



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN SISTEM INFORMASI PENDAFTARAN
SANTRI BARU MENGGUNAKAN FRAMEWORK LARAVEL:
STUDI KASUS DI ISLAMIC CENTER AL-HAZZA**

TUGAS AKHIR

**AGUNG PRIYO SEMBODO
0110219134**

**PROGRAM STUDI TEKNIK INFORMATIKA
DEPOK
AGUSTUS 2024**



**STT TERPADU
NURUL FIKRI**

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN SISTEM INFORMASI PENDAFTARAN
SANTRI BARU MENGGUNAKAN FRAMEWORK LARAVEL:
STUDI KASUS DI ISLAMIC CENTER AL-HAZZA**

TUGAS AKHIR

**AGUNG PRIYO SEMBODO
0110219134**

**PROGRAM STUDI TEKNIK INFORMATIKA
DEPOK
AGUSTUS 2024**

STT - NF

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 : Agung Priyo Sembodo

NIM : 0110219134

Depok, 25 Juli 2024



Agung Priyo Sembodo

STT - NF

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Agung Priyo Sembodo

NIM : 0110219134

Program Studi : Teknik Informatika

Judul Skripsi : Rancang Bangun Sistem Informasi Pendaftaran Santri Baru
Menggunakan Framework Laravel: Studi Kasus Di Islamic Center
Al-Hazza

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



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

Penguji



(Dr. Lukman Rosyidi, M.T, M.M.)

Ditetapkan di : Depok

Tanggal : 25 Juli 2024

STT - NF

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Allah Subhanahu Wa Ta'ala, karena dengan anugerah dan kasih rahmat-Nya, penulis berhasil menuntaskan skripsi/Tugas Akhir ini. Penulisan karya akademik ini dilaksanakan sebagai salah satu persyaratan untuk meraih gelar Sarjana Komputer dari Program Studi Teknik Informatika di Sekolah Tinggi Teknologi Terpadu Nurul Fikri. Penulis menyadari betapa krusialnya peran berbagai pihak dalam setiap tahap, mulai dari masa kuliah hingga proses penyelesaian skripsi. Tanpa dukungan serta bimbingan mereka, penulis tentu menghadapi tantangan besar dalam menyelesaikan skripsi/tugas akhir ini. Oleh karena itu, penulis ingin mengucapkan rasa terima kasih kepada :

1. Allah Subhanahu Wa Ta'ala.
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, M.T, M.M. 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.
5. Bapak Nasrul, S.Pd.I, S.Kom, M.Kom selaku Dosen Pembimbing Akademik dan Dosen Pembimbing Tugas Akhir yang telah membimbing penulis selama berkuliah di Sekolah Tinggi Teknologi Terpadu Nurul Fikri dan dalam menyelesaikan penulisan ilmiah ini.
6. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.
7. Pihak Islamic Center Al-Hazza yang telah membantu penulis untuk memberikan data yang diperlukan bagi penulisan ilmiah ini.

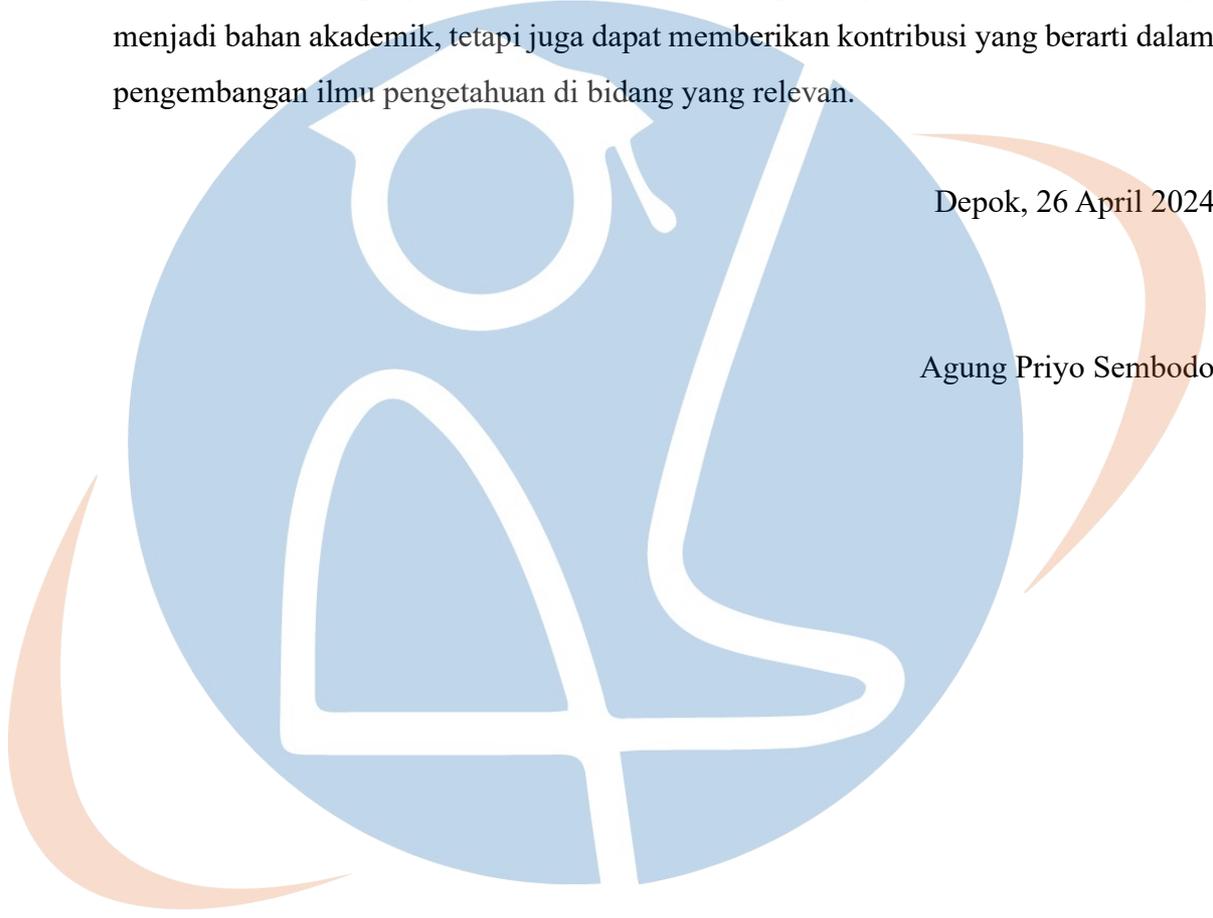
Dalam penyusunan karya ilmiah ini, mungkin terdapat beberapa kekurangan yang muncul akibat dari keterbatasan pengetahuan dan kapasitas yang penulis miliki. Meskipun demikian, penulis telah berupaya keras untuk menyajikan karya ini sebaik mungkin dengan kemampuan yang ada. Jika ada aspek yang kurang memuaskan,

penulis dengan tulus menerima kritik dan masukan dari pembaca sebagai bahan perbaikan di masa depan.

Akhir kata, penulis mengucapkan terima kasih kepada Allah SWT dan berdoa agar Dia membalas segala kebaikan yang telah diberikan oleh semua pihak yang turut serta membantu dalam penyelesaian skripsi ini. Semoga karya ilmiah ini tidak hanya menjadi bahan akademik, tetapi juga dapat memberikan kontribusi yang berarti dalam pengembangan ilmu pengetahuan di bidang yang relevan.

