  | 61 |
| 4.4                             | Pengujian.....                               | 63 |
| 4.4.1                           | Fokus Pengujian.....                         | 64 |
| 4.4.2                           | Hasil Pengujian <i>Blackbox</i> .....        | 65 |
| 4.4.3                           | Hasil Pengujian <i>Line Code</i> .....       | 69 |
| 4.4.4                           | Hasil Pengujian Efisiensi <i>Code</i> .....  | 71 |
| 4.4.5                           | Hasil Pengujian Struktur <i>Folder</i> ..... | 72 |
| 4.4.6                           | Hasil Pengujian Kecepatan <i>Web</i> .....   | 74 |
| 4.4.7                           | Evaluasi Pengujian Perbandingan.....         | 75 |
| BAB V KESIMPULAN DAN SARAN..... |  | 77 |

|                      |    |
|----------------------|----|
| 5.1 Kesimpulan.....  | 77 |
| 5.2 Saran.....       | 77 |
| DAFTAR PUSTAKA ..... | 79 |
| LAMPIRAN.....        | 81 |

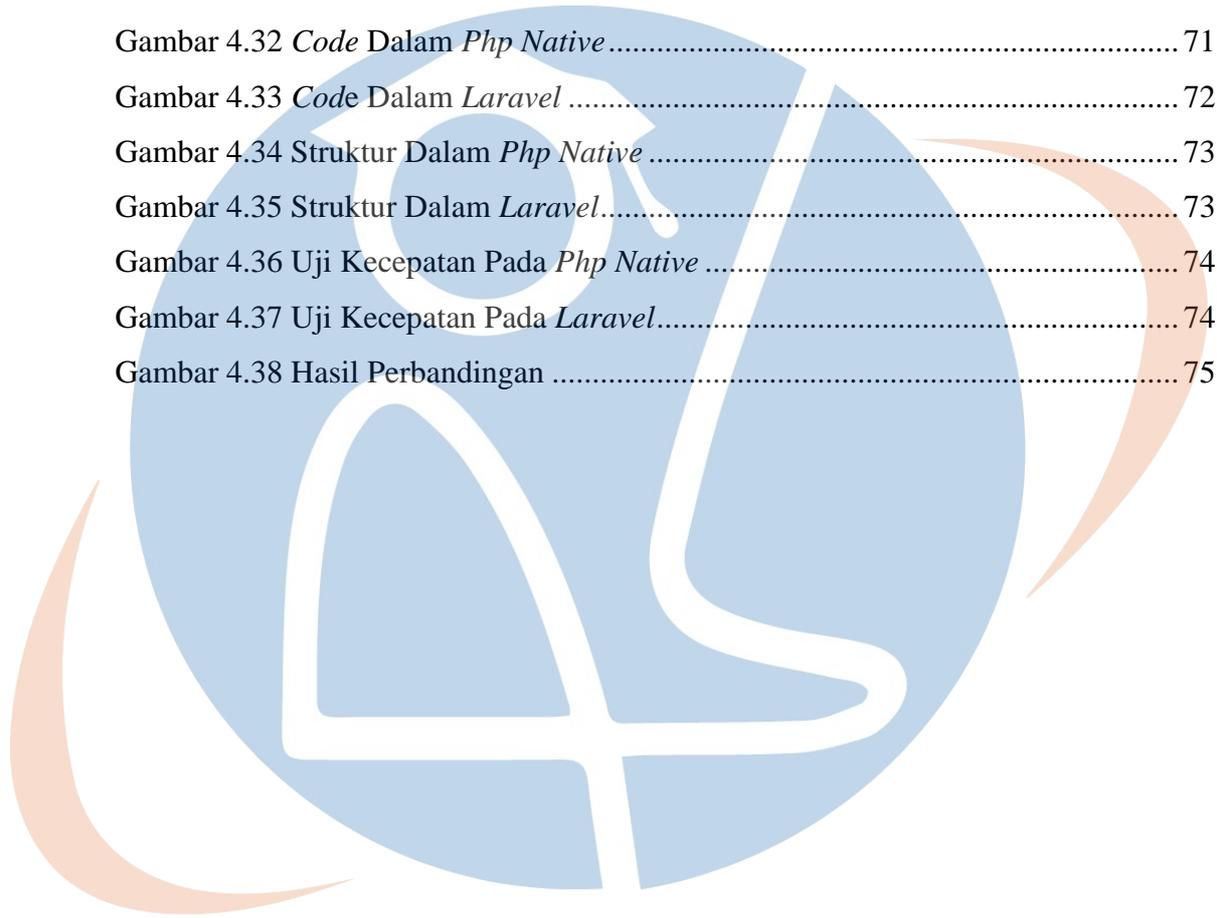


STT - NF

## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1 <i>Alur Framework Scrum</i> [14] .....         | 24 |
| Gambar 2.2 <i>Black Box Testing</i> [15] .....            | 25 |
| Gambar 3.1 Tahapan Penelitian .....                       | 30 |
| Gambar 4.1 <i>Usecase Diagram</i> .....                   | 42 |
| Gambar 4.2 <i>Class Diagram</i> .....                     | 43 |
| Gambar 4.3 <i>Activity Diagram</i> .....                  | 45 |
| Gambar 4.4 Halaman <i>Login</i> .....                     | 46 |
| Gambar 4.5 Halaman <i>Dashboard</i> .....                 | 47 |
| Gambar 4.6 Halaman Data Pegawai .....                     | 48 |
| Gambar 4.7 Halaman <i>Form</i> Tambah Pegawai.....        | 48 |
| Gambar 4.8 Halaman <i>Form</i> Ubah Pegawai .....         | 49 |
| Gambar 4.9 Halaman Data Siswa .....                       | 50 |
| Gambar 4.10 Halaman <i>Form</i> Tambah Siswa.....         | 50 |
| Gambar 4.11 Halaman <i>Form</i> Ubah Siswa .....          | 51 |
| Gambar 4.12 Halaman Data Kelas.....                       | 52 |
| Gambar 4.13 Halaman <i>Form</i> Tambah Kelas .....        | 52 |
| Gambar 4.14 Halaman <i>Form</i> Ubah Kelas.....           | 53 |
| Gambar 4.15 Halaman Data Ruangan.....                     | 54 |
| Gambar 4.16 Halaman <i>Form</i> Tambah Ruangan .....      | 54 |
| Gambar 4.17 Halaman <i>Form</i> Ubah Ruangan.....         | 55 |
| Gambar 4.18 Halaman Data Mapel.....                       | 56 |
| Gambar 4.19 Halaman <i>Form</i> Tambah Mapel .....        | 56 |
| Gambar 4.20 Halaman <i>Form</i> Ubah Mapel.....           | 57 |
| Gambar 4.21 Halaman Data Tahun Ajaran.....                | 58 |
| Gambar 4.22 Halaman <i>Form</i> Tambah Tahun Ajaran ..... | 58 |
| Gambar 4.23 Halaman <i>Form</i> Ubah Tahun Ajaran.....    | 59 |
| Gambar 4.24 Halaman Data Jabatan.....                     | 60 |
| Gambar 4.25 Halaman <i>Form</i> Tambah Jabatan .....      | 60 |
| Gambar 4.26 Halaman <i>Form</i> Ubah Jabatan.....         | 61 |

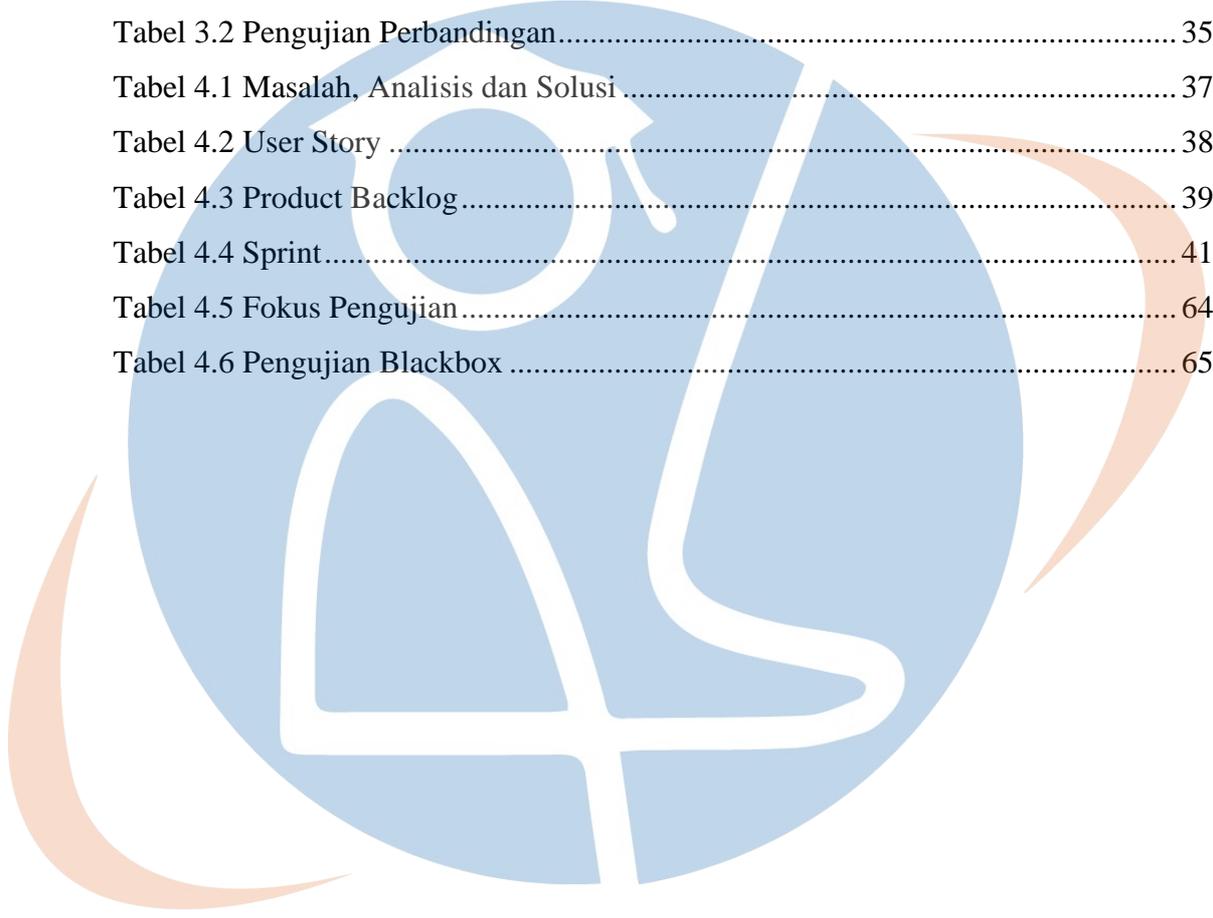
|  |    |
|--|----|
| Gambar 4.27 Halaman Data Unit.....                         | 62 |
| Gambar 4.28 Halaman <i>Form</i> Tambah Unit .....          | 62 |
| Gambar 4.29 Halaman <i>Form</i> Ubah Unit.....             | 63 |
| Gambar 4.30 <i>Line Code</i> Dalam <i>Php Native</i> ..... | 70 |
| Gambar 4.31 <i>Line Code</i> Dalam <i>Laravel</i> .....    | 70 |
| Gambar 4.32 <i>Code</i> Dalam <i>Php Native</i> .....      | 71 |
| Gambar 4.33 <i>Code</i> Dalam <i>Laravel</i> .....         | 72 |
| Gambar 4.34 Struktur Dalam <i>Php Native</i> .....         | 73 |
| Gambar 4.35 Struktur Dalam <i>Laravel</i> .....            | 73 |
| Gambar 4.36 Uji Kecepatan Pada <i>Php Native</i> .....     | 74 |
| Gambar 4.37 Uji Kecepatan Pada <i>Laravel</i> .....        | 74 |
| Gambar 4.38 Hasil Perbandingan .....                       | 75 |



STT - NF

## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 2.1 Tahapan Penelitian.....           | 27 |
| Tabel 2.2 Posisi Penelitian.....            | 28 |
| Tabel 3.1 Blackbox Testing.....             | 34 |
| Tabel 3.2 Pengujian Perbandingan.....       | 35 |
| Tabel 4.1 Masalah, Analisis dan Solusi..... | 37 |
| Tabel 4.2 User Story.....                   | 38 |
| Tabel 4.3 Product Backlog.....              | 39 |
| Tabel 4.4 Sprint.....                       | 41 |
| Tabel 4.5 Fokus Pengujian.....              | 64 |
| Tabel 4.6 Pengujian Blackbox.....           | 65 |



STT - NF

## BAB I PENDAHULUAN

Bab ini merupakan tahapan awal penelitian yang menjelaskan latar belakang, rumusan masalah, tujuan dan manfaat penelitian, serta menentukan batasan masalah agar dalam pembahasan penelitian tidak meluas.

### 1.1 Latar belakang

Perkembangan teknologi telah membawa kemudahan dalam pengelolaan data sekolah melalui penyediaan satu *platform* sistem informasi akademik yang mencakup *input* data, hasil, dan rekapitulasi[1]. Namun, di balik kemudahan tersebut, terdapat beberapa tantangan yang perlu diperhatikan, seperti optimasi sistem dan kolaborasi antara *developer*. Kedua aspek ini merupakan faktor penting dalam pengembangan sistem informasi akademik, yang jika diabaikan dapat berdampak pada kinerja sistem dan kesulitan dalam perawatan serta pengembangan masa depan.

SMK Bina Husada Mandiri merupakan sebuah institusi pendidikan menengah kejuruan yang memiliki komitmen yang kuat terhadap peningkatan mutu sistem informasi akademik. Setelah melakukan observasi dan diskusi dengan staf *developer* sistem informasi akademik, ditemukan bahwa sistem yang sedang digunakan saat ini dibangun menggunakan *PHP Native*. Meskipun sistem ini telah berjalan dengan baik, penggunaan *PHP Native* menimbulkan beberapa kekurangan yang tidak bisa diabaikan, seperti kurangnya struktur dan standar pengembangan yang hanya dipahami oleh *developer* aslinya, serta efisiensi kode yang kurang optimal[2].

Sebagai solusi, penulis akan melakukan perancangan ulang sistem informasi akademik menggunakan *framework* PHP, khususnya *Laravel*. Pemilihan *Laravel* sebagai *framework* dikarenakan keunggulan yang dimilikinya, salah satunya adalah kejelasan dan ketertarikan pada arsitektur yang ditawarkan. Hal ini karena *developer* dapat dengan mudah menemukan dokumentasi resmi *Laravel* di situs *web* resmi mereka[3]. Oleh karena itu, penggunaan *Laravel* diharapkan tidak hanya

meningkatkan aspek arsitektur perangkat lunak, tetapi juga memperbaiki kelemahan yang mungkin terdapat dalam penggunaan *PHP Native*.

Manajemen penelitian akan mengadopsi metode *Agile*, dengan fokus pada kerangka kerja *Scrum*, untuk mengelola proyek pengembangan sistem informasi sekolah. Diharapkan, pendekatan ini akan meningkatkan transparansi, kolaborasi, dan responsivitas, serta meningkatkan kepuasan pemangku kepentingan dalam pengembangan sistem informasi akademik. Peran *product owner* akan diemban oleh staf *developer* sistem informasi akademik SMK Bina Husada Mandiri, sementara peran *scrum master* dan tim *developer* akan diisi oleh penulis.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah pada penelitian ini yaitu:

1. Bagaimana rancang bangun sistem informasi akademik SMK Bina Husada Mandiri dengan *framework Laravel* dan metode pengembangan *Scrum*?
2. Apakah sistem informasi akademik yang dikembangkan sudah sesuai dengan kebutuhan pengguna?

## **1.3 Tujuan dan Manfaat Penelitian**

Tujuan dan manfaat dari penelitian ini atas dasar rumusan masalah di atas adalah sebagai berikut.

### **A. Tujuan Penelitian**

1. Membangun sistem informasi akademik SMK Bina Husada Mandiri menggunakan *framework Laravel* dengan menerapkan metode pengembangan *framework Scrum*.
2. Melakukan pengujian untuk memastikan bahwa sistem informasi akademik yang dikembangkan sesuai dengan kebutuhan pengguna.

### **B. Manfaat Penelitian**

1. Meningkatkan efisiensi operasional SMK Bina Husada Mandiri melalui sistem informasi akademik yang lebih canggih, aman, dan mudah diadaptasi dengan kebutuhan pendidikan yang berubah-ubah.

2. Menyediakan contoh nyata dari manfaat migrasi ke *framework* yang lebih modern, yang dapat dijadikan panduan untuk pengembangan sistem informasi lainnya.
3. Membantu mengatasi beberapa kesulitan yang di hadapi oleh *developer* lama ataupun *developer* baru di sekolah tersebut.

#### **1.4 Batasan Masalah**

Untuk memastikan fokus dan kejelasan dalam penelitian serta pengembangan sistem, batasan masalah yang ditetapkan adalah meliputi pengelolaan data yang terbatas pada informasi siswa, pegawai, mata pelajaran, kelas, ruangan, dan unit dalam sistem informasi akademik.

#### **1.5 Sistematika Penulisan**

Dalam proses penulisan tugas akhir, penulis berinisiatif membuat sistematika penulisan agar pembahasan dalam penelitian ini dapat terfokus sampai akhir sesuai tujuan, sebagai berikut.

### **BAB I PENDAHULUAN**

Bab ini merupakan tahap awal bagi pembaca untuk memahami latar belakang, permasalahan yang dihadapi, tujuan penelitian, serta batasan-batasan yang relevan. Pendahuluan ini membangun fondasi yang kuat untuk penelitian yang akan dilakukan.

### **BAB II KAJIAN LITERATUR**

Bab ini menyajikan tinjauan mendalam mengenai teori, penelitian, dan kajian yang relevan terhadap topik yang diselidiki. Penulis memaparkan sudut pandang yang telah diambil oleh para ahli sebelumnya untuk memberikan landasan yang kokoh bagi penelitian ini.