Depok, 26 April 2024

Agung Priyo Sembodo



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 : Agung Priyo Sembodo

NIM : 0110219134

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 Pendaftaran Santri Baru Menggunakan

Framework Laravel: Studi Kasus Di Islamic Center Al-Hazza

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 : 7 Agustus 2024

STT - NF

Yang Menyatakan



(Agung Priyo Sembodo)

ABSTRAK

Nama : Agung Priyo Sembodo
NIM : 0110219134
Program Studi : Teknik Informatika
Judul : Rancang Bangun Sistem Informasi Pendaftaran Santri Baru Menggunakan Framework Laravel: Studi Kasus Di Islamic Center Al-Hazza

Pesantren Islamic Center Al-Hazza merupakan sebuah lembaga pendidikan Islam yang menawarkan jenjang pendidikan mulai dari TK hingga SMA. Salah satu masalah yang dihadapi oleh Pesantren Islamic Center Al-Hazza adalah ketiadaan website resmi untuk proses pendaftaran santri baru. Dari permasalahan ini, diperlukan pengembangan sistem informasi pendaftaran santri baru sebagai solusi yang tepat. Teknologi yang digunakan dalam rancang bangun sistem informasi ini mempermudah pengolahan kode dengan konsep berbasis *Model, View, Controller (MVC)*. Oleh karena itu, teknologi yang digunakan untuk menyelesaikan pembangunan web ini adalah *framework Laravel*. Dalam perancangan website ini, metode Extreme Programming digunakan untuk mempermudah rancangan hingga pengujian sistem. Dengan demikian, tidak ada rancangan yang terlewatkan dalam proses pembuatan. Penelitian ini melibatkan *User Acceptance Test (UAT)* dan hasilnya menunjukkan tingkat keberhasilan fungsional fitur yang sangat baik, mencapai 96,7%. Harapannya, hasil penelitian ini memberikan gambaran keseluruhan mengenai efektivitas solusi yang diusulkan dalam mengatasi kendala pendaftaran santri baru, meningkatkan kemudahan pengguna, dan mempercepat proses pendaftaran santri baru.

Kata Kunci : Pesantren, Pendaftaran Santri Baru, Laravel, Extreme Programming, *Model View Controller (MVC)*, *User Acceptance Test (UAT)*.

ABSTRACT

Name : Agung Priyo Sembodo
NIM : 0110219134
Study Program : *Informatics Engineering*
Title : *Design and Development of New Student Registration Information System Using Laravel Framework: A Case Study at Islamic Center Al-Hazza*

The Islamic Center Al-Hazza Pesantren is an Islamic educational institution offering education levels from kindergarten to high school. One of the issues faced by the Islamic Center Al-Hazza Pesantren is the absence of an official website for the new student registration process. To address this problem, the development of a new student registration information system is required as an appropriate solution. The technology used in the design and development of this information system facilitates code management with a Model, View, Controller (MVC) based concept. Therefore, the technology chosen for building this web application is the Laravel framework. In designing this website, the Extreme Programming method is utilized to streamline the design and testing phases of the system. This ensures that no design aspect is overlooked during the development process. This research involves a User Acceptance Test (UAT), and the results show an excellent functional feature success rate of 96.7%. It is hoped that the findings of this research provide an overall picture of the effectiveness of the proposed solution in addressing the challenges of new student registration, enhancing user convenience, and speeding up the registration process for new students.

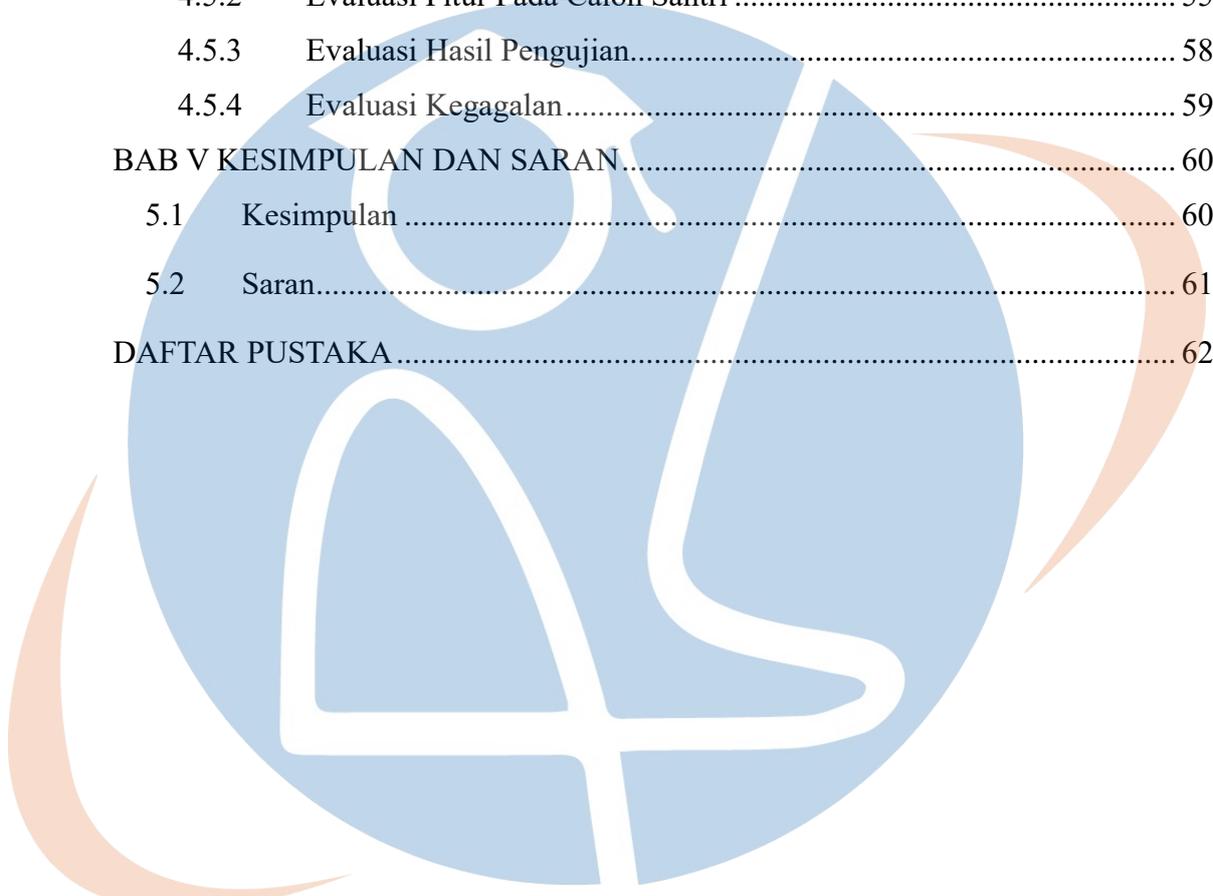
Key Words : *Pesantren, New Student Registration, Laravel, Extreme Programming, Model View Controller (MVC), User Acceptance Test (UAT).*

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian	2
1.5 Batasan Masalah.....	2
1.6 Sistematika Penulisan.....	3
BAB II KAJIAN LITERATUR.....	4
2.1 Sistem Informasi Pendaftaran Santri Baru.....	4
2.2 <i>Framework</i>	4
2.3 <i>Framework Laravel</i>	5
2.4 <i>Model View Controller (MVC)</i>	6
2.5 MySQL.....	6
2.6 <i>Metode Extreme Programming (XP)</i>	7
2.7 <i>Unified Modeling Language</i>	8
2.8 <i>Black Box Testing</i>	9
2.9 <i>User Acceptance Testing (UAT)</i>	9

2.10	Penelitian Terkait.....	10
2.11	Posisi Penelitian	11
BAB III METODOLOGI PENELITIAN.....		14
3.1	Tahapan Penelitian	14
3.1.1	Studi Pendahuluan.....	14
3.1.2	Analisa Perancangan	15
3.1.3	Implementasi	15
3.1.4	Pengujian.....	15
3.1.5	Kesimpulan	16
3.2	Rancangan Penelitian	16
3.2.1	Jenis Penelitian.....	16
3.2.2	Metode Analisis Data	17
3.2.3	Metode Pengumpulan Data	17
3.2.4	Metode Pengujian.....	17
3.2.5	Metode Implementasi dan Evaluasi	19
3.2.6	Lingkungan Pengembangan.....	19
BAB IV IMPLEMENTASI DAN EVALUASI.....		21
4.1	Analisis dan Perencanaan Sistem.....	21
4.1.1	Analisis Sistem Berjalan	21
4.1.2	Analisis Permasalahan dan Usulan Perbaikan	22
4.1.3	Analisis Kebutuhan Sistem	23
4.2	Perancangan Sistem	25
4.2.1	<i>Design Use Case Diagram</i>	25
4.2.2	<i>Design Class Diagram</i>	26
4.2.3	<i>Design Activity Diagram</i>	28
4.3	Implementasi Sistem Pendaftaran Santri Baru.....	32
4.3.1	Implementasi Kode	32
4.3.2	Implementasi Antarmuka	43
4.4	Pengujian Sistem.....	48

4.4.1	<i>Black Box Testing</i>	48
4.4.2	<i>User Acceptance Test</i>	51
4.5	Evaluasi Sistem	52
4.5.1	Evaluasi Fitur Pada Admin.....	53
4.5.2	Evaluasi Fitur Pada Calon Santri	55
4.5.3	Evaluasi Hasil Pengujian.....	58
4.5.4	Evaluasi Kegagalan.....	59
BAB V KESIMPULAN DAN SARAN.....		60
5.1	Kesimpulan	60
5.2	Saran.....	61
DAFTAR PUSTAKA		62



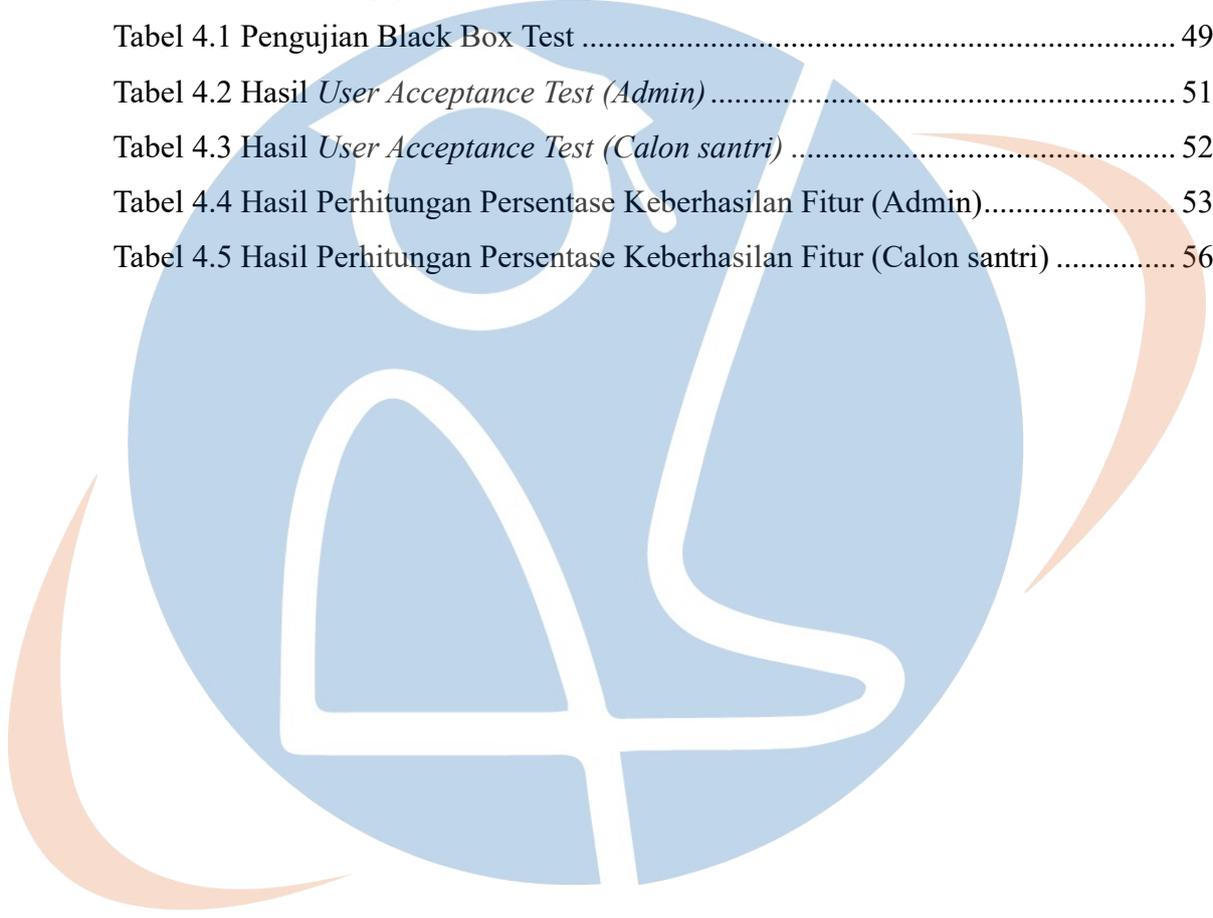
STT - NF

DAFTAR GAMBAR

Gambar 2.1 Metode <i>Extreme Programming</i>	8
Gambar 3.1 Tahapan Penelitian	14
Gambar 4.1 Alur Pendaftaran Santri Baru Saat Ini	21
Gambar 4.2 <i>Use Case Diagram</i>	25
Gambar 4.3 <i>Class Diagram</i>	27
Gambar 4.4 <i>Activity Diagram</i> Calon Santri Melakukan Pendaftaran	28
Gambar 4.5 <i>Activity Diagram</i> Verifikasi Data oleh Panitia Pendaftaran atau Admin	30
Gambar 4.6 Model Santri	32
Gambar 4.7 Model Pendaftaran.....	33
Gambar 4.8 Model Detail Pendaftaran.....	33
Gambar 4.9 Model Profil Pesantren	34
Gambar 4.10 <i>View</i> Pendaftaran Santri	35
Gambar 4.11 <i>View</i> Detail Santri	36
Gambar 4.12 <i>View</i> Dashboard Admin	37
Gambar 4.13 <i>View</i> Kelola Pendaftaran	38
Gambar 4.14 <i>Controller</i> Admin	39
Gambar 4.15 <i>Controller</i> Admin Pendaftaran	40
Gambar 4.16 <i>Controller</i> Santri.....	41
Gambar 4.17 <i>Controller</i> Pendaftaran	42
Gambar 4.18 <i>Dashboard</i> Admin	43
Gambar 4.19 Halaman Kelola Pendaftaran.....	44
Gambar 4.20 Detail Calon Santri	45
Gambar 4.21 Halaman Kelola Calon Santri.....	45
Gambar 4.22 Halaman Landing Page	46
Gambar 4.23 Halaman Pembuatan Akun Pendaftaran.....	47
Gambar 4.24 Halaman Form Pendaftaran.....	47
Gambar 4.25 Gambar Halaman Pendaftaran	48