### **BAB III METODE PENELITIAN**

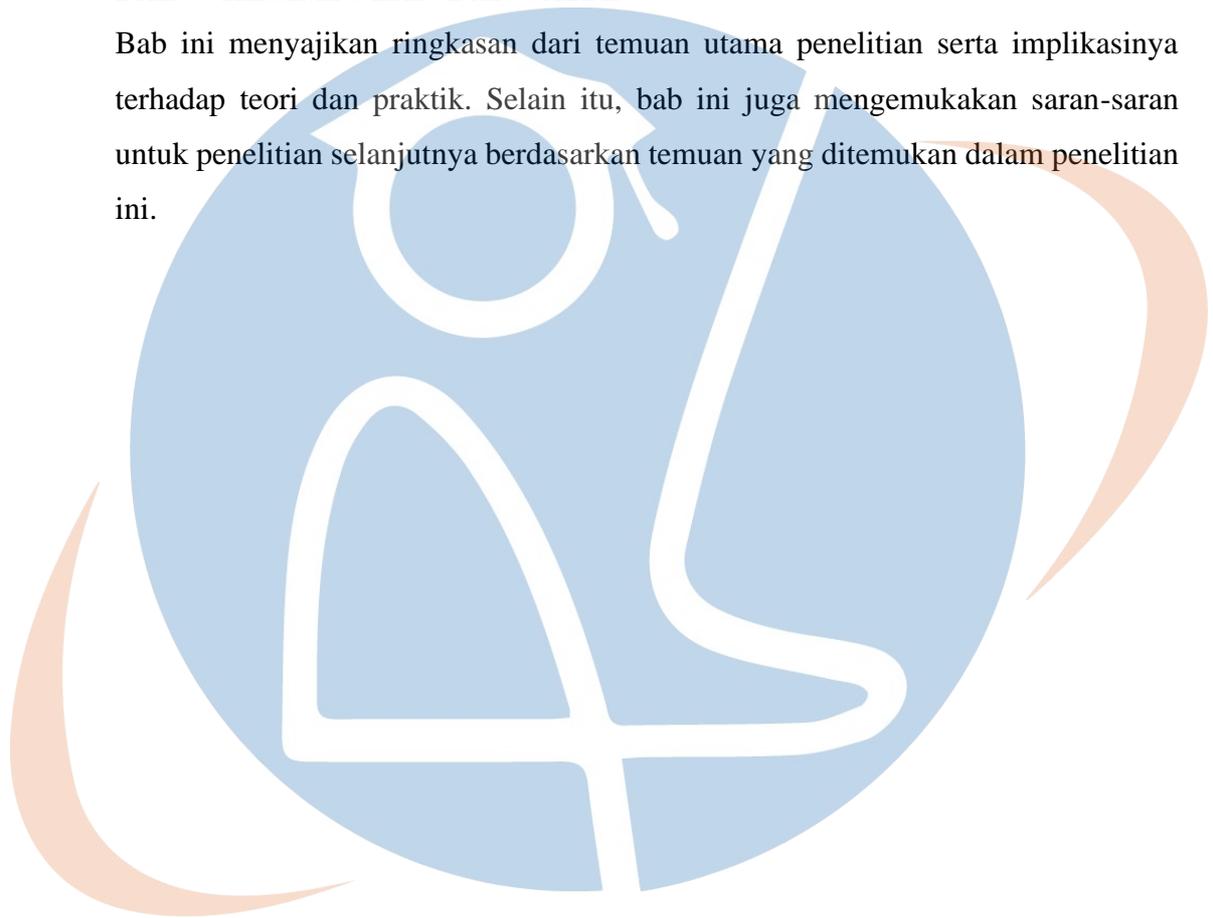
Bab ini menjelaskan secara rinci langkah-langkah yang diambil dalam pelaksanaan penelitian. Penjelasan yang cermat dalam bab ini memberikan gambaran yang jelas tentang bagaimana penelitian dilakukan.

#### **BAB IV IMPLEMENTASI DAN HASIL**

Bab ini menyajikan hasil dari penelitian serta implementasi dari temuan yang diperoleh. Penulis memaparkan temuan-temuan utama beserta interpretasinya, serta menghubungkannya dengan teori yang telah dipelajari dalam bab kajian literatur.

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

Bab ini menyajikan ringkasan dari temuan utama penelitian serta implikasinya terhadap teori dan praktik. Selain itu, bab ini juga mengemukakan saran-saran untuk penelitian selanjutnya berdasarkan temuan yang ditemukan dalam penelitian ini.



**STT - NF**

## BAB II

### KAJIAN LITERATUR

Sebelum melakukan penelitian, penting untuk melakukan kajian literatur guna memahami landasan teori dan temuan yang relevan dengan topik penelitian.

#### 2.1 *Framework*

*Framework* adalah kerangka kerja atau struktur konseptual yang digunakan untuk memfasilitasi pembangunan dan pengembangan aplikasi atau sistem. Ini menyediakan serangkaian alat, aturan, dan konvensi yang memudahkan *developer* dalam membangun sesuatu dengan cepat dan efisien. Dalam konteks pengembangan perangkat lunak, *framework* sering kali berisi perpustakaan kode yang telah dibangun sebelumnya, modul-modul yang dapat digunakan kembali. Membangun suatu aplikasi tanpa menggunakan *framework*, seperti membangun sebuah rumah tanpa menggunakan kerangka yang sudah ada. Seorang *developer* aplikasi bisa saja memulai dari awal dengan membangun kerangka sendiri, tetapi hal itu akan memakan banyak waktu dan usaha. Sebaliknya dengan menggunakan *framework*, *developer* aplikasi bisa mendapatkan kerangka yang sudah ada, sehingga dapat fokus pada penambahan fungsionalitas yang lebih tinggi ke dalam aplikasi[4].

*Framework* memiliki berbagai jenis yang dapat digunakan sesuai kebutuhan pengembangan, seperti untuk membangun *single-page web*, *static web*, *dynamic web*, dan lain sebagainya. Sebagai contoh, jika Anda ingin membuat sistem yang berbentuk *website* menggunakan bahasa pemrograman PHP, Anda bisa menggunakan *framework Laravel*. Dengan menggunakan *framework* yang sesuai, *developer* dapat mempercepat proses pengembangan aplikasi sesuai dengan kebutuhan spesifik yang diinginkan[5].

#### 2.2 *Laravel*

*Laravel* merupakan sebuah *framework PHP* yang dirancang dengan tujuan untuk menyederhanakan serta mempercepat proses pengembangan aplikasi *web*.

Keberadaan *Laravel* menjadi penting dalam pengembangan *web* karena menyajikan sejumlah fitur dan alat bantu yang handal. Salah satu keunggulan utama yang dimiliki oleh *Laravel* adalah sistem *routing* yang efisien, yang memudahkan pengembang dalam manajemen rute URL aplikasi dengan baik. Selain itu, *Laravel* juga dilengkapi dengan ORM (*Object-Relational Mapping*) yang kuat melalui *Eloquent*, memudahkan pengembang dalam berinteraksi dengan basis data. Fitur-fitur seperti otentikasi pengguna, validasi formulir, serta alat bantu lainnya turut mendukung pengembangan aplikasi dengan efektif.

Selain fitur teknis yang unggul, kelebihan lain dari *Laravel* adalah dokumentasi yang lengkap dan mudah dipahami, serta komunitas yang aktif dan besar. Faktor-faktor ini memberikan kemudahan bagi pengembang dalam mempelajari dan mengembangkan aplikasi menggunakan *Laravel*. Karena itu, *Laravel* telah menjadi pilihan yang populer dalam pengembangan aplikasi *web* yang modern dan kompleks[6].

### **2.3 Sistem Informasi Akademik**

Sistem Informasi Akademik merupakan sebuah sistem yang dirancang khusus untuk mengelola informasi dan proses-proses terkait di lingkungan pendidikan, terutama di institusi pendidikan seperti sekolah. Sistem ini bertujuan untuk menyediakan alat yang diperlukan bagi manajemen pendidikan untuk mengorganisasi data, mengelola proses akademik, serta memfasilitasi pengambilan keputusan yang berbasis data. Dalam konteks pendidikan, sistem informasi akademik memainkan peran penting dalam mengintegrasikan berbagai aspek, seperti informasi siswa, jadwal pelajaran, kehadiran, nilai, dan pembayaran, guna mencapai tujuan pendidikan yang ditetapkan[7].

Sistem informasi akademik juga memberikan dukungan dalam memfasilitasi interaksi antara berbagai pemangku kepentingan di lingkungan pendidikan. Dengan menyediakan akses yang mudah dan terstruktur terhadap informasi yang relevan, sistem ini membantu meningkatkan efisiensi, transparansi, dan akuntabilitas dalam pengelolaan dan penyampaian layanan pendidikan.

## 2.4 *Tailwind* CSS

*Tailwind* CSS adalah sebuah *framework* CSS yang mengadopsi pendekatan *utility-first*, yang memungkinkan *developer* untuk membangun antarmuka pengguna dengan cepat dan efisien. *Framework* ini menawarkan kumpulan kelas CSS bawaan yang dapat digunakan secara langsung dalam *markup* HTML, menghilangkan kebutuhan untuk menulis CSS khusus untuk setiap elemen. Dengan *Tailwind* CSS, *developer* sistem informasi akademik dapat dengan mudah mengatur tata letak, warna, tipografi, dan komponen antarmuka lainnya dengan cepat dan konsisten.

Salah satu keunggulan utama dari *Tailwind* CSS adalah fleksibilitasnya yang tinggi dalam menyesuaikan gaya dan tampilan antarmuka pengguna. Dengan kelas-kelas bawaan yang komprehensif, *developer* dapat dengan mudah menyesuaikan gaya sesuai dengan kebutuhan dan preferensi desain mereka tanpa harus menulis CSS kustom. Selain itu, *Tailwind* CSS juga menyediakan alat bantu yang kuat untuk mengelola responsivitas, sehingga antarmuka sistem informasi akademik dapat diakses dengan baik di berbagai perangkat dan ukuran layar[8].

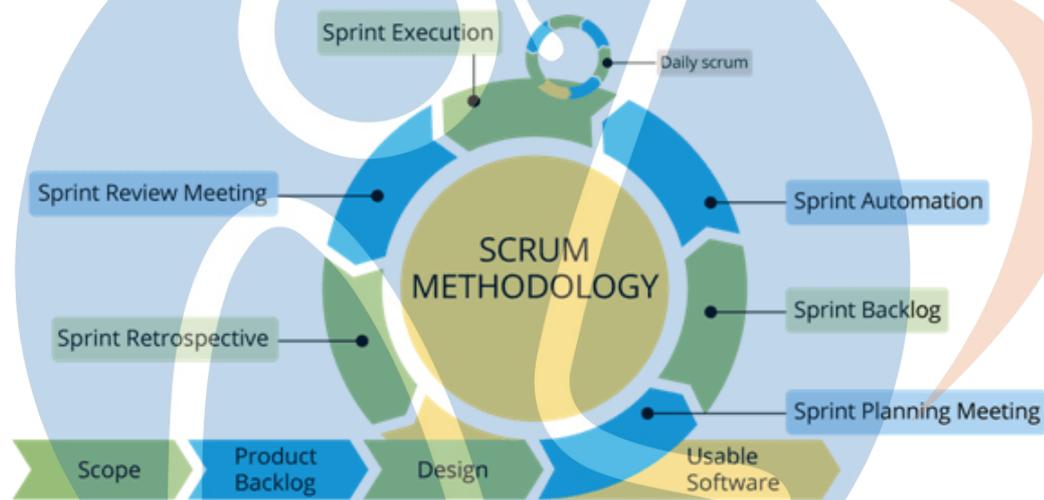
## 2.5 *Scrum*

*Scrum* adalah sebuah kerangka kerja yang digunakan dalam penerapan metodologi *Agile* untuk pengembangan sistem informasi akademik. Dengan menggunakan pendekatan ini, tim *developer* dapat bekerja secara kolaboratif dalam *sprints* (periode waktu terbatas), dengan fokus pada pengembangan iteratif dan penyelesaian prioritas berdasarkan kebutuhan pengguna. *Scrum* menekankan pada transparansi, inspeksi, dan adaptasi, yang memungkinkan tim untuk terus memperbaiki dan menyesuaikan proses pengembangan sesuai dengan perubahan kebutuhan dan kondisi yang muncul.

Salah satu keunggulan utama dari *Scrum* adalah kemampuannya untuk memfasilitasi komunikasi yang efektif dan kolaborasi antara anggota tim. Dengan rutinitas pertemuan harian, *review sprint*, dan retrospektif, *Scrum* memungkinkan tim untuk terus berkomunikasi, membagikan pembaruan, dan menyelesaikan masalah dengan cepat. Selain itu, kerangka kerja ini juga memungkinkan untuk

pengelolaan risiko yang lebih baik dan pengambilan keputusan yang lebih cepat, sehingga memungkinkan *developer* sistem informasi akademik untuk menghadapi tantangan dengan lebih responsif dan efisien[9].

Dalam implementasi *Scrum*, terdapat sejumlah tahapan yang harus dilalui oleh tim, termasuk perencanaan *sprint*, pembuatan *backlog* produk, melakukan *sprint*, melakukan pertemuan harian, melakukan *review sprint*, melakukan retrospektif, dan meninjau dan memperbarui *backlog* produk.



Gambar 2.1 Alur Framework Scrum[14]

## 2.6 *Research and Development(R&D)*

Metode penelitian *Research and Development (R&D)* merupakan pendekatan yang digunakan untuk menghasilkan inovasi dan pengembangan baru dalam berbagai bidang, termasuk teknologi, ilmu sosial, dan pendidikan. Dalam konteks industri, R&D berfokus pada eksplorasi, penemuan, dan implementasi solusi baru yang dapat meningkatkan efisiensi, kualitas, dan daya saing perusahaan. Proses R&D melibatkan langkah-langkah seperti identifikasi masalah, pengumpulan dan analisis data, perencanaan desain, pengembangan prototipe, uji coba, dan evaluasi. Tujuan utama dari metode R&D adalah untuk menciptakan produk, layanan, atau proses baru yang dapat memberikan nilai tambah serta menjawab tantangan yang dihadapi oleh organisasi atau masyarakat secara luas. Melalui pendekatan ini,

inovasi dapat terus berkembang dan memberikan kontribusi positif terhadap kemajuan dan perkembangan suatu bidang atau industri[10].

## 2.7 Black Box Testing

Metode pengujian *black box testing* merupakan pendekatan yang difokuskan pada pengujian fungsionalitas suatu perangkat lunak tanpa memperhatikan struktur internal atau kode programnya. Dalam *black box testing*, pengujian dilakukan dengan menguji *input* dan *output* dari perangkat lunak tanpa memperhatikan bagaimana proses internalnya bekerja. Pendekatan ini mirip dengan cara pengguna akhir berinteraksi dengan perangkat lunak, sehingga memungkinkan untuk mengevaluasi apakah perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditetapkan. Metode ini melibatkan berbagai teknik pengujian, seperti pengujian ekuitas kelas yang setara, pengujian batas, pengujian aliran kontrol, dan lainnya. Dengan menggunakan *black box testing*, pengembang perangkat lunak dapat menemukan *bug* atau kelemahan dalam fungsionalitas perangkat lunak yang mungkin tidak terdeteksi dengan metode pengujian lainnya, sehingga membantu memastikan kualitas dan kehandalan perangkat lunak sebelum dirilis ke pengguna akhir[11].



Gambar 2.2 Black Box Testing[15]

## 2.8 RDBMS

Sistem Manajemen Basis Data Relasional (RDBMS) adalah sebuah jenis sistem manajemen basis data yang didesain untuk menyimpan dan mengelola data dalam bentuk tabel yang saling terhubung satu sama lain melalui relasi atau kunci

asing. Dalam RDBMS, data disimpan dalam bentuk tabel yang terdiri dari baris dan kolom, dengan setiap baris mewakili satu rekaman atau entitas, dan setiap kolom mewakili atribut atau karakteristik dari entitas tersebut. Sistem ini menggunakan bahasa kueri SQL (*Structured Query Language*) untuk melakukan manipulasi data, seperti penyimpanan, pengambilan, pembaruan, dan penghapusan data. Salah satu karakteristik utama dari RDBMS adalah kemampuannya untuk menjaga integritas data dengan menerapkan aturan referensial antar tabel dan konsistensi data. RDBMS juga menawarkan fleksibilitas dalam mengakses dan mengelola data, serta memungkinkan untuk melakukan berbagai operasi analisis dan pemrosesan data secara efisien. Dengan fitur-fitur tersebut, RDBMS menjadi pilihan yang umum digunakan dalam berbagai aplikasi dan industri untuk menyimpan, mengelola, dan menganalisis data dengan baik[12].

## 2.9 *MariaDB*

*MariaDB* adalah sistem manajemen basis data relasional (RDBMS) *open-source* yang dikembangkan sebagai respons terhadap akuisisi *MySQL* oleh Oracle pada tahun 2010, yang menimbulkan kekhawatiran di kalangan pengguna. Dibuat oleh tim yang dipimpin oleh Michael "Monty" Widenius, salah satu pendiri *MySQL*, *MariaDB* bertujuan untuk menjadi alternatif yang bebas dan terbuka bagi *MySQL*. Sistem ini dibangun dari kode sumber yang sama dengan *MySQL* namun telah mengalami peningkatan fitur dan performa. Dengan tetap menjaga kompatibilitas penuh, *MariaDB* memungkinkan pengguna untuk beralih tanpa perlu melakukan perubahan besar pada aplikasi dan *query* SQL yang telah ada. Seiring berjalannya waktu, *MariaDB* telah diadopsi secara luas oleh pengembang dan perusahaan yang mencari RDBMS yang andal dan aman, dengan fitur-fitur seperti replikasi, optimisasi *query*, keamanan data, dan dukungan untuk berbagai *engine* penyimpanan. Aplikasi seperti XAMPP kini lebih sering menyertakan *MariaDB*, mencerminkan popularitasnya di komunitas *open-source*[13].

## 2.10 Penelitian Terkait

Dalam sub bab ini, penulis merangkum jurnal-jurnal terkait yang menjadi landasan dalam penyusunan penelitian ini. Daftar jurnal yang relevan dapat ditemukan dalam tabel di bawah ini:

Tabel 2.1 Tahapan Penelitian

| No | Nama dan Tahun   | Judul   | Topik   | Subjek                               | Hasil  |
|----|--|---|---|--------------------------------------|--|
| 1  | Robby Yuli Endra, Yuthsi Aprilinda, Yanuarius Yanu Dharmawan, Wahyu Ramadhan, 2022 | Analisis Perbandingan Bahasa Pemrograman PHP <i>Laravel</i> dengan <i>PHP Native</i> pada Pengembangan <i>Website</i>             | Perbandingan  | <i>Laravel</i> dan <i>PHP Native</i> | Hasil Perbandingan   |
| 2  | Husnil Kamil, M.T, 2023  | Pembangunan Sistem Informasi Pengelolaan Data Pegawai Universitas Andalas Berbasis <i>Web</i> Dengan Fasilitas <i>Web Service</i> | Pembangunan Sistem Informasi Akademik Berbasis <i>Web</i>     | Pengelolaan Data Pegawai             | Sistem Informasi Berbasis <i>web</i> dengan fasilitas <i>Web Service</i> |
| 3  | Irma Yunita Ruhiawati, Siti Sopiah, 2019   | Perancangan Sistem Informasi Akademik Sekolah Berbasis <i>Website</i> Dengan <i>Laravel 5</i> Pada SMK Negeri 1 Cileles           | Perancangan sistem informasi akademik berbasis <i>website</i> | SMK Negeri 1 Cileles                 | Sistem Informasi Akademik Berbasis <i>Web</i>                            |

Pada Tabel 2.1 yang menyajikan penelitian terkait, penelitian yang dilakukan memiliki dasar dan acuan yang relevan dengan topik yang diteliti, yaitu perancangan sistem menggunakan *framework Laravel*. Penelitian-penelitian

tersebut memberikan wawasan tentang implementasi dan keuntungan menggunakan *Laravel* dalam berbagai proyek sistem informasi. Dengan merujuk pada penelitian-penelitian tersebut, penelitian ini dapat mengoptimalkan penggunaan *framework Laravel* untuk menghasilkan sistem yang lebih efektif dan efisien.

## 2.11 Posisi Penelitian

Tabel 2.2 Posisi Penelitian

| No | Nama dan Tahun   | Judul   | Pengujian berdasarkan penelitian terdahulu | Framework Scrum | Laravel | Studi Kasus Nyata |
|----|--|---|--|-----------------|---------|-------------------|
| 1  | Robby Yuli Endra, Yuthsi Aprilinda, Yanuarius Yanu Dharmawan, Wahyu Ramadhan, 2022 | Analisis Perbandingan Bahasa Pemrograman PHP <i>Laravel</i> dengan <i>PHP Native</i> pada Pengembangan <i>Website</i>             | ✓  |                 | ✓       |                   |
| 2  | Husnil Kamil, M.T, 2023  | Pembangunan Sistem Informasi Pengelolaan Data Pegawai Universitas Andalas Berbasis <i>Web</i> Dengan Fasilitas <i>Web Service</i> |  |                 | ✓       | ✓                 |
| 3  | Irma Yunita Ruhiawati, Siti Sopiah, 2019   | Perancangan Sistem Informasi Akademik Sekolah Berbasis <i>Website</i> Dengan <i>Laravel 5</i> Pada SMK Negeri 1 Cileles           |  |                 | ✓       | ✓                 |

| No | Nama dan Tahun                               | Judul  | Pengujian berdasarkan penelitian terdahulu | Framework Scrum | Laravel | Studi Kasus Nyata |
|----|--|--|--|-----------------|---------|-------------------|
| 4  | Fadhlorrohman Zuhdi, Tifani Nabarian, Nasrul | Rancang Bangun Sistem Informasi Akademik untuk Optimasi dan Peningkatan kolaborasi antar <i>developer</i> menggunakan <i>framework scrum</i> | ✓  | ✓               | ✓       | ✓                 |

Pada Tabel 2.2 yang menyajikan penelitian terkait, penelitian yang dilakukan memiliki dasar dan acuan yang relevan dengan topik yang diteliti, yaitu perancangan sistem menggunakan *framework Laravel*. Penelitian-penelitian tersebut memberikan wawasan tentang implementasi dan keuntungan menggunakan

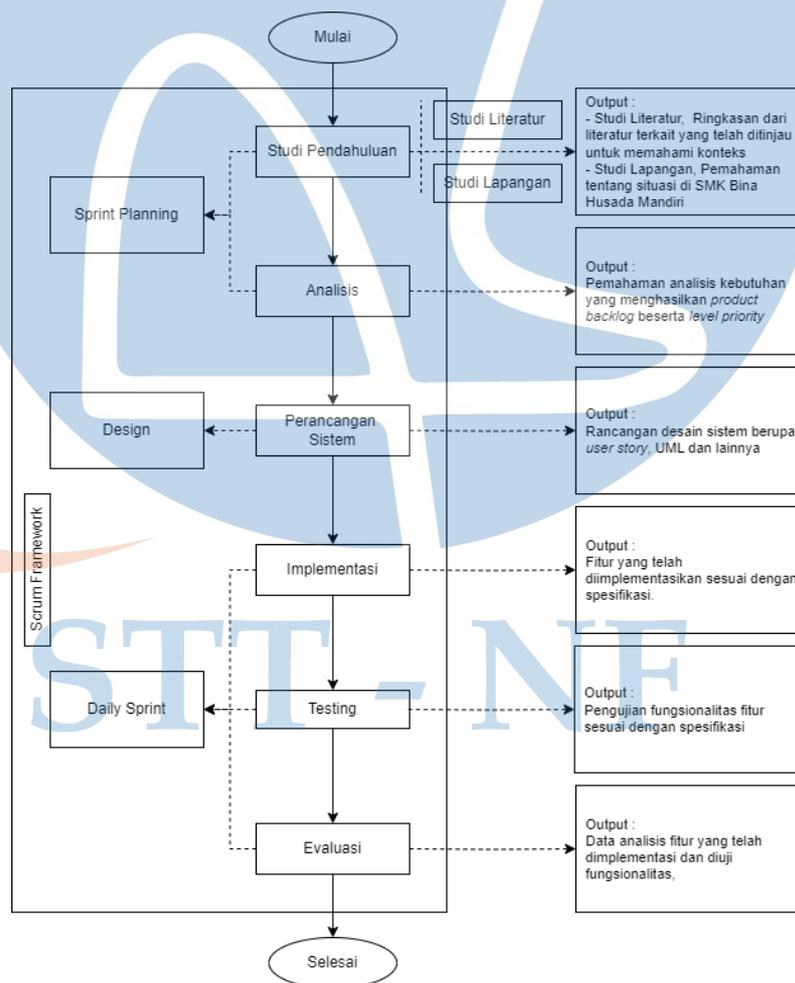
STT - NF

## BAB III METODOLOGI PENELITIAN

Bab ini memberikan penjelasan detail mengenai langkah-langkah yang dilakukan dalam pelaksanaan penelitian, yang memberikan gambaran yang jelas tentang proses penelitian.

### 3.1 Tahapan Penelitian

Dalam penelitian ini, tahapan penelitian akan mengikuti metode *Agile*, khususnya kerangka kerja *Scrum*. Berikut merupakan tahapan dari penelitian :



*Gambar 3.1 Tahapan Penelitian*

### **3.1.1 Studi Pendahuluan(*Sprint Planning*)**

Studi pendahuluan melibatkan dua tahapan utama, yaitu studi literatur dan studi lapangan. Dalam *sprint planning*, tim pengembangan melakukan *review* terhadap literatur terkait serta melakukan pengumpulan data dari lapangan, seperti wawancara dengan *stakeholder* atau observasi langsung di sekolah. Hal ini bertujuan untuk memahami kebutuhan dan tantangan yang dihadapi serta menetapkan rencana untuk tahapan-tahapan berikutnya.

### **3.1.2 Analisis(*Sprint Planning*)**

Tahap analisis dilakukan sebagai bagian dari *sprint planning* untuk menganalisis data yang telah dikumpulkan selama studi pendahuluan. Penulis melakukan pemahaman mendalam terhadap kebutuhan *developer* dan merumuskan spesifikasi sistem yang jelas. Ini melibatkan identifikasi fitur dari sistem yang sudah berjalan dan fungsionalitas yang diperlukan dalam sistem informasi akademik.

### **3.1.3 Perancangan Sistem(*Design*)**

Setelah proses analisis selesai, penulis memulai tahap perancangan sistem informasi akademik. Tahapan ini mencakup pembuatan desain arsitektur sistem, desain basis data, serta desain antarmuka pengguna. Rancangan ini berfungsi sebagai panduan yang digunakan oleh penulis selama proses implementasi sistem.

### **3.1.4 Implementasi(*Daily Sprint*)**

Tahap implementasi dimulai setelah rancangan sistem selesai. penulis mulai mengkodekan sistem berdasarkan rancangan yang telah dibuat. Penulis menggunakan *framework Laravel* dan *MariaDB* sebagai DBMS sesuai dengan batasan masalah. Setiap hari, tim menyelesaikan bagian-bagian tertentu dari kode dan melakukan integrasi kontinu.

### **3.1.5 Testing(*Daily Sprint*)**

Pengujian dilakukan secara terus-menerus selama proses implementasi. Tim pengembangan menguji setiap fitur dan fungsi yang telah diimplementasikan

untuk memastikan bahwa sistem berfungsi dengan baik dan sesuai dengan spesifikasi yang telah ditetapkan. Pengujian mencakup pengujian fungsional dan perbandingan sistem yang saat ini sudah berjalan dengan sistem yang baru dibuat.

### **3.1.6 Evaluasi(Daily Sprint)**

Setelah semua fitur telah diimplementasikan dan diuji, dilakukan evaluasi dalam dua tahap yaitu *sprint review* dan *sprint retrospective*. Pada *sprint review*, sistem yang telah dibangun didemonstrasikan kepada *stakeholder* untuk mendapatkan umpan balik. Di *sprint retrospective*, tim mengadakan pertemuan reflektif untuk mengevaluasi proses pengembangan, mengidentifikasi pembelajaran, dan merencanakan perbaikan di masa mendatang.

## **3.2 Rancangan Penelitian**

Rancangan penelitian ini menggunakan pendekatan *Research and Development* (R&D) dengan fokus pada pengembangan sistem informasi akademik di SMK Bina Husada Mandiri. Metode analisis yang diterapkan adalah kualitatif, dengan pengumpulan data melalui observasi dan wawancara. Lingkungan pengembangan mencakup perangkat keras dan lunak tertentu. Pengujian dilakukan dengan beberapa pengujian salah satunya metode *black box testing*, sementara implementasi dan evaluasi mengikuti pendekatan *Scrum* dan R&D dengan pendekatan iteratif.

### **3.2.1 Jenis Penelitian**

Penelitian ini termasuk dalam jenis penelitian pengembangan dengan pendekatan *Research and Development* (R&D) yang bertujuan untuk mengembangkan sistem informasi akademik yang lebih efisien dan terstruktur dengan memanfaatkan metode pengembangan *Agile*, khususnya *framework Scrum*. *Output* penelitiannya adalah perancangan sistem informasi akademik berbasis *web* menggunakan *framework Laravel*, yang memungkinkan optimalisasi pengelolaan data siswa, pegawai, mata pelajaran, kelas, ruangan, dan unit di SMK Bina Husada Mandiri. Dengan pendekatan ini, diharapkan sistem informasi akademik yang

dikembangkan dapat meningkatkan efisiensi operasional, kolaborasi antar *developer*, dan responsivitas terhadap perubahan kebutuhan pendidikan di SMK Bina Husada Mandiri.

### **3.2.2 Metode Analisis Data**

Metode analisis yang digunakan adalah metode kualitatif. Pendekatan kualitatif digunakan untuk memahami secara mendalam fenomena yang terkait dengan pengembangan sistem informasi akademik di SMK Bina Husada Mandiri. Analisis kualitatif akan membantu dalam menggali pemahaman yang lebih dalam terkait dengan tantangan yang dihadapi, kebutuhan pengguna, dan aspek-aspek kualitatif lainnya yang relevan dengan pengembangan sistem informasi akademik menggunakan *framework Laravel*.

### **3.2.3 Metode Pengumpulan Data**

Metode pengumpulan data dilakukan melalui observasi dan wawancara. Observasi dilakukan dengan memeriksa *source code* sistem informasi akademik yang sudah ada saat ini untuk memahami struktur, fitur, dan kinerja sistem tersebut. Sementara itu, wawancara dilakukan dengan staf *developer* sekolah untuk mendapatkan pemahaman yang lebih mendalam tentang tantangan yang dihadapi, kebutuhan pengguna, dan harapan terhadap pengembangan sistem informasi akademik yang baru. Kombinasi dari observasi dan wawancara diharapkan dapat memberikan pemahaman yang komprehensif dan mendalam terhadap kondisi eksisting serta kebutuhan yang harus dipenuhi dalam pengembangan sistem informasi akademik yang baru.

### **3.2.4 Metode Pengujian**

Dalam pengembangan sistem informasi akademik, pengujian merupakan tahapan kritis untuk memastikan bahwa sistem yang dibangun memenuhi spesifikasi dan kebutuhan pengguna. Metode pengujian yang digunakan mencakup dua pendekatan utama, yaitu *black box testing* dan pengujian perbandingan antara

sistem baru dan sistem lama. Berikut adalah penjelasan detail mengenai kedua metode pengujian tersebut:

1. *Blackbox Testing*

Metode ini digunakan untuk memastikan semua fungsionalitas sistem informasi akademik berjalan sesuai dengan spesifikasi yang telah ditetapkan. Pengujian dilakukan oleh staf *developer* sekolah, yang akan menguji sistem melalui tampilan dari sudut pandang pengguna akhir tanpa perlu mengetahui implementasi internal sistem. Pengujian akan mencakup proses *Create, Read, Update, Delete* (CRUD) pada berbagai entitas dalam sistem.

Tabel 3.1 *Blackbox Testing*

| No | Objek Uji                       | Kegiatan Pengujian   | Kesimpulan               |
|----|---------------------------------|--|--------------------------|
| 1  | Data Siswa dan detail siswa     | Menambah, Mengubah dan Menghapus data siswa dari antarmuka pengguna          | Sesuai atau tidak sesuai |
| 2  | Data Pegawai dan detail pegawai | Menambah, Mengubah dan Menghapus data pegawai dari antarmuka pengguna        | Sesuai atau tidak sesuai |
| 3  | Data Kelas                      | Menambah, Mengubah dan Menghapus data kelas dari antarmuka pengguna          | Sesuai atau tidak sesuai |
| 4  | Data Ruangan                    | Menambah, Mengubah dan Menghapus data ruangan dari antarmuka pengguna        | Sesuai atau tidak sesuai |
| 5  | Data Mata Pelajaran             | Menambah, Mengubah dan Menghapus data mata pelajaran dari antarmuka pengguna | Sesuai atau tidak sesuai |
| 6  | Data Tahun Ajaran               | Menambah, Mengubah dan Menghapus data tahun ajaran dari antarmuka pengguna   | Sesuai atau tidak sesuai |

| No | Objek Uji    | Kegiatan Pengujian  | Kesimpulan               |
|----|--------------|---|--------------------------|
| 7  | Data Unit    | Menambah, Mengubah dan Menghapus data unit dari antarmuka pengguna    | Sesuai atau tidak sesuai |
| 8  | Data Jabatan | Menambah, Mengubah dan Menghapus data jabatan dari antarmuka pengguna | Sesuai atau tidak sesuai |

## 2. Pengujian Perbandingan

Metode ini melibatkan perbandingan parameter tertentu antara sistem baru dan sistem lama berdasarkan penelitian yang dilakukan oleh Endra, Robby Yuli, dkk. *Parameter* yang dibandingkan meliputi *line code*, efisiensi kode, dan struktur *folder*. Dengan menggunakan metode ini, penulis dapat mengevaluasi perbedaan antara kedua sistem dan memastikan bahwa sistem baru memiliki kualitas dan keandalan yang lebih baik sebelum diluncurkan.

Tabel 3.2 Pengujian Perbandingan

| No | Objek Uji  | Kegiatan Pengujian  | Kesimpulan               |
|----|--|---------------------|--------------------------|
| 1  | Code untuk ubah data tahun ajaran dari sistem lama dan sistem baru         | Uji Line Code       | Unggul atau tidak unggul |
| 2  | Code untuk menampilkan data dari database dari sistem lama dan sistem baru | Uji Efisiensi Code  | Unggul atau tidak unggul |
| 3  | Struktur folder keseluruhan dari sistem lama dan sistem baru               | Uji Struktur folder | Unggul atau tidak unggul |
| 4  | Tampilan Halaman Data Tahun Ajaran   | Uji Kecepatan Web   | Unggul atau tidak unggul |

### 3.2.5 Metode Implementasi dan Evaluasi

Metode implementasi dan evaluasi dilakukan sesuai dengan pendekatan *Scrum* dan *Research and Development (R&D)*. Implementasi dilakukan melalui pendekatan iteratif yang dipimpin oleh tim *developer*, dengan mengikuti siklus *sprint* dalam kerangka kerja *Scrum*. Setiap iterasi, atau *sprint*, melibatkan

perencanaan, pengembangan, pengujian, dan penyelesaian pekerjaan yang diawasi oleh *Scrum Master*. Evaluasi dilakukan secara terus-menerus dalam setiap *sprint*, dengan melakukan peninjauan progres dan umpan balik dari pemangku kepentingan. Hal ini memungkinkan penyesuaian dan perbaikan yang cepat dalam pengembangan sistem informasi akademik.

### 3.2.6 Lingkungan Pengembangan

Lingkungan pengembangan yang digunakan untuk membangun sistem informasi akademik meliputi:

1. Perangkat Keras:

Laptop ASUS TUF F15 dengan spesifikasi:

- Windows 10
- RAM 16GB
- SSD 512GB
- *Prosesor* Intel i5-11300H
- VGA RTX 3050

2. Perangkat Lunak:

- *Text Editor*: Visual Studio Code
- Bahasa Pemrograman: PHP 8
- *Framework*: Laravel versi 11
- *CSS Framework*: Tailwind CSS
- *DBMS*: MariaDB
- *Server*: XAMPP

Dengan menggunakan lingkungan pengembangan yang telah disebutkan di atas, diharapkan proses pengembangan sistem informasi akademik dapat berjalan dengan lancar dan efisien sesuai dengan tujuan penelitian.

## BAB IV IMPLEMENTASI DAN EVALUASI

Bab ini menjelaskan proses implementasi dan evaluasi sistem informasi akademik yang telah dirancang ulang menggunakan *Laravel*. Metode penelitian yang digunakan adalah *Research and Development (R&D)* dengan pendekatan *Agile Scrum*. Proses ini mencakup analisis kebutuhan, perancangan, dan pengembangan fitur melalui beberapa iterasi *Sprint*. Pada akhir bab, dilakukan pengujian untuk memastikan bahwa sistem memenuhi kebutuhan dan spesifikasi yang telah ditetapkan.

### 4.1 Analisis

Pada tahap ini, penulis menganalisis hasil wawancara dengan tim pengembangan sistem dan melakukan observasi terhadap *source code* sistem yang sudah berjalan di SMK Bina Husada Mandiri. Tujuan dari analisis ini adalah untuk mengidentifikasi masalah dan kebutuhan, serta merancang solusi untuk optimasi dan peningkatan kolaborasi tim pengembang tanpa mengubah alur sistem informasi yang ada.