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait.....	10
Tabel 2.2 Posisi Penelitian	12
Tabel 3.1 Tabel Pengujian <i>User Acceptance Test</i>	18
Tabel 4.1 Pengujian Black Box Test	49
Tabel 4.2 Hasil <i>User Acceptance Test (Admin)</i>	51
Tabel 4.3 Hasil <i>User Acceptance Test (Calon santri)</i>	52
Tabel 4.4 Hasil Perhitungan Persentase Keberhasilan Fitur (Admin).....	53
Tabel 4.5 Hasil Perhitungan Persentase Keberhasilan Fitur (Calon santri)	56



STT - NF

BAB I PENDAHULUAN

1.1 Latar Belakang

Pesantren, sebagai institusi pendidikan Islam yang telah ada sejak lama di Indonesia, masih memiliki peran yang signifikan dalam berbagai bidang, khususnya dalam bidang sosial dan keagamaan[1]. Pesantren Islamic Center Alhazza, yang berlokasi di Balai Karang, Sanggau, Kalimantan Barat, merupakan sebuah lembaga pendidikan Islam yang menawarkan jenjang pendidikan dari TK hingga SMA. Pesantren ini dikenal sebagai pusat pendidikan Islam yang dilengkapi dengan fasilitas yang mendukung dalam proses belajar santri. Selain itu, pesantren ini juga menawarkan beragam kegiatan ekstrakurikuler, termasuk kegiatan keagamaan, seni, olahraga, dan pelayanan Masyarakat, pesantren ini dikelola oleh staf pengajar yang berkualifikasi dan berpengalaman.

Salah satu masalah yang dihadapi oleh Pesantren Islamic Center Alhazza adalah tidak memiliki website resmi untuk proses penerimaan santri baru. Karena tidak adanya *website*, calon santri dan keluarga mereka kesulitan untuk memperoleh informasi yang jelas dan terperinci tentang prosedur pendaftaran, persyaratan yang harus dipenuhi, serta berbagai informasi penting lainnya terkait dengan penerimaan santri baru. Hal ini dapat menghambat potensi calon santri yang berminat untuk bergabung dengan pesantren ini serta menimbulkan kesulitan dalam mengelola dan menyebarkan informasi secara efektif kepada masyarakat yang berpotensi tertarik.

Maka, penelitian diarahkan untuk merancang dan mengembangkan solusi berupa *website* pesantren yang informatif dan interaktif, yang dapat memudahkan proses penerimaan santri baru di Pesantren Islamic Center Alhazza. Dengan merancang website yang mencakup informasi lengkap tentang profil pesantren, kurikulum, fasilitas, kegiatan ekstrakurikuler, serta prosedur dan persyaratan penerimaan santri baru, diharapkan calon santri dan keluarga mereka dapat dengan mudah mengakses informasi yang dibutuhkan. Selain itu, disediakan pula formulir pendaftaran online yang praktis untuk memudahkan proses pendaftaran. Dengan adanya solusi ini, diharapkan pengalaman penerimaan santri baru menjadi lebih efisien

dan menyenangkan bagi semua pihak yang terlibat, dan karena itu penulis memberi judul penelitian tugas akhir ini dengan judul **“Rancang Bangun Sistem Informasi Pendaftaran Santri Baru Menggunakan *Framework* Laravel: Studi Kasus Islamic Center Al-Hazza”**.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang ada pada latar belakang, maka rumusan masalah dapat dirumuskan sebagai berikut :

1. Bagaimana rancang bangun sistem informasi pendaftaran santri baru studi kasus menggunakan framework Laravel di Islamic Center Al-Hazza?
2. Apakah sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza berjalan dengan baik dan sesuai kebutuhan pengguna?

1.3 Tujuan Penelitian

1. Merancang dan membangun sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza.
2. Menguji sistem informasi yang dikembangkan sudah berjalan dengan baik dan sesuai kebutuhan pengguna.

1.4 Manfaat Penelitian

Berlandaskan tujuan-tujuan yang telah dipaparkan, penelitian ini diharapkan untuk memberikan manfaat sebagai berikut:

1. Sistem informasi yang dirancang dan dibangun akan mempermudah pendaftaran santri baru sehingga lebih efisien dan mudah.
2. Dengan bantuan sistem informasi, data santri baru dapat dicatat dengan lebih akurat dan sistematis sehingga mengurangi kemungkinan terjadinya kesalahan manual.
3. Dengan adanya sistem pendaftaran santri diharapkan mengurangi waktu dan sumber daya yang diperlukan untuk proses pendaftaran.

1.5 Batasan Masalah

Penelitian ini berfokus pada perancangan dan pembangunan sistem informasi pendaftaran santri baru, dengan batasan masalah sebagai berikut :

1. Sistem informasi yang dikembangkan akan fokus pada pendaftaran santri baru di Islamic Center Al-Hazza.
2. Fitur yang terdapat di aplikasi ini dibatasi tidak sampai pada proses pembayaran.

1.6 Sistematika Penulisan

BAB I PENDAHULUAN

Menjelaskan mengenai latar belakang masalah, rumusan masalah, tujuan dan manfaat penelitian, batasan masalah dari ruang lingkup penelitian sistem informasi pendaftaran santri baru studi kasus di Islamic Center Al-Hazza

BAB II KAJIAN LITERATUR

Menjelaskan mengenai teori-teori yang mendukung serta berkaitan dengan proses pembangunan sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza yang relevan, serta posisi penelitian ini dalam konsep yang lebih luas.

BAB III METODOLOGI PENELITIAN

Menjelaskan mengenai tahapan penelitian, analisis masalah, metode pengumpulan data, perancangan dan pembangunan sistem informasi, hasil penelitian, *black box testing* dan *user acceptance test* (uat) dari pembangunan sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza.

BAB IV IMPLEMENTASI DAN EVALUASI

Tahap berfokus pada implementasi aplikasi atau sistem informasi pendaftaran santri baru berdasarkan perancangan serta tahap evaluasi sebagai penilaian untuk memastikan aplikasi/sistem berfungsi dengan baik dan memenuhi tujuan penelitian dan kebutuhan dari pesantren Islamic Center Al-Hazza

BAB V KESIMPULAN DAN SARAN

Menjelaskan mengenai kesimpulan akhir penelitian dan saran-saran untuk perbaikan proses pengembangan selanjutnya dari penelitian yg dilakukan.

BAB II KAJIAN LITERATUR

2.1 Sistem Informasi Pendaftaran Santri Baru

Sistem informasi pendaftaran santri baru adalah suatu sistem yang dirancang dan dikembangkan untuk memfasilitasi proses pendaftaran santri baru di sebuah lembaga pendidikan, seperti pesantren[2]. Sistem Informasi tersebut dapat digunakan oleh admin maupun calon peserta pendaftar untuk mempermudah proses pendaftaran menjadi lebih efisien dan efektif. Salah satu manfaatnya adalah mengurangi penggunaan kertas serta menyediakan kemudahan bagi admin dalam mengelola data penerimaan santri baru, serta memfasilitasi pendaftaran bagi calon santri dengan lebih mudah. Selain itu, sistem informasi ini juga memungkinkan informasi penerimaan santri baru dapat disebarkan dengan lebih cepat.

Bagi pihak administrasi atau pengelola Pesantren, sistem ini berguna untuk mengelola data pendaftaran santri baru, memantau status pendaftaran, dan mengatur proses seleksi atau verifikasi dokumen. Selain itu, sistem ini dapat menghasilkan laporan dan analisis mengenai jumlah pendaftar, tren pendaftaran dari waktu ke waktu, serta statistik lain yang dapat mendukung pengambilan keputusan[3]. Manfaat dari sistem informasi pendaftaran santri baru adalah mempercepat dan mempermudah proses pendaftaran, mengurangi kesalahan administrasi, serta meningkatkan efisiensi pengelolaan data. Selain itu, sistem ini juga dapat meningkatkan transparansi dan akses informasi bagi calon santri dan pihak administrasi. Secara keseluruhan, Sistem Informasi Pendaftaran Santri Baru merupakan alat yang penting dalam meningkatkan efektivitas dan efisiensi proses pendaftaran santri baru di lembaga pendidikan.