#### 4.1.1 Masalah, Analisis dan Solusi

Hasil dari wawancara dan observasi menunjukkan beberapa masalah utama yang perlu diatasi. Berikut adalah tabel yang merangkum masalah, analisis, dan solusi yang diidentifikasi:

*Tabel 4.1 Masalah, Analisis dan Solusi*

| No | Masalah              | Analisis  | Solusi   |
|----|----------------------|---|--|
| 1  | Minimnya Dokumentasi | Sistem hanya dipahami oleh pengembang pertama, menyebabkan kesulitan dalam kolaborasi | Migrasi ke <i>Laravel</i> untuk struktur dan dokumentasi yang lebih baik |

| No | Masalah                                   | Analisis   | Solusi  |
|----|---|--|---|
| 2  | Struktur Kode Tidak Terstruktur           | Kode tidak terorganisir, menyulitkan pengembangan dan pemeliharaan | Penggunaan <i>Laravel</i> yang menyediakan arsitektur MVC                                       |
| 3  | <i>Query SQL</i> pada Halaman <i>View</i> | Mengurangi keamanan dan kinerja sistem                             | Implementasi <i>Eloquent ORM</i> di <i>Laravel</i> untuk pemisahan logika bisnis dan presentasi |
| 4  | Penggunaan Tiga <i>Database</i> Utama     | Inefisiensi dan potensi inkonsistensi data                         | Penggabungan tiga <i>database</i> menjadi satu <i>database</i> terintegrasi                     |

Dengan demikian, analisis ini memberikan landasan yang kuat untuk perbaikan sistem dan meningkatkan kolaborasi tim *developer*. Implementasi solusi yang diusulkan diharapkan dapat mengatasi masalah yang ada dan meningkatkan kualitas sistem secara keseluruhan.

#### 4.1.2 *User Story*

*User story* adalah deskripsi sederhana dari sebuah fitur yang ditulis dari perspektif pengguna akhir. Ini membantu tim pengembang memahami kebutuhan dan harapan pengguna terhadap sistem yang sedang dibangun. Berikut adalah tabel *user story* untuk sistem ini:

Tabel 4.2 *User Story*

| No | Sebagai      | Saya Ingin   | Sehingga  |
|----|--------------|--|---|
| 1  | Admin        | Menginput data siswa, pegawai, mata pelajaran, kelas, ruangan, tahun ajaran dan unit               | Saya dapat mengelola data secara keseluruhan dengan mudah |
| 2  | Admin Cabang | Menginput data siswa, pegawai, mata pelajaran, kelas, ruangan yang terkait dengan unit/cabang saya | Saya dapat mengelola data cabang saya dengan mudah        |

| No | Sebagai      | Saya Ingin   | Sehingga   |
|----|--------------|--|--|
| 3  | Admin        | Melihat seluruh daftar data siswa, pegawai, mata pelajaran, kelas, ruangan, tahun ajaran dan unit              | Saya dapat mengetahui informasi semua data dengan mudah            |
| 4  | Admin Cabang | Melihat daftar siswa, pegawai, mata pelajaran, kelas, ruangan yang terkait dengan unit/cabang saya             | Saya dapat mengetahui informasi data cabang saya dengan mudah      |
| 5  | Admin        | Merubah dan menghapus data siswa, pegawai, mata pelajaran, kelas, ruangan, tahun ajaran dan unit               | Saya dapat memperbarui dan menghapus data dengan tepat             |
| 6  | Admin Cabang | Merubah dan menghapus data siswa, pegawai, mata pelajaran, kelas, ruangan yang terkait dengan unit/cabang saya | Saya dapat memperbarui dan menghapus data cabang saya dengan tepat |

#### 4.1.3 *Product Backlog*

Tabel berikut memuat daftar fitur atau pekerjaan yang perlu dilakukan dalam pengembangan sistem informasi akademik SMK Bina Husada Mandiri. Setiap item dalam *Product Backlog* ini memiliki deskripsi singkat tentang fitur atau pekerjaan tersebut, serta prioritasnya.

Tabel 4.3 *Product Backlog*

| No | Fitur atau Pekerjaan      | Deskripsi   | Prioritas |
|----|---------------------------|---|-----------|
| 1  | Integrasi <i>Database</i> | Integrasi tiga <i>database</i> utama menjadi satu <i>database</i> terintegrasi untuk meningkatkan efisiensi pengelolaan data. | Tinggi    |

| No | Fitur atau Pekerjaan                           | Deskripsi  | Prioritas |
|----|--|--|-----------|
| 2  | Migrasi ke <i>Framework Laravel</i>            | Migrasi sistem informasi ke <i>framework Laravel</i> untuk meningkatkan struktur dan kinerja kode.         | Sedang    |
| 3  | Refaktorisasi Struktur Kode                    | Refaktorisasi struktur kode untuk meningkatkan keterbacaan, skalabilitas, dan kemudahan pemeliharaan.      | Sedang    |
| 4  | Otentikasi Pengguna menggunakan <i>Laravel</i> | Implementasi fitur otentikasi pengguna menggunakan <i>Laravel</i> .  | Tinggi    |
| 5  | Manajemen Guru, dan Siswa                      | Pengembangan modul manajemen guru, dan siswa.  | Tinggi    |
| 6  | Manajemen Jadwal Pelajaran                     | Implementasi fitur manajemen jadwal pelajaran untuk pengelolaan yang lebih efisien.                        | Sedang    |
| 7  | Manajemen Unit (Kelas, Ruang, Mapel)           | Pengembangan fitur untuk pengelolaan unit seperti kelas, ruangan, dan mata pelajaran.                      | Sedang    |
| 8  | Manajemen Tahun Ajaran                         | Pengembangan fitur untuk pengelolaan tahun ajaran, termasuk jadwal akademik dan informasi penting lainnya. | Sedang    |

#### 4.1.4 Fase *Sprint*

Tabel di bawah ini menunjukkan rencana fase *Sprint* dalam pengembangan sistem informasi akademik SMK Bina Husada Mandiri. Setiap *Sprint* memiliki durasi waktu tertentu dan tujuan yang harus dicapai oleh tim pengembangan.

Tabel 4.4 Sprint

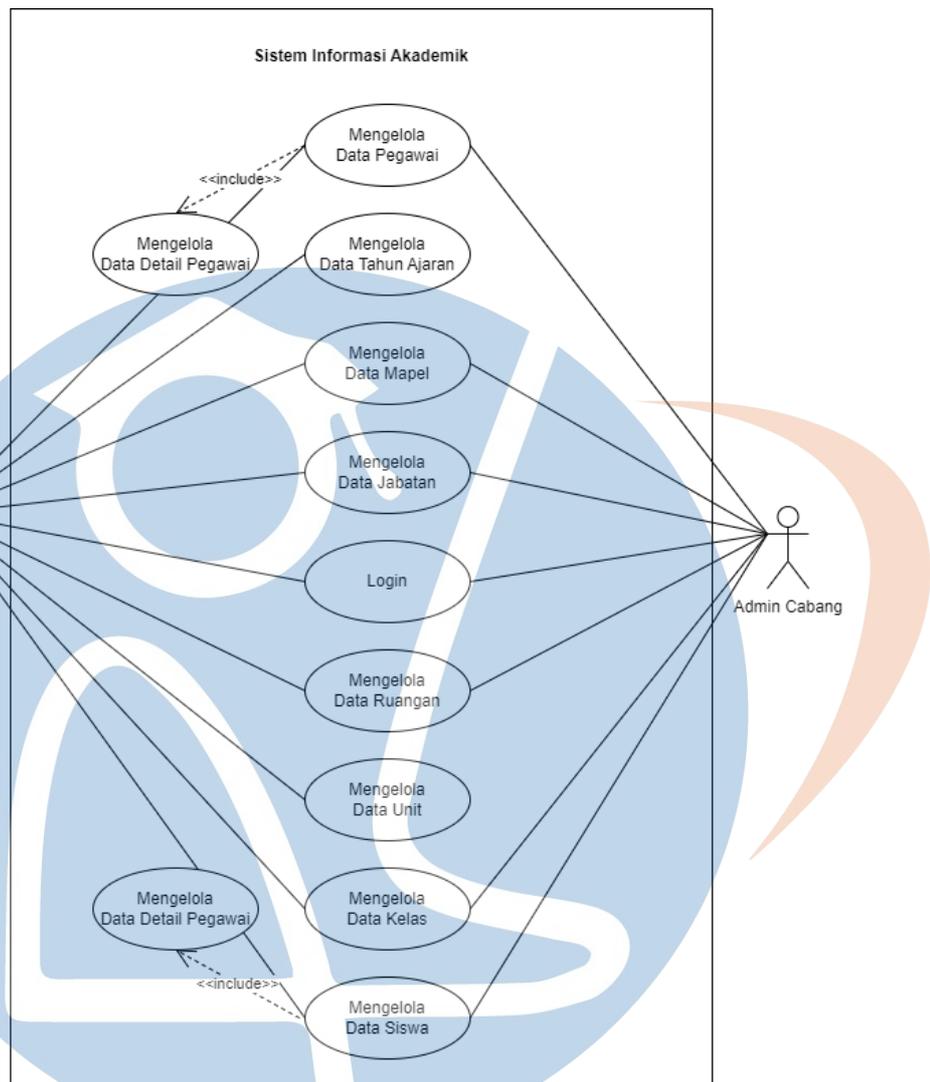
| No | Sprint   | Durasi   | Tujuan Sprint   |
|----|----------|----------|---|
| 1  | Sprint 1 | 2 minggu | Integrasi <i>Database</i> , Migrasi ke <i>Framework Laravel</i>             |
| 2  | Sprint 2 | 2 minggu | Refaktorisasi Struktur Kode, Otentikasi Pengguna menggunakan <i>Laravel</i> |
| 3  | Sprint 3 | 2 minggu | Manajemen Guru, dan Siswa, Manajemen Jadwal Pelajaran                       |
| 4  | Sprint 4 | 2 minggu | Manajemen Unit (Kelas, Ruangan, Mapel), Manajemen Tahun Ajaran              |

## 4.2 Perancangan

Pada tahap perancangan sistem informasi akademik SMK Bina Husada Mandiri, penulis melakukan serangkaian langkah untuk merancang struktur dan fungsionalitas sistem secara komprehensif. Sub bab ini akan membahas langkah-langkah perancangan tersebut, yang meliputi *Use Case Diagram*, *Class Diagram* dan *Activity Diagram*.

### 4.2.1 Use Case Diagram

*Use Case Diagram* merupakan representasi visual dari fungsionalitas sistem dan interaksi antara pengguna dengan sistem. Diagram ini akan membantu dalam pemahaman tentang kebutuhan pengguna dan skenario penggunaan sistem.

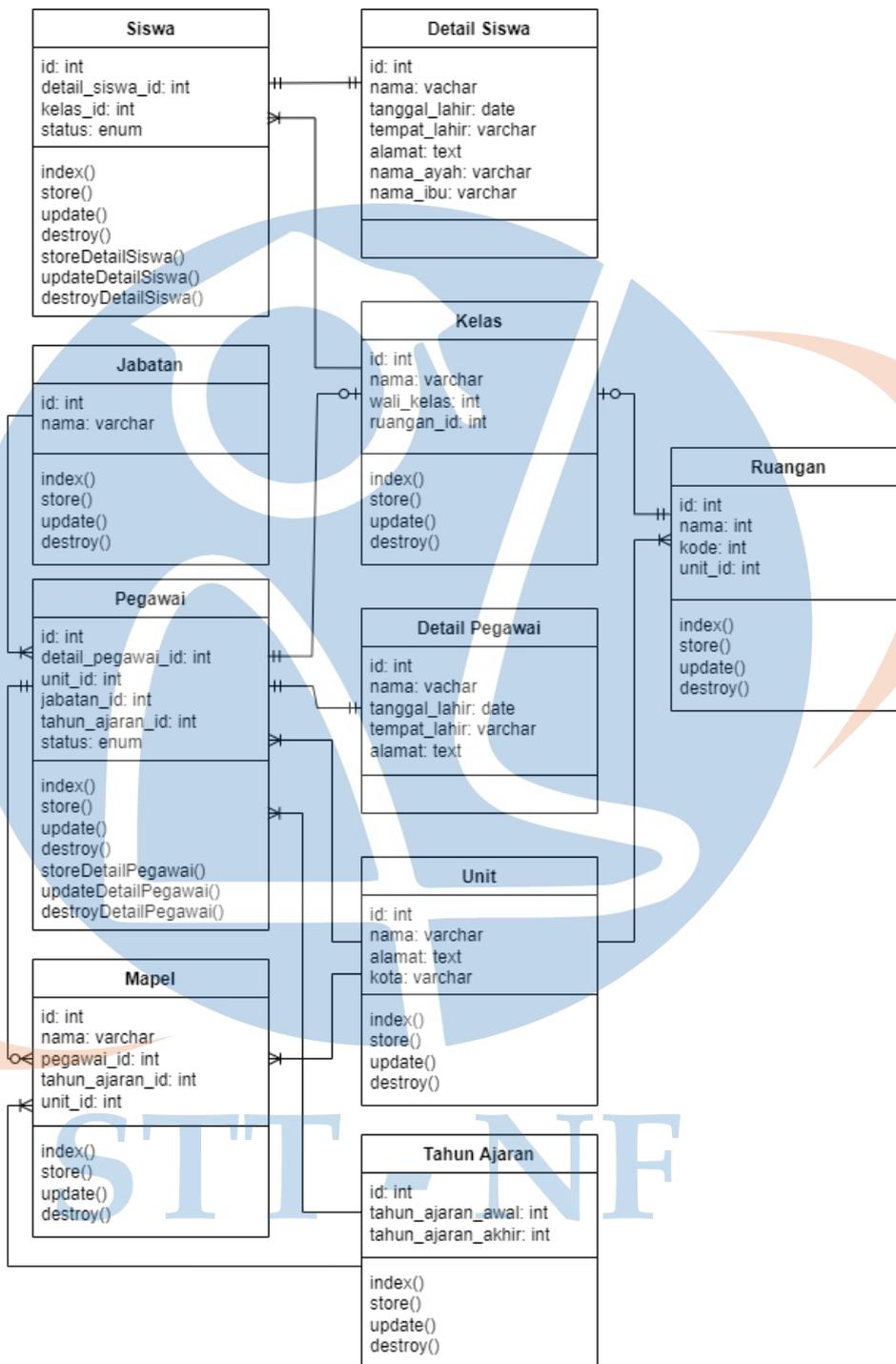


Gambar 4.1 Usecase Diagram

Dalam sistem ini, admin memiliki kemampuan untuk mengolah semua data, termasuk data pegawai, detail pegawai, tahun ajaran, mata pelajaran, jabatan, ruangan, unit, kelas, siswa, dan detail siswa. Sedangkan admin cabang hanya dapat mengolah semua data kecuali unit dan tahun ajaran. Seluruh akses ke data harus melalui proses *login* untuk memastikan keamanan sistem.

#### 4.2.2 Class Diagram

*Class Diagram* adalah diagram yang menggambarkan struktur kelas-kelas, atribut-atribut, dan hubungan antara kelas-kelas dalam sistem. Di bawah ini adalah *Class Diagram* untuk sistem informasi akademik SMK Bina Husada Mandiri.



Gambar 4.2 Class Diagram

Pada gambar 4.2 yang menunjukkan diagram kelas, setiap entitas memiliki serangkaian metode utama, yakni *index*, *store*, *update*, dan *destroy*. Metode *index*

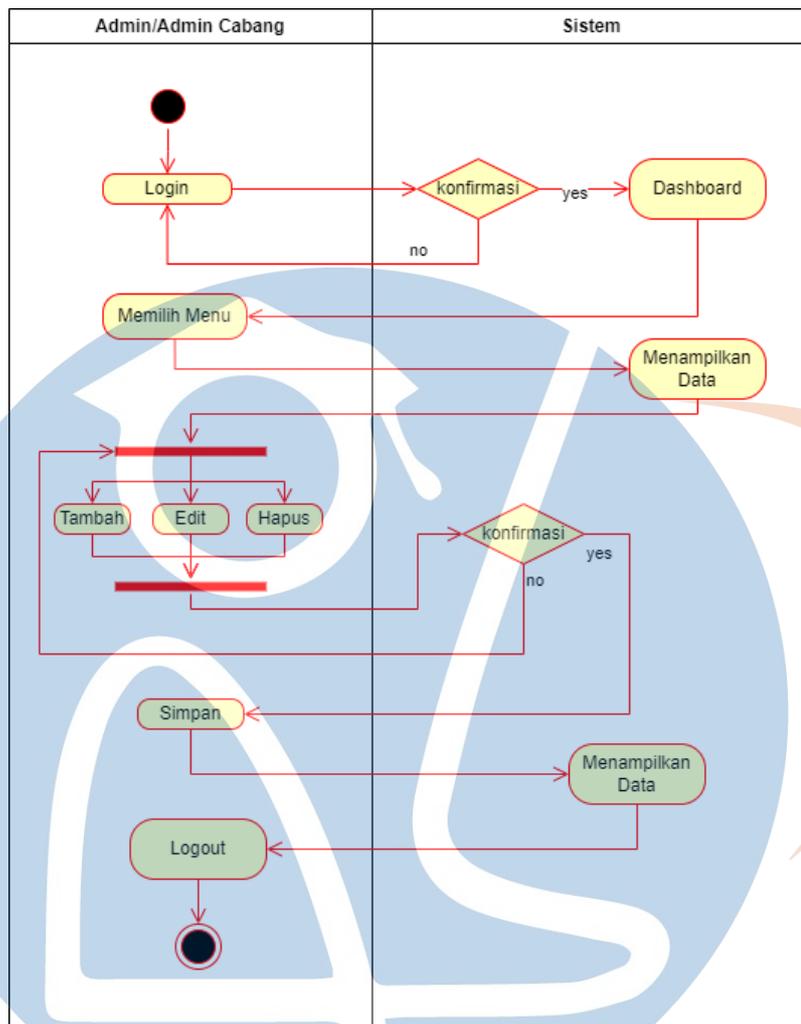
digunakan untuk menampilkan daftar entitas, *store* untuk menambahkan data baru, *update* untuk memperbarui data yang sudah ada, dan *destroy* untuk menghapus data. Semua entitas menerapkan metode-metode ini, kecuali Detail Siswa dan Detail Pegawai. Untuk entitas-entitas ini, manipulasi data dilakukan melalui entitas induk mereka, yaitu Siswa dan Pegawai. Oleh karena itu, pengelolaan data Detail Siswa dan Detail Pegawai dilakukan secara tidak langsung melalui entitas Siswa dan Pegawai. Relasi antar entitas dalam diagram kelas ini terdiri dari:

1. Relasi *One to One*: Siswa-DetailSiswa, Pegawai-DetailPegawai, Kelas-Ruangan, Kelas-Pegawai, Kelas-Siswa.
2. Relasi *One to Many*: Kelas-Siswa, Jabatan-Pegawai, TahunAjaran-Mapel, Unit-Mapel, Unit-Pegawai, TahunAjaran-Pegawai, Pegawai-Mapel.