2.2 *Framework*

Framework merupakan sebuah kerangka kerja atau struktur yang telah dirancang secara sistematis untuk memudahkan proses pembuatan aplikasi. Tujuan utamanya adalah untuk memfasilitasi pengembangan software dengan memberikan kumpulan alat, *library*, dan aturan standar yang dapat digunakan kembali. Dengan demikian, *framework* tidak hanya mempercepat pembuatan aplikasi, tetapi juga memudahkan dalam melakukan modifikasi atau perubahan fitur dengan cepat dan efisien[4].

Keberadaan *framework* memungkinkan programmer untuk bekerja dengan lebih terstruktur dan fokus pada logika bisnis aplikasi tanpa harus memulai dari awal. Ini tentu saja menghemat waktu dan upaya dalam pengembangan *software*. Selain itu, dengan menggunakan *framework*, programmer dapat menghasilkan *source code* yang lebih rapi, terorganisir, dan mudah untuk dipahami, baik oleh *developer* lain maupun untuk pengembangan di masa depan.

Selain manfaat dalam pengembangan, *framework* juga mendukung prinsip penggunaan kembali kode program (*code reusability*) dalam pembuatan aplikasi lain yang memiliki kebutuhan atau karakteristik serupa[5]. Hal ini tentu saja memberikan nilai tambah dalam efisiensi dan produktivitas tim pengembangan. Secara keseluruhan, *framework* dapat dianggap sebagai fondasi atau dasar dalam pengembangan perangkat lunak yang membantu mengatasi berbagai kompleksitas dan tantangan yang mungkin dihadapi oleh *developer*. Dengan adanya *framework*, proses pengembangan *software* menjadi lebih terstruktur, efisien, dan produktif, sehingga menghasilkan aplikasi yang berkualitas tinggi dan mudah untuk dikembangkan atau diperbarui di masa mendatang.

2.3 Framework Laravel

Laravel adalah sebuah kerangka kerja (*framework*) dalam bahasa pemrograman PHP yang dirancang berdasarkan konsep *Model, View, Controller* (MVC) dan dirilis di bawah lisensi MIT. Laravel merupakan hasil pengembangan dari website berbasis MVP (*Minimum Viable Product*) yang awalnya ditulis menggunakan PHP. Tujuan utama pengembangan Laravel adalah untuk meningkatkan kualitas perangkat lunak, mengurangi biaya dalam pengembangan dan pemeliharaan sistem, serta meningkatkan produktivitas kerja dengan menyediakan ekspresi yang jelas dan meminimalkan waktu yang diperlukan dalam pengembangan aplikasi[6]. *Framework* Laravel memberikan sejumlah manfaat signifikan bagi para pengembang aplikasi *web*. Pertama, Laravel membantu meningkatkan produktivitas pengembangan dengan menyajikan beragam fitur dan alat bawaan yang mempercepat proses pembuatan aplikasi.

Selain itu, dengan mengadopsi pola desain *Model, View, Controller* (MVC), Laravel berhasil memisahkan logika bisnis dari antarmuka pengguna dan pengelolaan data, yang tentunya mempermudah pengaturan dan pemeliharaan kode[7]. Keamanan

juga menjadi fokus utama dalam Laravel, dengan menyediakan berbagai fitur keamanan bawaan yang efektif dalam melindungi aplikasi dari berbagai serangan, termasuk *SQL injection* dan *cross-site scripting*. Selain itu, dukungan yang kuat dari komunitas pengembang yang besar serta dokumentasi resmi yang lengkap membuat proses pengembangan dengan Laravel menjadi lebih lancar. Terakhir, Laravel didesain untuk menghadirkan skalabilitas yang baik, memungkinkan aplikasi yang dikembangkan dengan *framework* ini dapat ditingkatkan kapasitasnya sesuai dengan kebutuhan bisnis. Secara keseluruhan, Laravel adalah pilihan yang populer dan efektif untuk pengembangan aplikasi *web* modern dan aman.

2.4 Model View Controller (MVC)

Model View Controller (MVC), adalah sebuah pola atau teknik pemrograman yang dirancang untuk memisahkan tiga komponen kunci dalam pengembangan perangkat lunak. Ketiga komponen tersebut adalah *bisnis logic* (logika aplikasi), *data logic* (manajemen data), dan *presentation logic* (tampilan antarmuka pengguna). Dengan pendekatan ini, aplikasi dapat lebih mudah dikelola, dipahami, dan dikembangkan karena setiap komponen memiliki tanggung jawab yang jelas[8]. Dalam pola desain pengembangan perangkat lunak, atau design pattern, seperti MVC, terdapat deskripsi tentang bagaimana *class* dan objek saling berinteraksi untuk memecahkan masalah tertentu. Pola ini menyediakan struktur yang terstruktur dan terorganisir dalam menyusun komponen-komponen aplikasi, sehingga mempermudah proses pengembangan.

Salah satu keuntungan dari penerapan MVC adalah memudahkan *developer* dalam mengorganisir kode, memisahkan tanggung jawab antara desain aplikasi, manajemen data, dan tampilan pengguna. Hal ini memungkinkan untuk melakukan perubahan atau penyesuaian fitur dengan lebih efisien di masa mendatang, serta memudahkan dalam perawatan dan pengembangan aplikasi secara keseluruhan[9].

2.5 MySQL

MySQL adalah jenis Sistem Manajemen Basis Data Relasional (RDBMS) yang mendukung berbagai bahasa pemrograman termasuk PHP. Dengan menggunakan bahasa permintaan terstruktur SQL, MySQL memungkinkan pengguna untuk

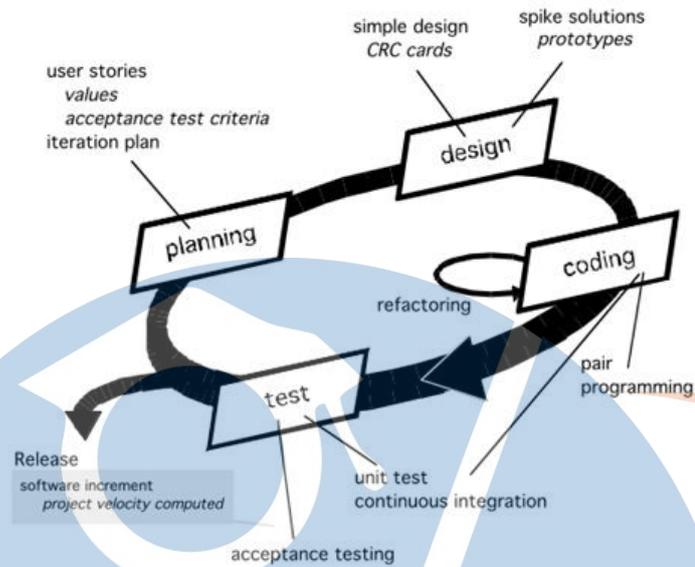
mengelola data dalam model relasional, di mana setiap tabel memiliki kolom dan baris yang mewakili atribut dan entitas data. Sebagai server RDBMS, MySQL berfungsi sebagai perangkat lunak yang menjalankan operasi pengelolaan dan akses ke basis data. Dengan demikian, MySQL adalah alat yang vital dalam pengembangan aplikasi web dan perangkat lunak yang membutuhkan pengelolaan data yang efisien dan handal[10].