#### **4.2.3 Activity Diagram**

*Activity Diagram* menggambarkan alur kerja yang dilakukan oleh pengguna dalam sistem. Diagram ini menunjukkan langkah-langkah yang diambil oleh pengguna untuk mengolah data dalam sistem informasi akademik SMK Bina Husada Mandiri. Berikut adalah *Activity Diagram* untuk pengguna (admin dan admin cabang) dalam mengolah seluruh data di sistem:

STT - NF



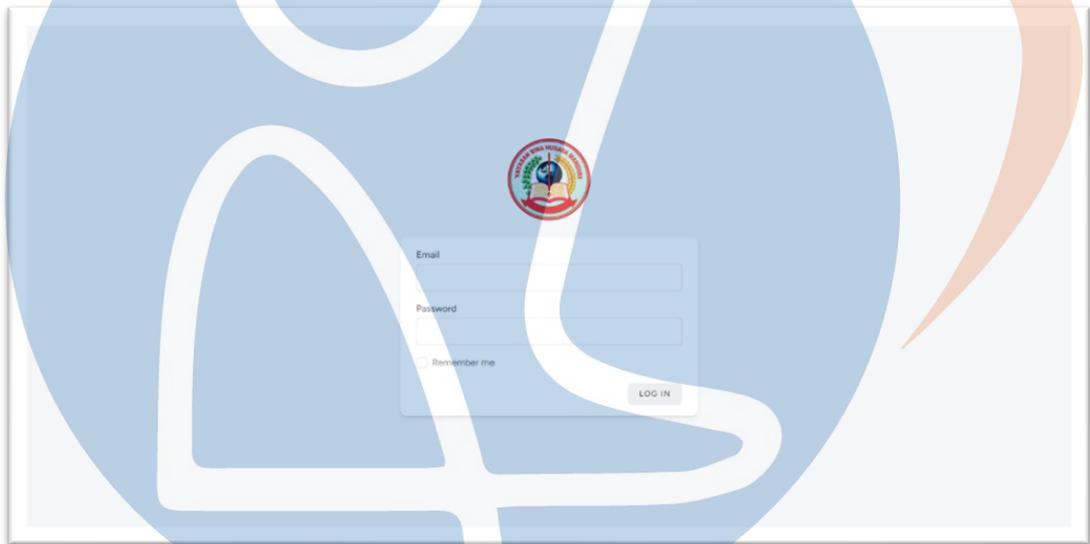
Gambar 4.3 Activity Diagram

Pada gambar 4.3 *Activity Diagram*, menunjukkan langkah-langkah yang diambil oleh pengguna saat mengolah data dalam sistem informasi akademik SMK Bina Husada Mandiri. Proses dimulai dengan pengguna melakukan *login* untuk mengakses sistem. Setelah berhasil *login*, pengguna dapat memilih data yang ingin diolah (siswa, pegawai, unit, ruangan, kelas, mata pelajaran, atau tahun ajaran). Pengguna kemudian dapat melakukan operasi *Create, Read, Update, Delete* (CRUD) sesuai dengan peran dan hak akses mereka. Admin memiliki akses penuh untuk mengolah semua data, sementara admin cabang hanya dapat mengolah data kecuali unit dan tahun ajaran. Setelah selesai, pengguna dapat *logout* dari sistem.

### 4.3 Implementasi

Pada tahap ini, penulis akan membahas halaman-halaman yang terdapat dalam sistem informasi akademik SMK Bina Husada Mandiri. Setiap halaman dirancang untuk menguji fungsionalitas sistem. Penelitian ini tidak berfokus pada tampilan akhir, melainkan pada optimasi dan peningkatan kolaborasi antar *developer*. Tujuannya adalah memastikan bahwa semua fitur berfungsi dengan baik sebelum dilakukan pengembangan lebih lanjut dan penyempurnaan antarmuka..

#### 4.3.1 Halaman Login



Gambar 4.4 Halaman Login

Halaman *login* digunakan oleh pengguna (admin dan admin cabang) untuk mengakses sistem. Pengguna harus memasukkan *username* dan *password* yang *valid* untuk dapat masuk ke dalam sistem.

### 4.3.2 Dashboard



Gambar 4.5 Halaman Dashboard

*Dashboard* adalah halaman utama yang ditampilkan setelah pengguna berhasil *login*. Pada halaman ini, pengguna akan melihat kalimat selamat datang dan navigasi ke berbagai modul dalam sistem.

### 4.3.3 Halaman Manajemen Pegawai

Halaman ini digunakan untuk mengelola data pegawai. Pengguna dapat melihat daftar pegawai, menambahkan pegawai baru, mengubah informasi pegawai, dan menghapus pegawai yang ada.

STT - NF

| No | Nama    | Jabatan | Unit | Tahun Ajaran | Status      |
|----|---------|---------|------|--------------|-------------|
| 1  | Junaedi | Safpam  | SMK  | 2022 - 2023  | -           |
| 2  | Zibda   | Guru    | SMPT | 2022 - 2023  | Tidak Aktif |

*Gambar 4.6 Halaman Data Pegawai*

Pada gambar 4.6 menampilkan halaman data pegawai dengan tabel daftar pegawai yang disesuaikan berdasarkan peran pengguna. Admin cabang hanya dapat melihat data pegawai yang terkait dengan unit atau cabangnya, sedangkan admin pusat memiliki akses ke semua data pegawai tanpa batasan. Ini memastikan akses data sesuai dengan hak akses masing-masing pengguna.

*Gambar 4.7 Halaman Form Tambah Pegawai*

Pada gambar 4.7 menunjukkan halaman *form* tambah data pegawai di *website* Yayasan Bina Husada Mandiri. *Form* ini wajib diisi dengan data nama lengkap, tanggal dan tempat lahir, alamat, unit sekolah, jabatan, tahun ajaran, dan status kepegawaian untuk kebutuhan administrasi dan manajemen.

*Gambar 4.8 Halaman Form Ubah Pegawai*

Pada gambar 4.8 menunjukkan halaman ubah data pegawai di *website* Yayasan Bina Husada Mandiri. Berbeda dengan halaman *form* tambah data, setiap *field* dalam *form* ubah data ini sudah terisi dengan informasi yang sesuai dengan data pegawai yang ada. Data yang tampil meliputi nama lengkap, tanggal dan tempat lahir, alamat, unit kerja, jabatan, tahun ajaran, dan status kepegawaian. Pengguna dapat mengubah informasi yang ada dan menyimpan perubahan untuk memperbarui data pegawai tersebut.

#### **4.3.4 Halaman Manajemen Siswa**

Halaman ini digunakan untuk mengelola data siswa. Pengguna dapat melihat daftar siswa, menambahkan siswa baru, mengubah informasi siswa, dan menghapus siswa yang ada.

| No | Nama | Kelas  | Tempat Lahir | Tanggal Lahir | Alamat         | Nama Ayah | Nama Ibu | Unit  | Status |
|----|------|--------|--------------|---------------|----------------|-----------|----------|-------|--------|
| 1  | Ucup | IPA-11 | Jakarta      | 2024-06-18    | Jalan Guntur 1 | Supardi   | Mulyani  | SMK   | Aktif  |
| 2  | Taep | -      | Depok        | 2024-06-18    | -              | -         | -        | SMPIT | -      |

*Gambar 4.9 Halaman Data Siswa*

Pada gambar 4.9 menampilkan halaman data siswa dengan tabel daftar siswa yang disesuaikan berdasarkan peran pengguna. Admin cabang hanya dapat melihat data siswa yang terkait dengan unit atau cabangnya, sedangkan admin pusat memiliki akses ke semua data siswa tanpa batasan. Ini memastikan akses data sesuai dengan hak akses masing-masing pengguna.

**Add Data Siswa**

Nama Siswa:

Tanggal Lahir:

Tempat Lahir:

Alamat:

Nama Ayah:

Nama Ibu:

Kelas:

Pilih Kelas:

Status:

Pilih Status:

Unit:

Pilih Unit:

*Gambar 4.10 Halaman Form Tambah Siswa*

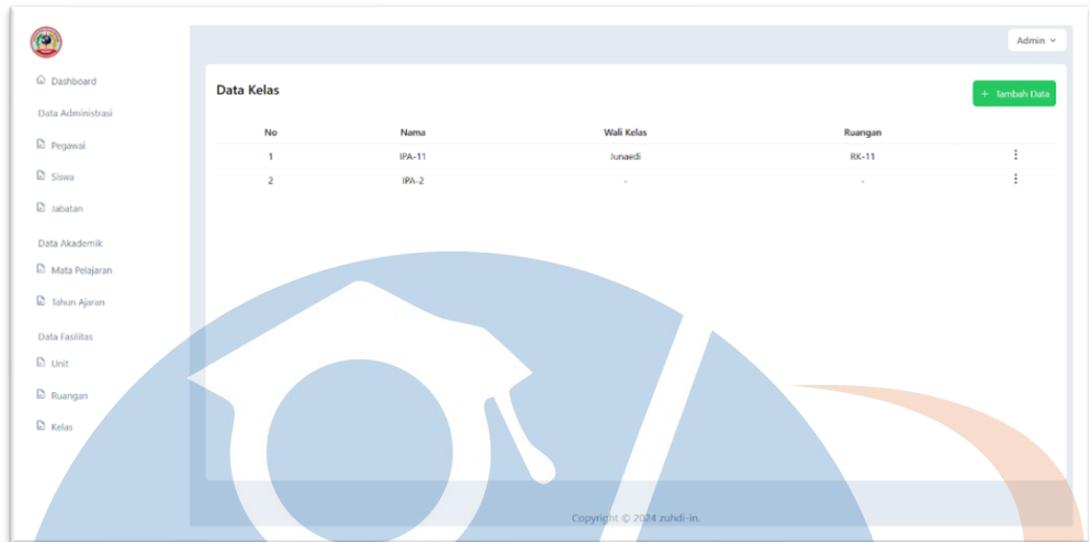
Pada gambar 4.10 menunjukkan halaman *form* tambah data siswa di *website* Yayasan Bina Husada Mandiri. *Form* ini wajib diisi dengan data nama lengkap, tanggal dan tempat lahir, alamat, nama orang tua, kelas, unit sekolah dan status untuk kebutuhan administrasi dan manajemen.

Gambar 4.11 Halaman Form Ubah Siswa

Pada gambar 4.11 menunjukkan halaman *edit* data siswa di *website* Yayasan Bina Husada Mandiri. Berbeda dengan halaman *form* tambah data, setiap *field* dalam *form edit* data ini sudah terisi dengan informasi yang sesuai dengan data siswa yang ada. Data yang tampil meliputi nama lengkap, tanggal dan tempat lahir, alamat, nama orang tua, kelas, unit sekolah dan status. Pengguna dapat mengubah informasi yang ada dan menyimpan perubahan untuk memperbarui data tersebut.

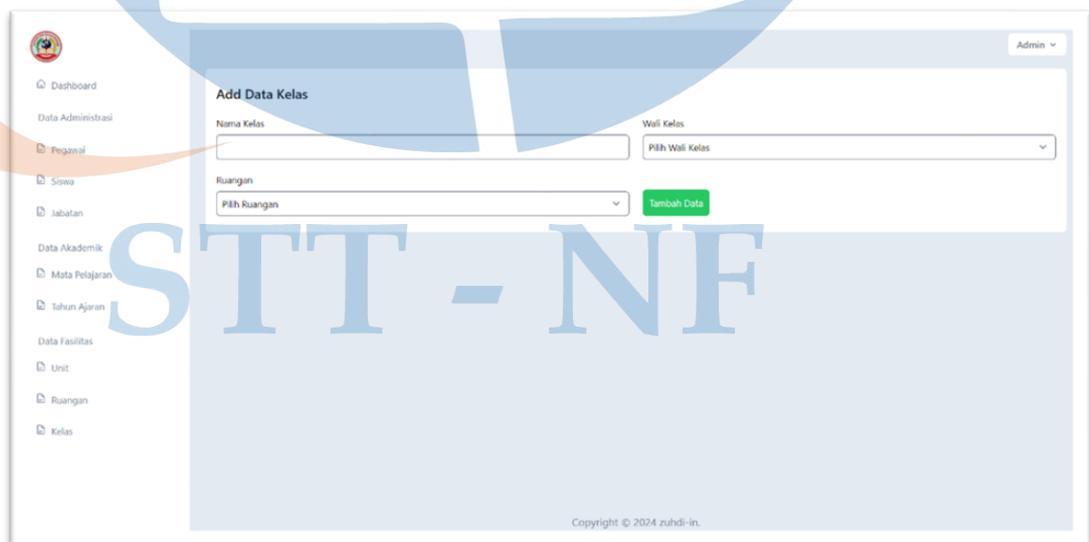
#### 4.3.5 Halaman Manajemen Kelas

Halaman ini digunakan untuk mengelola data kelas. Pengguna dapat melihat daftar kelas, menambahkan kelas baru, mengubah informasi kelas, dan menghapus kelas yang ada.



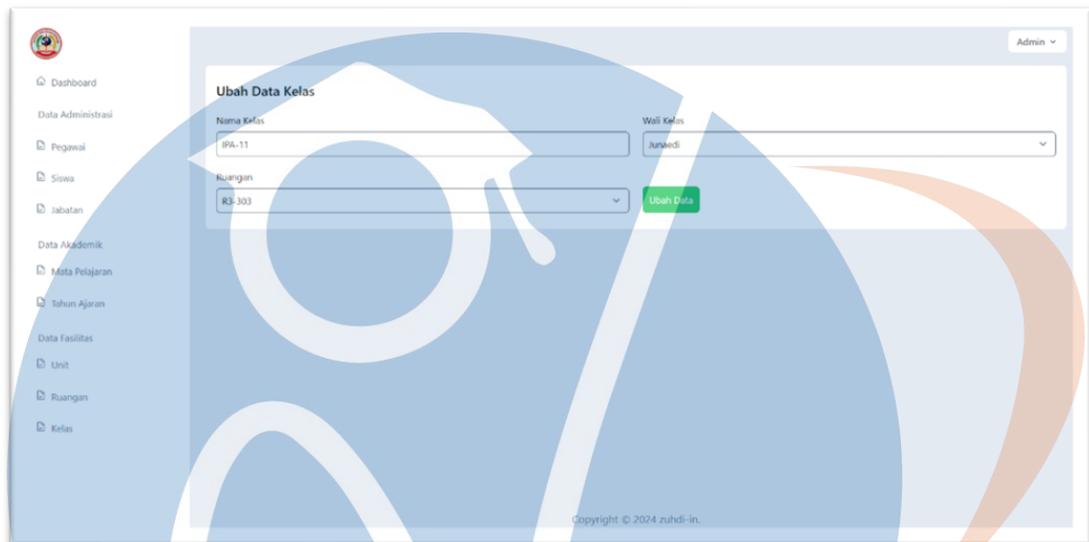
*Gambar 4.12 Halaman Data Kelas*

Pada gambar 4.12 menampilkan halaman data kelas dengan tabel daftar kelas yang disesuaikan berdasarkan peran pengguna. Admin cabang hanya dapat melihat data kelas yang terkait dengan unit atau cabangnya, sedangkan admin pusat memiliki akses ke semua data kelas tanpa batasan. Ini memastikan akses data sesuai dengan hak akses masing-masing pengguna.



*Gambar 4.13 Halaman Form Tambah Kelas*

Pada gambar 4.13 menunjukkan halaman *form* tambah data kelas di *website* Yayasan Bina Husada Mandiri. *Form* ini wajib diisi dengan data nama kelas, wali kelas dan ruangan untuk kebutuhan administrasi dan manajemen.

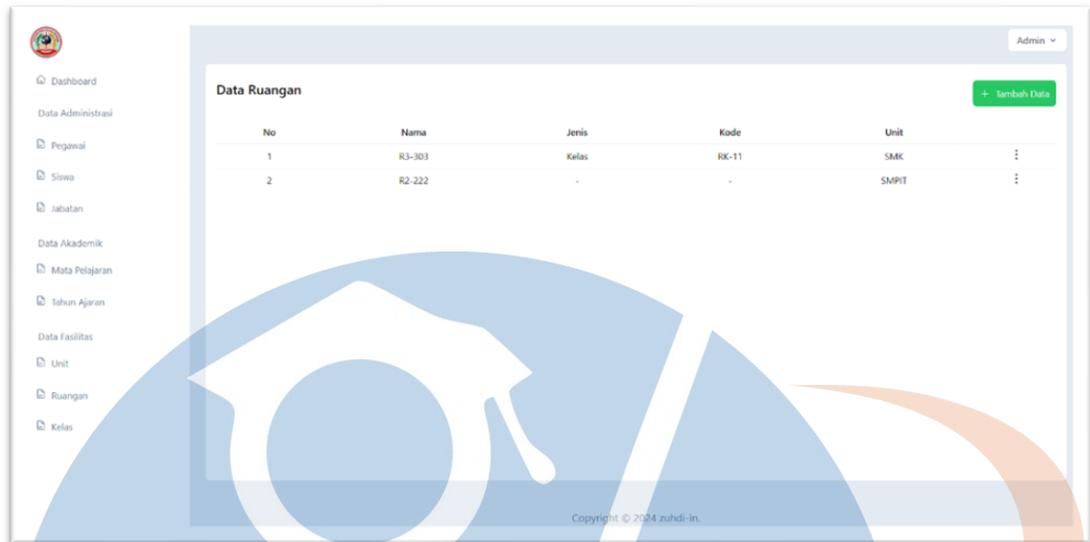


Gambar 4.14 Halaman Form Ubah Kelas

Pada gambar 4.14 menunjukkan halaman *edit* data kelas di *website* Yayasan Bina Husada Mandiri. Berbeda dengan halaman *form* tambah data, setiap *field* dalam *form edit* data ini sudah terisi dengan informasi yang sesuai dengan data kelas yang ada. Data yang tampil meliputi nama kelas, wali kelas dan ruangan. Pengguna dapat mengubah informasi yang ada dan menyimpan perubahan untuk memperbarui data tersebut.