MySQL menyajikan solusi penyimpanan data terstruktur dan handal bagi berbagai jenis aplikasi. Sebagai *platform open-source* yang terpercaya, MySQL menyediakan lingkungan yang dapat diandalkan dari situs *web* hingga aplikasi bisnis kompleks. Memanfaatkan bahasa SQL, pengguna dapat dengan mudah mengelola dan mengakses data. Selain itu, MySQL menonjol dengan kinerja tinggi yang mampu menangani volume data besar dengan cepat dan efisien. Fokusnya pada keamanan juga tampak, dengan berbagai fitur untuk melindungi data dari akses yang tidak sah. Secara keseluruhan, MySQL tetap menjadi pilihan yang populer dan andal untuk pengelolaan basis data dalam aplikasi modern.

2.6 Metode *Extreme Programming (XP)*

Extreme Programming (XP) adalah pendekatan dalam pengembangan perangkat lunak yang bertujuan untuk menyederhanakan proses tersebut agar lebih fleksibel dan adaptif[3]. Dengan mengedepankan prinsip-prinsip seperti pengujian terlebih dahulu, pengiriman berulang, dan pengkodean bersama, XP membantu meningkatkan efisiensi, kecepatan, dan ketepatan dalam pengembangan perangkat lunak, sambil mengurangi risiko.

Salah satu ciri utama XP adalah kecepatan, efisiensi, dan ketepatan dalam pengembangan. Meskipun demikian, dokumentasi formal seringkali tidak menjadi fokus utama dalam XP karena lebih menekankan pada interaksi langsung dengan klien atau pemangku kepentingan, terutama pada tahap perencanaan[11].



Gambar 2.1 Metode Extreme Programming

Sumber : www.extremeprogramming.org

Dengan demikian, XP menyajikan pendekatan yang kuat untuk pengembangan perangkat lunak yang adaptif dan fleksibel, namun tetap membutuhkan manajemen yang cermat terhadap dokumentasi dan komunikasi dengan klien untuk memastikan kesesuaian dengan kebutuhan bisnis dan pengguna akhir.

2.7 Unified Modeling Language

Unified Modeling Language (UML) adalah kumpulan standar teknik diagram yang memberikan representasi grafis yang kaya untuk model pengembangan sistem dari tahap analisis hingga implementasi. Saat ini, sebagian besar sistem yang dikembangkan menggunakan pendekatan berorientasi objek, dan UML menjadi alat yang umum digunakan untuk menggambarkan sistem yang sedang dikembangkan. UML menggunakan berbagai diagram untuk menggambarkan berbagai aspek dari sistem yang sedang dikembangkan, yang dapat dikelompokkan ke dalam dua klasifikasi utama: struktur dan perilaku. Diagram struktur mencakup *class*, *object*, *package*, *component*, *deployment*, dan *composite structure*, sementara diagram perilaku mencakup *activity*, *sequence*, *interaction overview*, *timing*, *communication*, *behavior state machine*, *protocol state machine*, dan *Use Case diagram*[12].

Dalam penerapan *Unified Modeling Language* UML, *Use Case Diagram* berfungsi untuk memvisualisasikan bagaimana pengguna berinteraksi dengan sistem. Diagram ini menggambarkan skenario interaksi antara aktor (pengguna atau sistem eksternal) dengan sistem, baik tindakan yang dilakukan oleh aktor maupun respons yang diberikan oleh sistem. Di sisi lain, *Activity Diagram* digunakan untuk menggambarkan alur proses atau fungsionalitas dalam sistem. Diagram ini memberikan gambaran yang jelas tentang langkah-langkah dalam suatu proses, termasuk keputusan, percabangan, dan interaksi antarproses, sehingga memudahkan pemahaman keseluruhan alur kerja dalam aplikasi[13].

2.1 Black Box Testing

Metode pengujian *black box testing* adalah salah satu pendekatan pengujian perangkat lunak yang fokus pada pengujian fungsionalitas tanpa memperhatikan detail internal atau struktur kode dari perangkat lunak tersebut. Dalam metode ini, pengujian hanya berkonsentrasi pada input dan output dari sistem, tanpa mengetahui bagaimana sistem tersebut bekerja di belakang layar. Proses pengujian *black box* dilakukan dengan memasukkan data ke dalam berbagai form yang ada dalam aplikasi, kemudian mengamati bagaimana sistem memproses data tersebut dan menghasilkan output[14].

Tujuan utama dari pengujian ini adalah untuk memastikan bahwa aplikasi berfungsi sesuai dengan alur (*flow*) yang telah direncanakan dan memenuhi persyaratan yang ditetapkan. Dengan demikian, *black box testing* membantu dalam mengidentifikasi bug atau kesalahan fungsional yang mungkin muncul saat pengguna berinteraksi dengan sistem. Pengujian ini juga berguna untuk memastikan bahwa semua fitur dan skenario penggunaan yang diharapkan telah terimplementasi dengan benar dan bahwa sistem dapat menangani berbagai jenis input dengan benar dan menghasilkan output yang sesuai.

2.9 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) adalah tahap krusial dalam pengembangan perangkat lunak yang dilakukan sebelum perangkat lunak itu dianggap siap untuk diluncurkan. Biasanya, UAT merupakan langkah terakhir dalam proses pengujian dan bertujuan untuk memverifikasi bahwa perangkat lunak telah memenuhi kebutuhan dan

harapan pengguna akhir. UAT sering juga disebut sebagai pengujian *beta*, pengujian aplikasi, atau pengujian pengguna akhir[15].

Pada tahap UAT, perangkat lunak diuji oleh pengguna akhir atau pihak yang berkepentingan langsung. Mereka akan menjalankan serangkaian skenario pengujian untuk memastikan bahwa perangkat lunak dapat beroperasi sesuai dengan kebutuhan bisnis atau pengguna. Hasil dari UAT menjadi penentu utama apakah perangkat lunak tersebut siap untuk dirilis ke atau masih memerlukan penyesuaian lebih lanjut.

2.10 Penelitian Terkait

Dalam proses penulisan tugas akhir ini, penulis terdorong dan terinspirasi oleh beberapa penelitian sebelumnya yang terkait dengan latar belakang masalah yang dibahas dalam tugas akhir ini. Beberapa penelitian yang relevan dengan topik tugas akhir ini meliputi:

Tabel 2.1 Penelitian Terkait

NO	Nama dan Tahun	Judul	Topik	Subjek	Hasil
1	Velis Dwi Cahyani, 2020	Perancangan Aplikasi Penerimaan Peserta Didik Baru (PPDB) Pada SMK Aditama Sidomulyo Berbasis <i>Web</i>	Perancangan Aplikasi PPDB	Peserta Didik Baru SMK Aditama Sidomulyo	Implementasi platform penerimaan siswa baru berbentuk situs <i>web</i> dirancang mendukung proses seleksi siswa yang diterima.
2	Efronius Paduansi, Sartika Lina Mulani Sitio, 2023	Implementasi Metode <i>Extreme Programming</i> Pada Perancangan Aplikasi PPDB	Implementasi Aplikasi PPDB Berbasis Android	Pihak Sekolah dan Calon Siswa	Aplikasi PPDB online berbasis android yang memudahkan pendaftaran siswa baru bagi calon

NO	Nama dan Tahun	Judul	Topik	Subjek	Hasil
		Online Berbasis Android			siswa dan administrasi/panitia.
3	Rian Ordila, Refni Wahyuni, Sukron Nasution, 2022	Rancang Sistem Penerimaan Santri Baru Pondok Pesantren Bahrul Ulum Berbasis <i>Web</i>	Rancang Sistem Penerimaan Santri Berbasis <i>Web</i>	Pihak Pesantren dan Orang tua/Calon Santri	Penerapan sistem informasi penerimaan santri baru secara online yang mempermudah orang tua/calon santri dalam proses pendaftaran santri baru.
4	Mohamad Rizki Hanif, 2023	Analisis dan Perancangan Sistem Informasi Pembayaran Sekolah Berbasis <i>Extreme Programming</i> Menggunakan <i>Framework MVC</i>	Sistem Informasi Pembayaran Sekolah	Tata Usaha SMK Walang Jaya	Aplikasi yang membantu administrasi sekolah dan siswa dalam proses pembayaran, sehingga meningkatkan efisiensi waktu pembayaran SPP.

Pada Tabel 2.1 Penelitian Terkait, Terdapat beberapa penelitian terkait Penerimaan Peserta Didik Baru yang telah dilakukan. Penelitian ini mengacu pada studi-studi sebelumnya yang berkaitan dengan topik yang sedang diteliti, membantu dalam memberikan dukungan serta kontekstualisasi terhadap penelitian yang sedang dilakukan.

2.11 Posisi Penelitian

Data penelitian ini mencerminkan variasi dalam teknologi yang digunakan, subjek penelitian yang berbeda serta metode pengujian yang digunakan, jika dibandingkan dengan penelitian sebelumnya.

Tabel 2.2 Posisi Penelitian

NO	Judul	Berbasis Web	PPDB	Laravel	Blackbox Testing	XP
1	Velis Dwi Cahyani, 2020 Perancangan Aplikasi Penerimaan Peserta Didik Baru (PPDB) Pada SMK Aditama Sidomulyo Berbasis <i>Web</i>	✓	✓			
2	Efronius Paduansi, Sartika Lina Mulani Sitio, 2023 Implementasi Metode <i>Extreme Programming</i> Pada Perancangan Aplikasi PPDB Online Berbasis Android		✓		✓	✓
3	Rian Ordila, Refni Wahyuni, Sukron Nasution, 2022 Rancang Sistem Penerimaan Santri Baru Pondok Pesantren Bahrul Ulum Berbasis <i>Web</i>	✓	✓	✓		
4	Mohamad Rizki Hanif, 2023 <i>Analisis dan Perancangan Sistem Informasi Pembayaran Sekolah Berbasis Extreme Programming Menggunakan Framework MVC</i>	✓		✓	✓	✓
5	Agung P S., 2024 Rancang Bangun Sistem Informasi Pendaftaran Santri Baru Menggunakan <i>Framework Laravel: Studi Kasus Islamic Center Al-Hazza</i>	✓	✓	✓	✓	✓

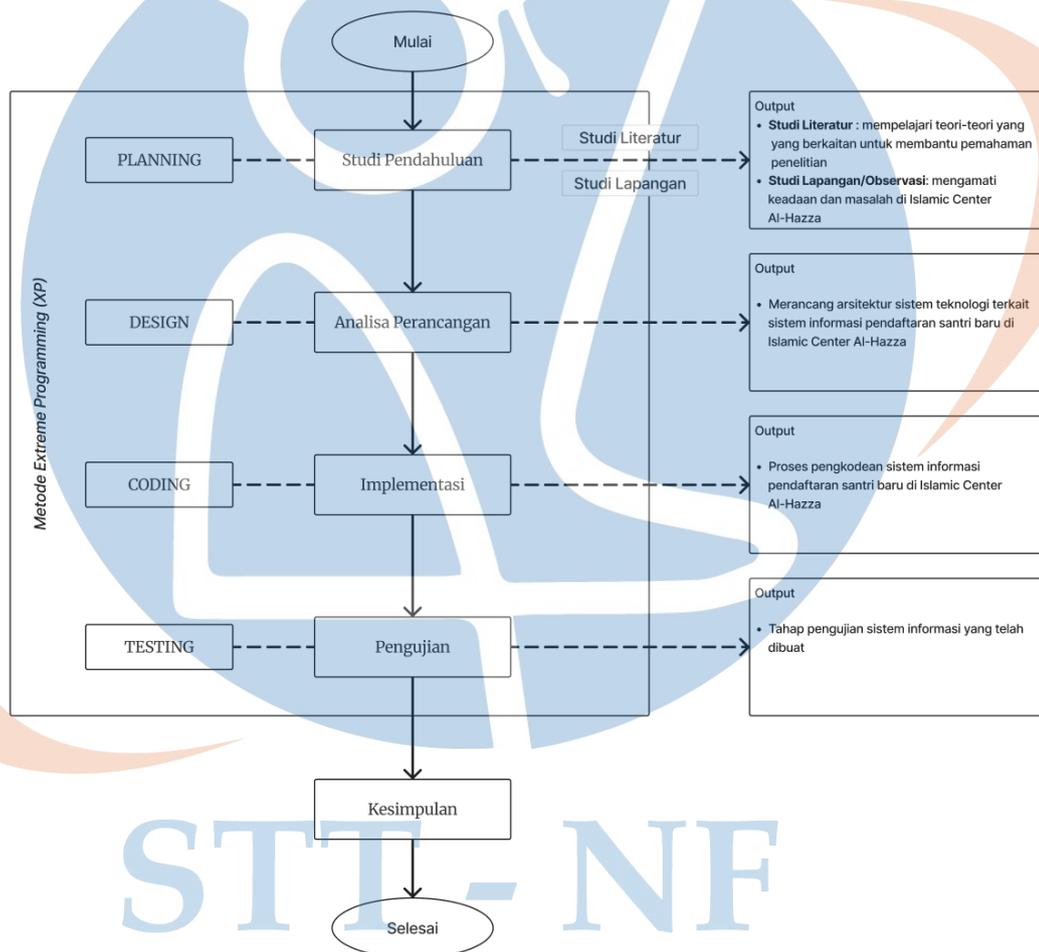
Pada Tabel 2.2 Posisi Penelitian, Beberapa penelitian memiliki kesamaan dalam hal tujuan, namun menggunakan pendekatan yang berbeda. Berikut adalah perbedaan masing-masing penelitian dengan penelitian terkait:

1. Penelitian Velis Dwi Cahyani menggunakan *php native* dan metode *waterfall* dalam merancang aplikasi penerimaan peserta didik baru berbasis website. Sedangkan penelitian pendaftaran santri baru menggunakan *framework Laravel* dengan metode *extreme programming* untuk merancang sistem informasi pendaftaran santri baru.
2. Penelitian Rian Ordilla, Refni Wahyuni, dan Sukron Nasution merancang sistem informasi penerimaan santri baru berbasis website menggunakan *framework Laravel* dengan metode *waterfall*. Perbedaan dengan penelitian pendaftaran santri baru adalah penggunaan metode *extreme programming* yang lebih iteratif dan fleksibel dibandingkan metode *waterfall* yang lebih linear dan berurutan.
3. Penelitian Efronius Paduansi dan Sartika Lina Mulani Sitio implementasi aplikasi PPDB Online menggunakan metode *extreme programming* dengan platform berbasis android. Perbedaan platform yang digunakan berbeda, dimana penelitian pendaftaran santri baru fokus pada aplikasi berbasis website menggunakan *framework Laravel*.
4. Penelitian Mohamad Rizki Hanif perancangan sistem informasi pembayaran sekolah menggunakan teknologi dan metode yang serupa. Perbedaannya dengan penelitian ini yaitu topik yang diangkat berbeda, dimana peneliti fokus pada pendaftaran santri baru sementara penelitian tersebut berfokus pada sistem informasi pembayaran sekolah.