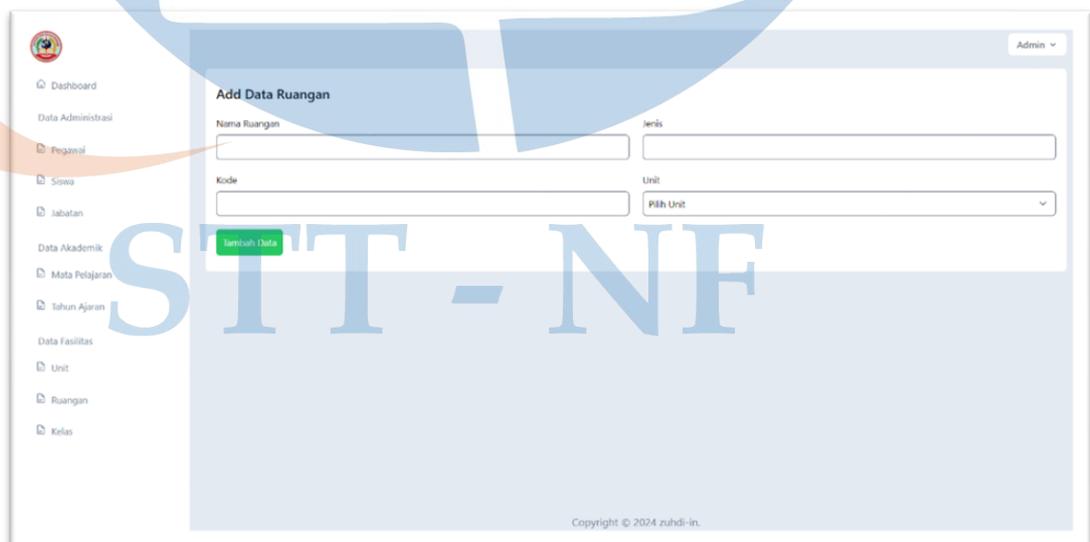
#### 4.3.6 Halaman Manajemen Ruangan

Halaman ini digunakan untuk mengelola data ruangan. Pengguna dapat melihat daftar ruangan, menambahkan ruangan baru, mengubah informasi ruangan, dan menghapus ruangan yang ada.



*Gambar 4.15 Halaman Data Ruangan*

Pada gambar 4.15 menampilkan halaman data ruangan dengan tabel daftar ruangan yang disesuaikan berdasarkan peran pengguna. Admin cabang hanya dapat melihat data ruangan yang terkait dengan unit atau cabangnya, sedangkan admin pusat memiliki akses ke semua data ruangan tanpa batasan. Ini memastikan akses data sesuai dengan hak akses masing-masing pengguna.



*Gambar 4.16 Halaman Form Tambah Ruangan*

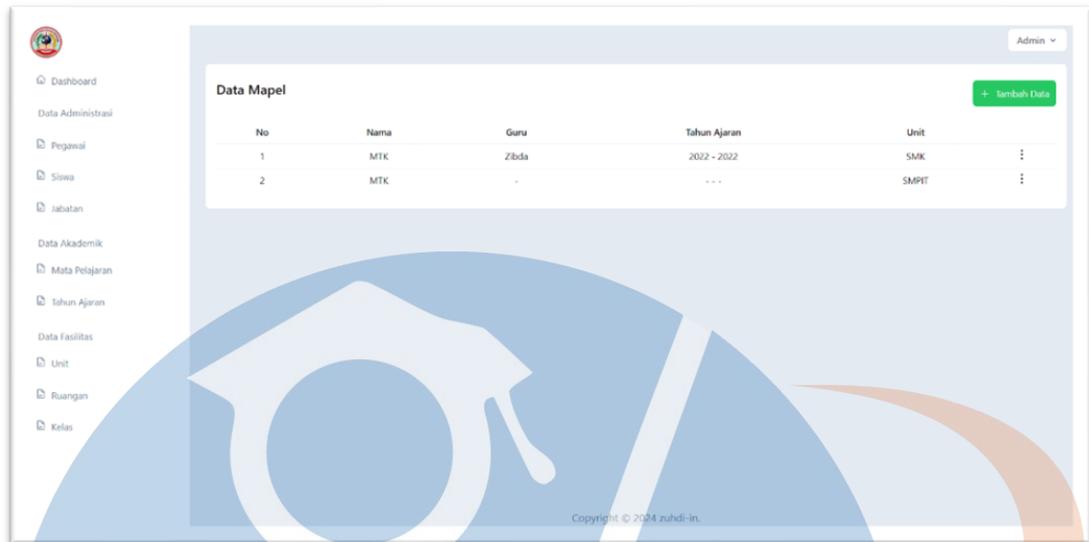
Pada gambar 4.16 menunjukkan halaman *form* tambah data ruangan di *website* Yayasan Bina Husada Mandiri. *Form* ini wajib diisi dengan data nama ruangan, jenis, kode dan unit sekolah untuk kebutuhan administrasi dan manajemen.

Gambar 4.17 Halaman Form Ubah Ruangan

Pada gambar 4.17 menunjukkan halaman *edit* data ruangan di *website* Yayasan Bina Husada Mandiri. Berbeda dengan halaman *form* tambah data, setiap *field* dalam *form edit* data ini sudah terisi dengan informasi yang sesuai dengan data ruangan yang ada. Data yang tampil meliputi nama ruangan, jenis, kode dan unit. Pengguna dapat mengubah informasi yang ada dan menyimpan perubahan untuk memperbarui data tersebut.

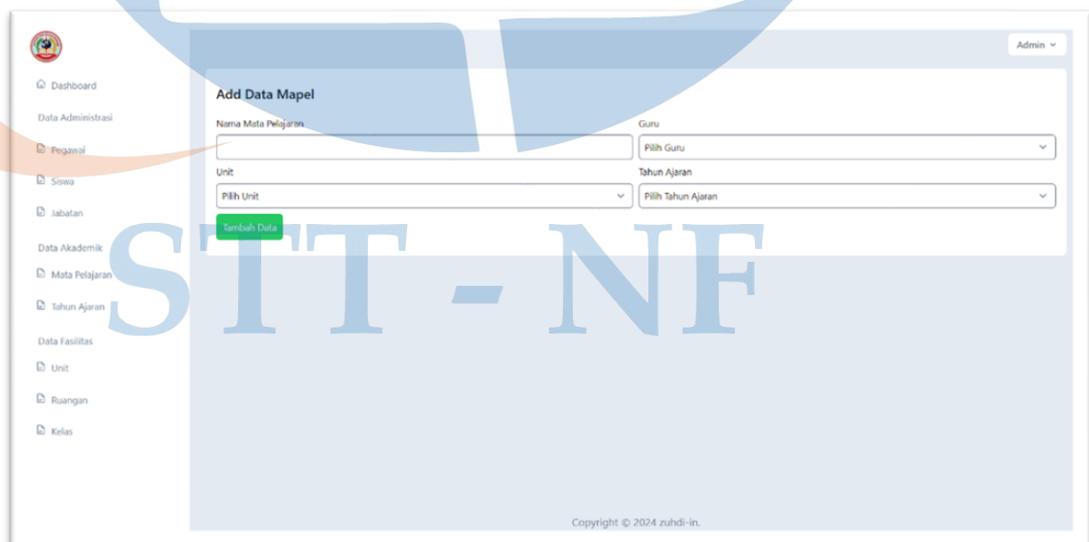
#### 4.3.7 Halaman Manajemen Mata Pelajaran

Halaman ini digunakan untuk mengelola data mata pelajaran. Pengguna dapat melihat daftar mata pelajaran, menambahkan mata pelajaran baru, mengubah informasi mata pelajaran, dan menghapus mata pelajaran yang ada.



*Gambar 4.0.18 Halaman Data Mapel*

Pada gambar 4.18 menampilkan halaman data mapel dengan tabel daftar mapel yang disesuaikan berdasarkan peran pengguna. Admin cabang hanya dapat melihat data mapel yang terkait dengan unit atau cabangnya, sedangkan admin pusat memiliki akses ke semua data mapel tanpa batasan. Ini memastikan akses data sesuai dengan hak akses masing-masing pengguna.



*Gambar 4.19 Halaman Form Tambah Mapel*

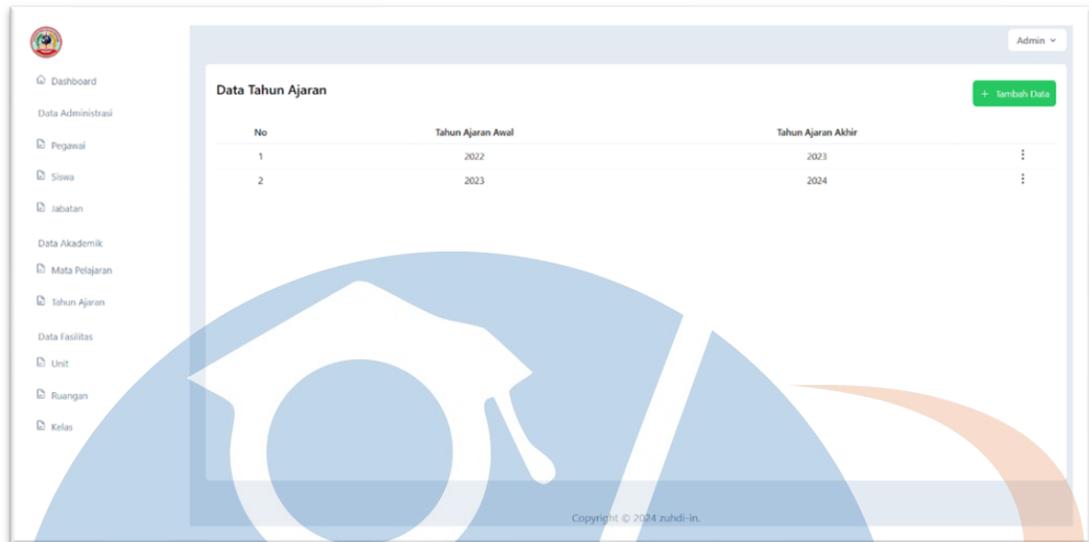
Pada gambar 4.19 menunjukkan halaman *form* tambah data mata pelajaran di *website* Yayasan Bina Husada Mandiri. *Form* ini wajib diisi dengan data nama mata pelajaran, guru, unit sekolah dan tahun ajaran untuk kebutuhan administrasi dan manajemen.

*Gambar 4.20 Halaman Form Ubah Mapel*

Pada gambar 4.20 menunjukkan halaman *edit* data mapel di *website* Yayasan Bina Husada Mandiri. Berbeda dengan halaman *form* tambah data, setiap *field* dalam *form edit* data ini sudah terisi dengan informasi yang sesuai dengan data mapel yang ada. Data yang tampil meliputi nama mata pelajaran, guru, unit sekolah dan tahun ajaran. Pengguna dapat mengubah informasi yang ada dan menyimpan perubahan untuk memperbarui data tersebut.

#### **4.3.8 Halaman Manajemen Tahun Ajaran**

Halaman ini digunakan untuk mengelola data tahun ajaran. Pengguna dapat melihat daftar tahun ajaran, menambahkan tahun ajaran baru, mengubah informasi tahun ajaran, dan menghapus tahun ajaran yang ada.



*Gambar 4.21 Halaman Data Tahun Ajaran*

Pada gambar 4.21 menampilkan halaman data tahun ajaran dengan tabel daftar tahun ajaran. Data ini hanya dapat diakses oleh admin pusat, yang memiliki akses penuh untuk melihat semua informasi terkait tanpa batasan. Ini memastikan bahwa hanya admin pusat yang memiliki wewenang untuk mengelola data tahun ajaran.



*Gambar 4.22 Halaman Form Tambah Tahun Ajaran*

Pada gambar 4.22 menunjukkan halaman *form* tambah data kelas di *website* Yayasan Bina Husada Mandiri. *Form* ini wajib diisi dengan data tahun ajaran awal dan tahun ajaran akhir untuk kebutuhan administrasi dan manajemen.

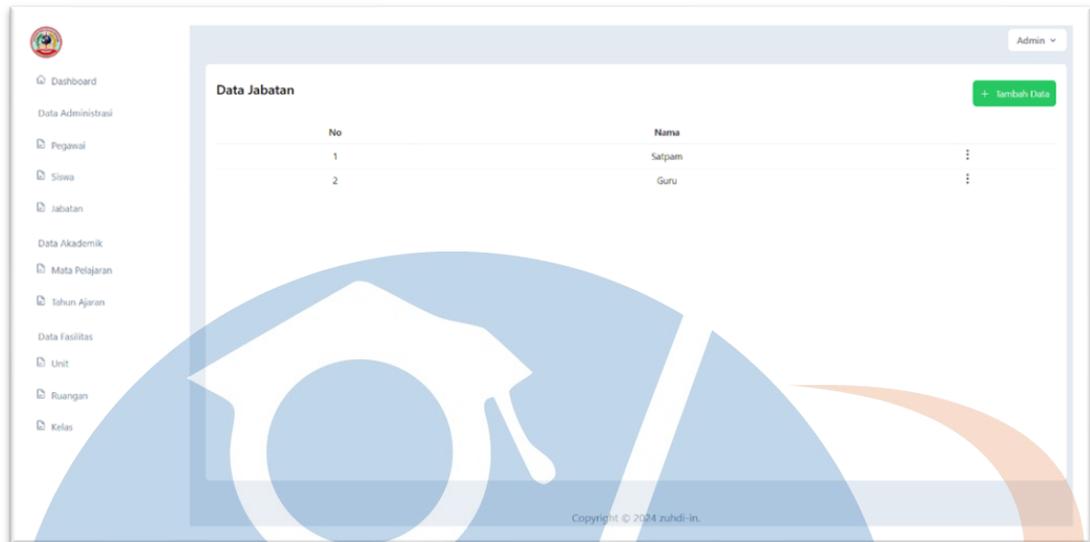


*Gambar 4.23 Halaman Form Ubah Tahun Ajaran*

Pada gambar 4.23 menunjukkan halaman *edit* data tahun ajaran di *website* Yayasan Bina Husada Mandiri. Berbeda dengan halaman *form* tambah data, setiap *field* dalam *form edit* data ini sudah terisi dengan informasi yang sesuai dengan data tahun ajaran yang ada. Data yang tampil meliputi tahun ajaran awal dan tahun ajaran akhir. Pengguna dapat mengubah informasi yang ada dan menyimpan perubahan untuk memperbarui data tahun ajaran tersebut.

#### **4.3.9 Halaman Manajemen Jabatan**

Halaman ini digunakan untuk mengelola data jabatan. Pengguna dapat melihat daftar jabatan, menambahkan jabatan baru, mengubah informasi jabatan, dan menghapus jabatan yang ada.



*Gambar 4.24 Halaman Data Jabatan.*

Pada gambar 4.24 menampilkan halaman data jabatan dengan tabel daftar jabatan. Data ini hanya dapat diakses oleh admin pusat, yang memiliki akses penuh untuk melihat semua informasi terkait tanpa batasan. Ini memastikan bahwa hanya admin pusat yang memiliki wewenang untuk mengelola data jabatan.



*Gambar 4.25 Halaman Form Tambah Jabatan*

Pada gambar 4.25 menunjukkan halaman *form* tambah data jabatan di *website* Yayasan Bina Husada Mandiri. *Form* ini wajib diisi dengan data nama jabatan untuk kebutuhan administrasi dan manajemen.

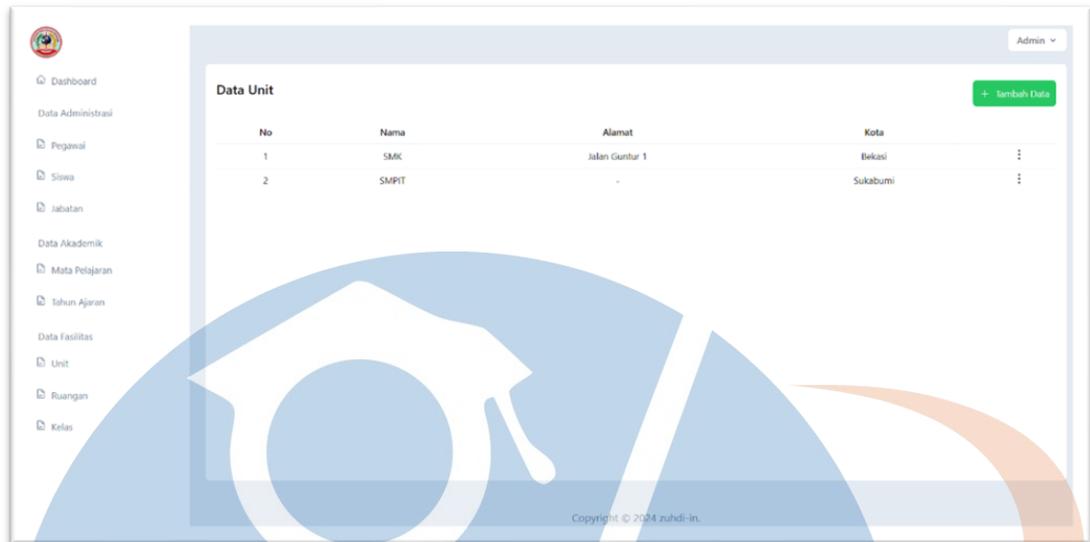


*Gambar 4.26 Halaman Form Ubah Jabatan*

Pada gambar 4.26 menunjukkan halaman *edit* data jabatan di *website* Yayasan Bina Husada Mandiri. Berbeda dengan halaman *form* tambah data, setiap *field* dalam *form edit* data ini sudah terisi dengan informasi yang sesuai dengan data jabatan yang ada. Data yang tampil meliputi nama jabatan. Pengguna dapat mengubah informasi yang ada dan menyimpan perubahan untuk memperbarui data tersebut.

#### **4.3.10 Halaman Manajemen Unit**

Halaman ini digunakan untuk mengelola data unit. Pengguna dapat melihat daftar unit, menambahkan unit baru, mengubah informasi unit, dan menghapus unit yang ada.



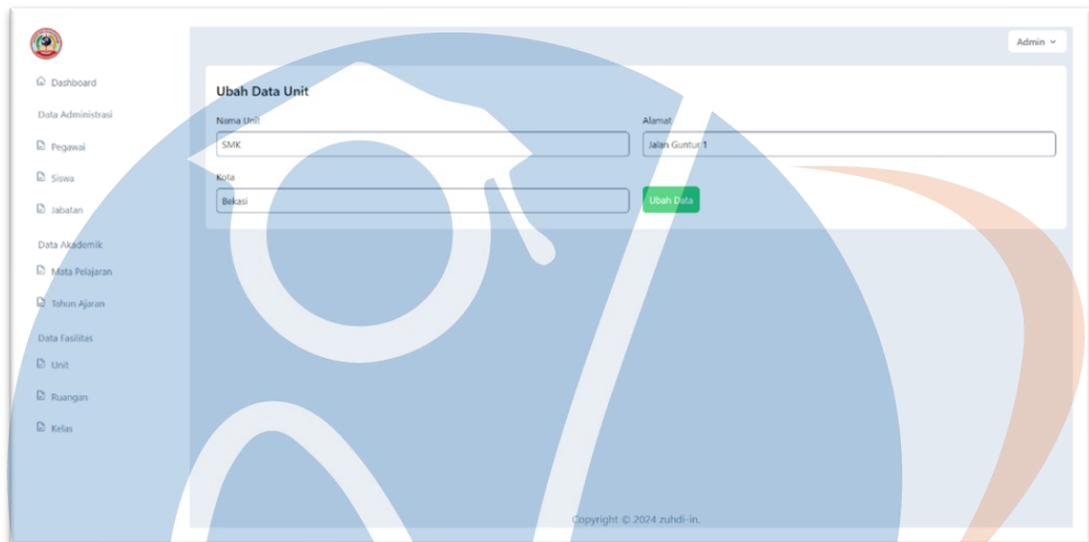
*Gambar 4.27 Halaman Data Unit*

Pada gambar 4.27 menampilkan halaman data unit dengan tabel daftar unit. Data ini hanya dapat diakses oleh admin pusat, yang memiliki akses penuh untuk melihat semua informasi terkait tanpa batasan. Ini memastikan bahwa hanya admin pusat yang memiliki wewenang untuk mengelola data unit.



*Gambar 4.28 Halaman Form Tambah Unit*

Pada gambar 4.28 menunjukkan halaman *form* tambah data unit sekolah di *website* Yayasan Bina Husada Mandiri. *Form* ini wajib diisi dengan data nama unit, alamat dan kota untuk kebutuhan administrasi dan manajemen.



Gambar 4.29 Halaman Form Ubah Unit

Pada gambar 4.29 menunjukkan halaman *edit* data unit di *website* Yayasan Bina Husada Mandiri. Berbeda dengan halaman *form* tambah data, setiap *field* dalam *form edit* data ini sudah terisi dengan informasi yang sesuai dengan data unit yang ada. Data yang tampil meliputi nama unit, alamat dan kota. Pengguna dapat mengubah informasi yang ada dan menyimpan perubahan untuk memperbarui data tersebut.

#### 4.4 Pengujian

Pada tahap ini, penulis melakukan pengujian untuk memastikan bahwa sistem informasi akademik SMK Bina Husada Mandiri berfungsi dengan baik sesuai dengan spesifikasi yang telah dirancang. penulis melakukan beberapa jenis pengujian, termasuk *blackbox testing*, uji *line code*, uji efisiensi *code*, uji keamanan *web* dan evaluasi struktur *folder* untuk membandingkan sistem lama dan sistem baru.

#### 4.4.1 Fokus Pengujian

Fokus pengujian ini meliputi beberapa aspek utama dari sistem untuk memastikan fungsionalitas dan efisiensi. Pengujian dilakukan terhadap pengelolaan data menggunakan *blackbox testing*, serta analisis perbandingan kode dan struktur *folder* antara sistem lama dan sistem baru.

Tabel 4.5 Fokus Pengujian

| No | Objek Uji                       | Pengguna            | Aksi                    | Jenis Uji                 |
|----|---------------------------------|---------------------|-------------------------|---------------------------|
| 1  | Autentikasi                     | Admin, Admin Cabang | <i>Login dan Logout</i> | <i>Blackbox</i>           |
| 2  | Data Pegawai dan Detail Pegawai | Admin, Admin Cabang | Tambah, Ubah dan Hapus  | <i>Blackbox</i>           |
| 3  | Data Siswa dan Detail Siswa     | Admin, Admin Cabang | Tambah, Ubah dan Hapus  | <i>Blackbox</i>           |
| 4  | Data Kelas                      | Admin, Admin Cabang | Tambah, Ubah dan Hapus  | <i>Blackbox</i>           |
| 5  | Data Ruangan                    | Admin, Admin Cabang | Tambah, Ubah dan Hapus  | <i>Blackbox</i>           |
| 6  | Data Mata Pelajaran             | Admin, Admin Cabang | Tambah, Ubah dan Hapus  | <i>Blackbox</i>           |
| 7  | Data Unit                       | Admin               | Tambah, Ubah dan Hapus  | <i>Blackbox</i>           |
| 8  | Data Jabatan                    | Admin               | Tambah, Ubah dan Hapus  | <i>Blackbox</i>           |
| 9  | Data Tahun Ajaran               | Admin               | Tambah, Ubah dan Hapus  | <i>Blackbox</i>           |
| 10 | Ubah Tahun Ajaran               | -                   | Perbandingan            | Uji <i>Line Code</i>      |
| 11 | <i>Query SQL</i>                | -                   | Perbandingan            | Uji Efisiensi <i>Code</i> |

| No | Objek Uji            | Pengguna | Aksi         | Jenis Uji         |
|----|----------------------|----------|--------------|-------------------|
| 12 | Folder Sistem        | -        | Perbandingan | Uji Struktur Code |
| 13 | Halaman Tahun Ajaran | -        | Perbandingan | Uji Kecepatan Web |

#### 4.4.2 Hasil Pengujian *Blackbox*

Penulis melakukan pengujian *blackbox* untuk memverifikasi fungsionalitas sistem yang dirancang. Semua pengelolaan data dilakukan dengan skenario yang sama, namun *inputnya* berbeda sesuai dengan data spesifik masing-masing. Sebagai contoh, pengelolaan data pegawai dijadikan representasi dari semua pengelolaan data. Pengecualian dilakukan hanya pada fitur autentikasi, yang diuji secara terpisah.

Tabel 4.6 Pengujian *Blackbox*

| No          | Skenario   | Input  | Hasil Yang Diharapkan                  | Hasil Pengujian                        | Status   |
|-------------|--|--|--|--|----------|
| Autentikasi |  |  |  |  |          |
| 1           | Email diisi, <i>password</i> tidak diisi tetapi klik <i>button login</i> | Email [admin@example.com] dan <i>password</i> [null] | <i>Please fill this out this field</i> | <i>Please fill this out this field</i> | Berhasil |
| 2           | Email tidak diisi, <i>password</i> diisi tetapi klik <i>button login</i> | Email [null] dan <i>password</i> [12345678]          | <i>Please fill this out this field</i> | <i>Please fill this out this field</i> | Berhasil |
| 3           | Email, <i>password</i> tidak diisi tetapi klik <i>button login</i>       | Email [null] dan <i>password</i> [null]              | <i>Please fill this out this field</i> | <i>Please fill this out this field</i> | Berhasil |

| No                              | Skenario  | Input  | Hasil Yang Diharapkan   | Hasil Pengujian   | Status   |
|---------------------------------|---|--|---|---|----------|
| 4                               | Email, <i>password</i> diisi klik <i>button login</i>                                       | Email [admin@example.com] dan <i>password</i> [12345678]   | Masuk ke halaman <i>dashboard</i>   | Masuk ke halaman <i>dashboard</i>                               | Berhasil |
| 5                               | Mencoba mengakses data yang hanya bisa diakses oleh admin dengan <i>user admin cabang</i>   | Pada kolom <i>url browser</i> memasukan <i>path</i> yang mengarah ke data jabatan/tahun ajaran dan unit  | <i>Redirect</i> ke halaman awal <i>dashboard</i>                            | <i>Redirect</i> ke halaman awal <i>dashboard</i>                | Berhasil |
| Data Pegawai dan Detail Pegawai |   |  |   |   |          |
| 6                               | Masuk Halaman <i>Form Tambah</i>  | Klik <i>button tambah</i>  | Masuk ke halaman <i>form tambah</i>   | Masuk ke halaman <i>form tambah</i>                             | Berhasil |
| 7                               | Semua <i>input</i> diisi kecuali <i>input</i> nama pegawai                                  | Nama pegawai [ <i>null</i> ], tanggal lahir [2002/05/03], tempat lahir [Jakarta], alamat [jalan guntur 1], unit [1], jabatan [1], tahun ajaran [1], status [aktif] | <i>Please select an item in the list</i>                                    | <i>Please select an item in the list</i>                        | Berhasil |
| 8                               | Semua <i>input</i> diisi kecuali salah satu <i>input</i> selain <i>input</i> nama lalu klik | Nama pegawai [Udin], tanggal lahir [2002/05/03], tempat lahir [Jakarta], alamat  | <i>Redirect</i> halaman <i>table data</i> lalu muncul data yang ditambahkan | <i>Redirect</i> halaman <i>table data</i> lalu muncul data yang | Berhasil |

| No | Skenario   | Input  | Hasil Yang Diharapkan  | Hasil Pengujian  | Status   |
|----|--|--|--|--|----------|
|    | tombol tambah  | [jalan guntur 1],<br>unit [null],<br>jabatan [1],<br>tahun ajaran [1],<br>status [aktif]   |  | ditambahkan  |          |
| 9  | Semua <i>input</i> diisi lalu klik tombol tambah                                     | Nama pegawai [Udin],<br>tanggal lahir [2002/05/03],<br>tempat lahir [Jakarta],<br>alamat [jalan guntur 1],<br>unit [2],<br>jabatan [1],<br>tahun ajaran [1],<br>status [aktif] | <i>Redirect</i> halaman table data lalu muncul data yang ditambahkan | <i>Redirect</i> halaman table data lalu muncul data yang ditambahkan | Berhasil |
| 10 | Masuk halaman <i>form</i> ubah   | Klik tombol ubah dan mengirim id pegawai   | Masuk ke halaman <i>form</i> ubah                                    | Masuk ke halaman <i>form</i> ubah                                    | Berhasil |
| 11 | Salah satu dari <i>input</i> tidak diubah tapi tetap klik tombol ubah                | Nama pegawai [Udin],<br>tanggal lahir [2002/05/03],<br>tempat lahir [Jakarta],<br>alamat [jalan guntur 1],<br>unit [2],<br>jabatan [1],<br>tahun ajaran [1],<br>status [aktif] | <i>Redirect</i> halaman table data lalu data ada yang terubah        | <i>Redirect</i> halaman table data lalu data ada yang terubah        | Berhasil |
| 12 | Semua <i>input</i> diisi kecuali <i>input</i> nama dikosongkan tapi klik tombol ubah | Nama pegawai [null],<br>tanggal lahir [2002/05/03],<br>tempat lahir [Jakarta],   | <i>Please select an item in the list</i>                             | <i>Please select an item in the list</i>                             | Berhasil |

| No                             | Skenario  | Input   | Hasil Yang Diharapkan  | Hasil Pengujian  | Status   |
|--------------------------------|---|---|--|--|----------|
|                                |   | alamat<br>[jalan guntur 1],<br>unit [2],<br>jabatan [1],<br>tahun ajaran [1],<br>status [aktif]   |  |  |          |
| 13                             | Semua <i>input</i> diisi kecuali salah satu <i>input</i> selain <i>input</i> nama dikosongkan tapi klik tombol ubah | Nama pegawai [udin],<br>tanggal lahir [2002/05/03],<br>tempat lahir [Jakarta],<br>alamat [jalan guntur 1],<br>unit [1],<br>jabatan [null],<br>tahun ajaran [1],<br>status [aktif] | <i>Redirect</i> halaman <i>table</i> data lalu data ada yang berubah               | <i>Redirect</i> halaman <i>table</i> data lalu data ada yang berubah               | Berhasil |
| 14                             | Semua <i>input</i> diisi lalu klik tombol tambah  | Nama pegawai[Udin],<br>tanggal lahir[2002/05/03],<br>tempat lahir[Jakarta],<br>alamat[jalan guntur 1],<br>unit[2],<br>jabatan,[1],<br>tahun ajaran[1],<br>status[aktif]           | <i>Redirect</i> halaman <i>table</i> data lalu data ada yang berubah sesuai id nya | <i>Redirect</i> halaman <i>table</i> data lalu data ada yang berubah sesuai id nya | Berhasil |
| 15                             | Hapus salah satu data   | Klik tombol hapus dan kirim id data   | <i>Redirect</i> halaman <i>table</i> data lalu data ada yang terhapus sesuai id    | <i>Redirect</i> halaman <i>table</i> data lalu data ada yang terhapus sesuai id    | Berhasil |
| <b>Persentase Keberhasilan</b> |   |   |  |  | 100%     |

Hasil pengujian *blackbox* menunjukkan bahwa sistem yang dikembangkan dengan *Laravel* berfungsi dengan baik. Semua fitur utama, termasuk pengelolaan data unit, jabatan, pegawai, siswa, tahun ajaran, mata pelajaran, kelas, ruangan, serta fungsi *login* dan *logout* untuk admin dan admin cabang, berfungsi sesuai kebutuhan. Pengujian dilakukan secara menyeluruh oleh tim pengembang SMK Bina Husada Mandiri, dengan hasil menunjukkan 100% keberhasilan pada semua skenario yang diuji, tanpa error signifikan. Persentase keberhasilan diperoleh dengan rumus:  $(\text{jumlah skenario berhasil} / \text{total skenario}) \times 100\%$ . Ini mencerminkan kualitas dan kehandalan implementasi *Laravel* yang baik..

#### **4.4.3 Hasil Pengujian *Line Code***

Pengujian *line code* dilakukan untuk mengevaluasi kebersihan dan ketepatan kode dalam pengelolaan data tahun ajaran antara penggunaan *PHP native* dan *Laravel*. Hasil pengujian ini memberikan gambaran perbandingan yang jelas antara kedua pendekatan tersebut.

Dalam pengelolaan data tahun ajaran, penulis melakukan perbandingan antara implementasi menggunakan *PHP native* dan *Laravel*. Berikut adalah hasil analisis perbandingan:

STT - NF

```

1  if ($_POST['tipe'] == "edit tahun ajaran") {
2      mysqli_query($dbMaster,"update tahun_ajaran set angkatan='".addslashes($_POST['angkatan'])."', ta_status='".$_POST['status']."' where id='".$_POST['id']."'");
3      if ($_POST['status'] == "1") {
4          mysqli_query($dbMaster,"update tahun_ajaran set ta_status='2' where id='".$_POST['id']."'");
5      }
6      echo"
7      <script>
8          alert('Data berhasil diperbarui !');
9          document.location.href = '../tahun-ajaran.php';
10     </script>
11     ";
12 }

```

*Gambar 4.30 Line code dalam php native*

```

1  public function update(Request $request, $id)
2  {
3      TahunAjaran::where('id', $id)->update([
4          'tahun_ajaran_awal' => $request->tahun_ajaran_awal,
5          'tahun_ajaran_akhir' => $request->tahun_ajaran_akhir,
6      ]);
7      return redirect()->route('tahun-ajaran.index')->with('success', 'Tahun Ajaran berhasil diupdate.');
```

*Gambar 4.31 Line code dalam laravel*

Perbandingan di atas menunjukkan bahwa penggunaan *Laravel* memiliki keunggulan dalam kejelasan dan ketepatan kode, serta mengurangi jumlah baris kode yang diperlukan untuk tugas yang sama dibandingkan dengan *PHP native*. Sebagai contoh, dalam implementasi *update* tahun ajaran, *PHP native* membutuhkan 12 baris kode, sedangkan dalam *Laravel* hanya memerlukan 8 baris kode. Hal ini mencerminkan efisiensi dan kejelasan dalam penggunaan *Laravel* dibandingkan dengan pendekatan tradisional menggunakan *PHP native*.

#### 4.4.4 Hasil Pengujian Efisiensi Code

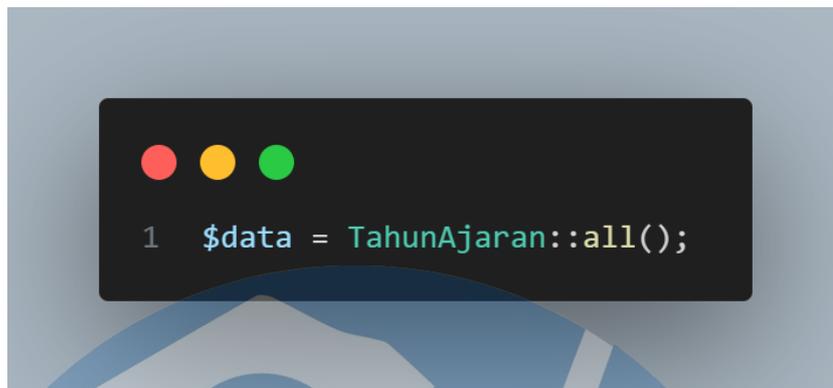
Pengujian efisiensi *code* dilakukan untuk mengevaluasi kinerja dan efisiensi kode dalam menampilkan seluruh data tahun ajaran antara penggunaan *PHP native* dan *Laravel*. Hasil pengujian ini memberikan gambaran perbandingan yang jelas antara kedua pendekatan tersebut.

Dalam menampilkan seluruh data tahun ajaran, penulis melakukan perbandingan antara implementasi menggunakan *PHP native* dan *Laravel*. Berikut adalah hasil analisis perbandingan:

A screenshot of a code editor window with a dark background and light-colored text. The code is PHP and shows database connection details and a query. The code is as follows:

```
1 <?php
2 // koneksi.php
3 define('host', 'localhost');
4 define('user', 'root');
5 define('pass', '');
6 define('dbMaster', 'master');
7 define('dbKurikulum', 'kurikulum');
8 define('dbSdm', 'sdm');
9 $dbMaster = new mysqli(host, user, pass, dbMaster);
10 $dbKurikulum = new mysqli(host, user, pass, dbKurikulum);
11 $dbSdm = new mysqli(host, user, pass, dbSdm);
12
13 // pegawai.php
14 $queryTahun = mysqli_query($dbMaster, "select * from tahun_ajaran");
15
```

Gambar 4.32 Code dalam *php native*



```
1 $data = TahunAjaran::all();
```

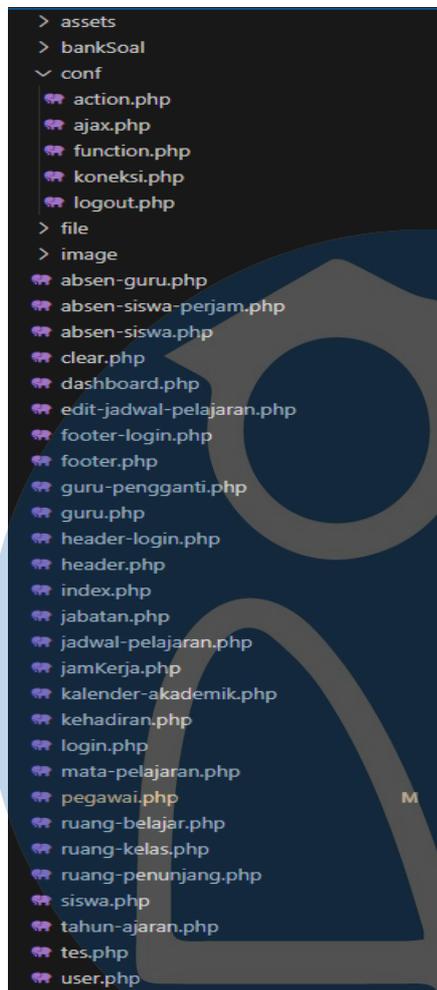
*Gambar 4.33 code dalam laravel*

Gambaran perbandingan di atas menunjukkan bahwa implementasi menggunakan *Laravel* memiliki keunggulan dalam efisiensi kode. *Laravel* mengabstraksi koneksi *database* dan *query SQL* ke dalam metode yang lebih terstruktur dan efisien, sehingga mengurangi redundansi dan risiko kesalahan yang terjadi ketika ada perubahan pada *variabel* koneksi.

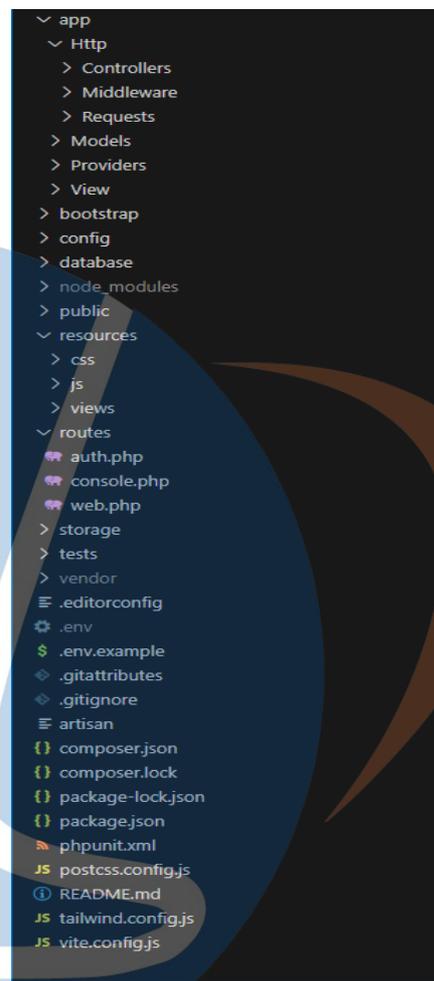
#### **4.4.5 Hasil Pengujian Struktur Folder**

Pengujian struktur *folder* dilakukan untuk mengevaluasi organisasi dan kebersihan struktur *folder* antara penggunaan *PHP native* dan *Laravel* dalam pengembangan sistem informasi akademik SMK Bina Husada Mandiri. Hasil pengujian ini memberikan gambaran perbandingan yang jelas antara kedua pendekatan tersebut.

Dalam evaluasi struktur *folder*, penulis membandingkan penggunaan *Laravel* dengan *PHP native* untuk mengidentifikasi keunggulan struktur kode pada masing-masing pendekatan.



Gambar 4.34 Struktur dalam php native

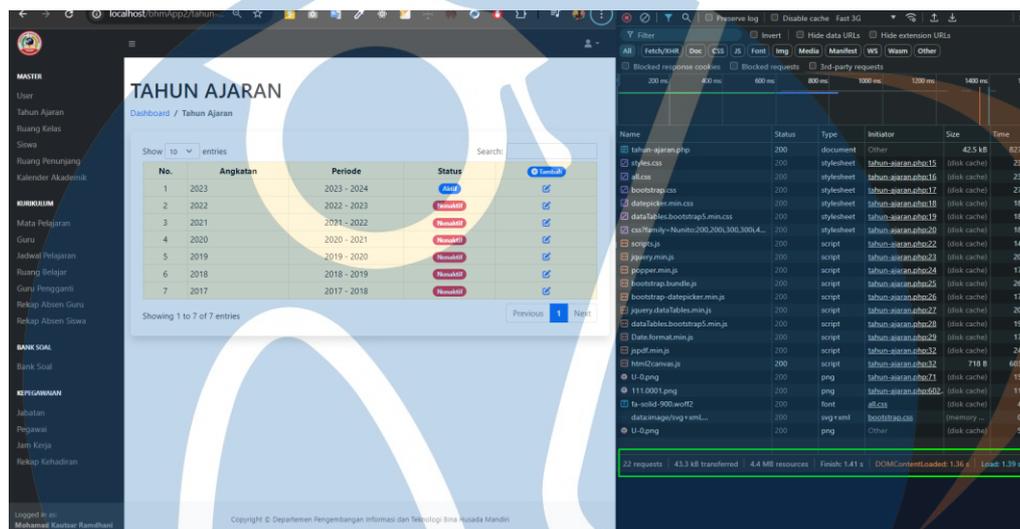


Gambar 4.35 Struktur dalam laravel

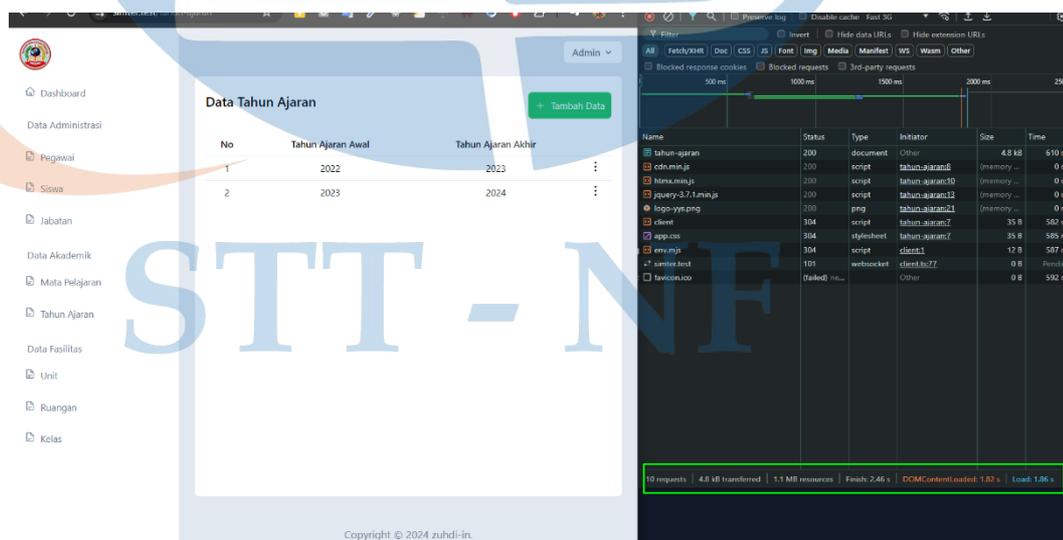
Perbandingan di atas menunjukkan bahwa *Laravel* menawarkan struktur kode yang lebih terorganisir dan terstruktur secara baku dibandingkan dengan pendekatan *PHP native*. Dalam konteks ini, sistem yang menggunakan *Laravel* mengadopsi konsep *Model-View-Controller* (MVC), yang tidak hanya mempermudah pengembangan aplikasi secara efisien dan pemeliharaan yang lebih mudah, tetapi juga mengurangi potensi kesalahan dan meningkatkan keberlanjutan aplikasi dalam jangka panjang. Sebelumnya, struktur sistem lama yang menggunakan *PHP native* tidak mengikuti pola desain MVC, sehingga kurang terstruktur dan membutuhkan upaya lebih besar dalam pengelolaan dan pemeliharannya.

#### 4.4.6 Hasil Pengujian Kecepatan Web

Pengujian kecepatan *web* dilakukan untuk mengevaluasi waktu yang dibutuhkan oleh sistem untuk memuat tampilan halaman data tahun ajaran. Dalam pengujian ini, kami membandingkan kecepatan memuat halaman antara penggunaan *PHP native* dan *Laravel*.



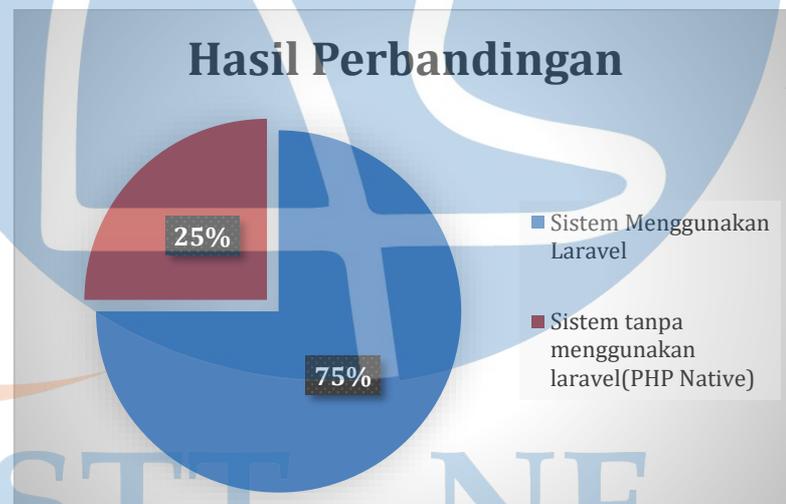
Gambar 4.36 Uji kecepatan pada *php native*



Gambar 4.37 Uji kecepatan pada *laravel*

Pada gambar 4.37 memperlihatkan perbandingan kecepatan memuat halaman antara *PHP native* dan *Laravel* berdasarkan pengujian menggunakan *DevTools* di *browser*. *PHP native* memiliki keunggulan dalam hal ini, karena langsung mengeksekusi skrip tanpa *overhead* dari *framework*. Misalnya, *PHP native* menghasilkan 22 *request* dengan ukuran halaman 42.5kB, dengan waktu *DOMContentLoaded* 1.36 detik dan waktu *load* 1.39 detik. Sementara itu, *Laravel* menghasilkan 10 *request* dengan ukuran halaman 4.8kB, namun waktu *DOMContentLoaded* mencapai 1.82 detik dan waktu *load* 1.86 detik. Meskipun *Laravel* menawarkan fitur-fitur tambahan dan lapisan abstraksi, ini sedikit memperlambat kinerja aplikasi dalam memuat halaman dibandingkan dengan *PHP native*.

#### 4.4.7 Evaluasi Pengujian Perbandingan



Gambar 4.38 Hasil Perbandingan

Pada gambar 4.38 ini menunjukkan perbandingan antara sistem lama dan sistem baru berdasarkan empat kriteria evaluasi utama, yaitu uji *line code*, uji efisiensi *code*, uji kecepatan *web*, dan uji struktur *folder*. Sistem baru mengungguli sistem lama dalam tiga kriteria evaluasi, yaitu uji *line code*, uji efisiensi *code*, dan uji struktur *folder*, sementara sistem lama menunjukkan keunggulan dalam uji kecepatan *web*. Diagram ini mengindikasikan bahwa transisi ke sistem baru telah

menghasilkan peningkatan yang signifikan dalam beberapa aspek, meskipun masih terdapat area di mana sistem lama tetap unggul.



**STT - NF**

## BAB V

### KESIMPULAN DAN SARAN

Pengembangan sistem informasi akademik SMK Bina Husada Mandiri menggunakan *framework Laravel* dan metode pengembangan *Scrum* telah melalui proses yang meliputi analisis, perancangan, implementasi, dan pengujian. Proses ini bertujuan untuk menjawab dua rumusan masalah utama dalam penelitian ini.

#### 5.1 Kesimpulan

Berdasarkan penelitian dan analisis yang telah dilakukan, berikut adalah kesimpulan utama dari pengembangan sistem informasi akademik SMK Bina Husada Mandiri:

1. Sistem informasi akademik SMK Bina Husada Mandiri berhasil dikembangkan menggunakan *framework Laravel* dan metode pengembangan *Scrum*. *Laravel* meningkatkan efisiensi pengembangan melalui fitur seperti *Eloquent ORM* dan struktur MVC. Metode *Scrum* memperbaiki transparansi dan kolaborasi tim, memungkinkan peneliti untuk menunjukkan keunggulan *Laravel* kepada staf *developer* sekolah
2. Sistem yang dikembangkan telah memenuhi kebutuhan pengguna dengan hasil uji *blackbox* menunjukkan tingkat keberhasilan 100%. Pengujian tambahan mengungkapkan bahwa *Laravel* unggul dalam efisiensi dan kualitas kode. Meskipun *PHP native* menawarkan kecepatan *web* yang lebih baik, *Laravel* sudah memenuhi kebutuhan pengguna secara keseluruhan.

#### 5.2 Saran

Berdasarkan hasil penelitian dan pengujian, terdapat beberapa aspek yang perlu diperhatikan untuk pengembangan lebih lanjut. *Laravel* terbukti unggul dalam struktur dan efisiensi kode serta mendukung kolaborasi tim pengembang. Namun, kecepatan *web* masih menjadi area yang memerlukan perhatian khusus. Meskipun *Laravel* menawarkan banyak kelebihan, *PHP native* masih lebih cepat

dalam hal kecepatan respons. Oleh karena itu, fokus pada peningkatan kecepatan *web* akan membantu sistem informasi akademik SMK Bina Husada Mandiri menjadi lebih optimal, baik dari segi fungsionalitas maupun performa, sehingga memberikan pengalaman pengguna yang lebih baik.



STT - NF

## DAFTAR PUSTAKA

- [1] S. Agustiani *dkk.*, “Pengembangan Sistem Informasi Akademik untuk Meningkatkan Efektivitas Pengelolaan Data pada SMK Mihadunal Ula,” *Jurnal Rekayasa Perangkat Lunak*, vol. 4, no. 1, 2023, [Daring]. Tersedia pada: <http://jurnal.bsi.ac.id/index.php/reputasi>
- [2] M. C. F. I., “Pemrograman PHP dengan *framework vs native*, mana yang lebih baik?” Diakses: 25 April 2024. [Daring]. Tersedia pada: [https://tegalkab.go.id/news/view/artikel/pemrograman\\_php\\_dengan\\_framework\\_vs\\_native\\_mana\\_yang\\_lebih\\_baik\\_20201013113939](https://tegalkab.go.id/news/view/artikel/pemrograman_php_dengan_framework_vs_native_mana_yang_lebih_baik_20201013113939)
- [3] P. F. *Laravel dkk.*, “PEMANFAATAN *FRAMEWORK LARAVEL* DAN *FRAMEWORK BOOTSTRAP* PADA PEMBANGUNAN APLIKASI PENJUALAN HIJAB BERBASIS *WEB*,” *Jurnal Media Infotama*, vol. 18, no. 1, hlm. 2022.
- [4] A. Haniefardy, M. Bayu, A. Fadhillah, dan S. Rochimah, “Tinjauan Literatur Sistematis: Pengaruh Penggunaan *Framework* Khusus dalam Proses Pengembangan dan Pembuatan *Web*,” 2019.
- [5] I. Gede Handika dan A. Purbasari, “Konferensi Nasional Sistem Informasi 2018 STMIK Atma Luhur Pangkalpinang,” 2018.
- [6] P. F. *Laravel dkk.*, “PEMANFAATAN *FRAMEWORK LARAVEL* DAN *FRAMEWORK BOOTSTRAP* PADA PEMBANGUNAN APLIKASI PENJUALAN HIJAB BERBASIS *WEB*,” *Jurnal Media Infotama*, vol. 18, no. 1, hlm. 2022.
- [7] I. Irawan Program Studi Pendidikan Guru Sekolah Dasar, F. Ilmu Pendidikan Universitas Pahlawan Tuanku Tambusai Jl Tuanku Tambusai No, dan B. Kamar, “PENGEMBANGAN SISTEM INFORMASI AKADEMIK UNIVERSITAS PAHLAWAN TUANKU TAMBUSAI RIAU,” vol. 1, no. 2, hlm. 55–66, 2018.
- [8] P. P. Arhandi, S. N. Arief, dan A. T. Firdausi, “JIP (Jurnal Informatika Polinema) PENGEMBANGAN *WEBSITE* PENDUKUNG *MASTERY BASED LEARNING* UNTUK PEMBELAJARAN MAHASISWA”.

- [9] F. Agustini, “Implementasi Metode *Scrum* Pada Aplikasi Penjualan Peta Dan Buku (Studi Kasus Pada CV Ubo Rampe Palwoko),” 2023. [Daring]. Tersedia pada: <https://jurnal.bsi.ac.id/index.php/akasia>
- [10] H. Uin, S. Maulana, dan H. Banten, “KONSEP PENELITIAN R&D DALAM BIDANG PENDIDIKAN,” 2017. [Daring]. Tersedia pada: <http://www.aftanalisis.com>
- [11] S. Dika Pratama dan M. Noviansyah Dadaprawira, “Pengujian *Black Box Testing* Pada Aplikasi Edu Digital Berbasis *Website* Menggunakan Metode *Equivalence* Dan *Boundary Value*,” *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, vol. 6, no. 2, hlm. 560–569, 2023, [Daring]. Tersedia pada: <https://ojs.trigunadharma.ac.id/index.php/jsk/index>
- [12] Y. Adisa, M. Irwan, dan P. Nasution, “Konsep Dan Peran Sistem Manajemen Basis Data Relasional Pada Sistem Informasi Manajemen,” vol. 1, no. 3, hlm. 76–83, 2023, doi: 10.59061/masip.v1i3.314.
- [13] A. Fajar dan Z. Imaduddin, “PEMBANGUNAN SISTEM INFORMASI PERTAHANAN SEKOLAH SMA ISLAM ALAZHAR 4 BERBASIS *WEB* MENGGUNAKAN *PHP* & *MARIADB*,” vol. 4, hlm. 82–89, Des 2018.
- [14] S Nisa Amalia Putri I, “*Scrum* Prinsip *Agile* dan Tahapan Dalam Metode *Scrum*,” [bif.telkomuniversity.ac.id](http://bif.telkomuniversity.ac.id).
- [15] “*Black Box Testing – Enhancing User Experience*,” *testbirds*.

STT - NF

## LAMPIRAN

### LAMPIRAN 1 : Sequence Diagram