BAB III METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Dalam penelitian ini, penulis mengikuti serangkaian tahapan yang sesuai dengan objektif penelitian. Setiap tahapan penelitian telah dirinci dengan cermat, mengikuti alur yang disusun berdasarkan metode *extreme programming*. Hal ini dilakukan agar penelitian dapat berjalan dengan terarah dan terorganisir.



Gambar 3.1 Tahapan Penelitian

3.1.1 Studi Pendahuluan

Dalam tahap studi pendahuluan ini, dilakukan penelitian literatur untuk memperoleh pemahaman mendalam tentang teknologi dan metode terbaru dalam pengembangan sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza.

Tujuannya adalah untuk mengidentifikasi teknologi yang paling cocok untuk mengatasi masalah dalam sistem pendaftaran santri baru tersebut. Selain itu, dilakukan juga observasi langsung terhadap sistem pendaftaran yang sudah ada, untuk mendapatkan wawasan yang lebih praktis dan menemukan solusi yang efektif. Observasi ini memberikan gambaran langsung tentang proses, kendala, dan potensi perbaikan yang bisa dilakukan dalam sistem pendaftaran santri baru di Islamic Center Al-Hazza.

3.1.2 Analisa Perancangan

Tahapan perancangan ini merupakan proses krusial dalam pengembangan sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza. Dalam tahap ini, dilakukan perancangan arsitektur sistem yang mencakup pemilihan teknologi, struktur database, dan alur kerja aplikasi. Salah satu indikator capaian utama adalah pemilihan penggunaan *framework* Laravel sebagai kerangka kerja dalam pengembangan sistem. *Framework* Laravel dipilih karena reputasinya dalam menyediakan struktur yang kuat dan dukungan komunitas yang luas, sehingga dapat mempercepat proses pengembangan dan meningkatkan kualitas sistem yang dihasilkan. Selain itu, dalam perancangan ini juga mempertimbangkan aspek keamanan, skalabilitas, dan user experience agar sistem dapat berfungsi dengan optimal dan memenuhi kebutuhan pengguna.

3.1.3 Implementasi

Dalam tahap ini, peneliti memulai dengan pembuatan kode pemrograman untuk sistem informasi pendaftaran santri baru. Fokus utama pada tahap ini adalah konfigurasi *framework laravel*, implementasi sistem pendaftaran santri, pengembangan algoritma yang mendukung proses yang diinginkan, serta pengembangan fitur-fitur esensial dalam sistem pendaftaran santri baru di Islamic Center Al-Hazza.

3.1.4 Pengujian

Tahap pengujian dilakukan terhadap sistem yang telah dirancang dengan melakukan serangkaian uji coba terhadap fungsionalitas sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza. Aplikasi ini mengimplementasikan sejumlah

aktivitas untuk memastikan bahwa tujuan penelitian telah terpenuhi melalui pengujian fungsional. Uji coba ini bertujuan untuk menjamin kinerja dan keandalan sistem sebelum diterapkan secara luas.

3.1.5 Kesimpulan

Langkah akhir melibatkan penyusunan kesimpulan berdasarkan hasil penelitian dan pengembangan. Peneliti akan menguraikan solusi yang telah diterapkan untuk mengatasi tantangan dalam pendaftaran santri baru di Islamic Center Al-Hazza. Dalam kesimpulan ini, akan dievaluasi efektivitas solusi, pencapaian tujuan yang telah ditentukan, serta dampak dari implementasi solusi terhadap permasalahan yang ada.

3.2 Rancangan Penelitian

Rancangan penelitian merinci rencana yang akan dijalankan dalam penelitian ini. Dalam rancangan penelitian akan dijelaskan jenis penelitian yang akan dilakukan, metode analisis yang digunakan, serta metode pengumpulan data yang dipilih.

3.2.1 Jenis Penelitian

Jenis penelitian yang akan dilakukan dalam penelitian ini adalah penelitian pengembangan atau disebut juga *research and development (R&D)*. Penelitian pengembangan berfokus pada pengembangan produk, sistem, atau teknologi untuk memberikan solusi konkret terhadap masalah yang dihadapi Islamic Center Al-Hazza. Dalam konteks ini, fokusnya adalah pada Sistem informasi pendaftaran santri baru menggunakan *Framework Laravel*. Penelitian ini melibatkan perancangan, pengembangan, dan pengujian teknologi untuk mengatasi tantangan yang dihadapi oleh Islamic Center Al-Hazza. Hasil dari penelitian ini meliputi :

1. Pengembangan sistem informasi yang mempermudah proses pendaftaran santri baru di Islamic Center Al-Hazza.
2. Penerapan teknologi *Framework Laravel* sebagai dasar dalam pengembangan untuk menjamin performa dan kestabilan sistem.
3. Evaluasi akan dilakukan untuk menilai sejauh mana penerapan *Framework Laravel* dalam pengembangan sistem informasi berhasil memudahkan proses pendaftaran santri baru di Islamic Center Al-Hazza. Hasil evaluasi ini akan menggambarkan seberapa efektif solusi yang telah dikembangkan.

3.2.2 Metode Analisis Data

Metode analisis data dalam penelitian ini adalah metode kuantitatif, yang dipilih berdasarkan pertimbangan yang matang. Metode ini memanfaatkan teknik survei dan analisis statistik untuk mendapatkan tingkat keberhasilan dalam penggunaan sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza. Metode kuantitatif memungkinkan pengukuran dan analisis data numerik secara objektif, serta mengidentifikasi aspek-aspek yang dapat diukur secara langsung melalui kuesioner.

Melalui pendekatan kuantitatif, peneliti dapat memperoleh pemahaman yang lebih terukur dan jelas tentang bagaimana pengguna merasakan, berinteraksi, dan memanfaatkan sistem informasi pendaftaran santri baru yang diteliti, serta faktor-faktor yang memengaruhi tingkat keberhasilan fungsional sistem informasi pendaftaran santri baru secara keseluruhan.

3.2.3 Metode Pengumpulan Data

Metode pengumpulan data dalam penelitian ini melibatkan teknik observasi, di mana peneliti melakukan pengamatan langsung di lokasi untuk mengamati proses dan aktivitas terkait. Selain observasi, penulis juga menggunakan kuisisioner untuk mengumpulkan tanggapan dan pendapat dari pihak Islamic Center Al-Hazza mengenai sistem yang ada. Wawancara juga dilakukan baik secara tatap muka maupun online untuk mendapatkan informasi mendalam dari para pihak pesantren.

Teknik observasi memberikan gambaran langsung mengenai situasi dan proses yang berlangsung di lapangan, sementara kuisisioner dan wawancara memberikan perspektif dari pengguna. Setelah semua data terkumpul, informasi yang diperoleh akan menjadi dasar untuk membantu penulis dalam merancang dan mengembangkan sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza. Dengan demikian, metode pengumpulan data ini dirancang untuk memastikan informasi yang diperoleh sekomprehensif mungkin untuk mendukung proses pengembangan sistem yang efektif dan sesuai kebutuhan.

3.2.4 Metode Pengujian

Pada tahap ini, Pengujian menggunakan *User Acceptance Testing (UAT)* yang fokus pada evaluasi kemampuan aplikasi dalam memfasilitasi proses pendaftaran

santri baru di Islamic Center Al-Hazza dari sisi pengguna baik dari perspektif calon santri maupun tim administrasi. Dengan menerapkan metode *User Acceptance Testing (UAT)*, pengguna memiliki kesempatan untuk memberikan umpan balik secara langsung terhadap kinerja dan kemudahan penggunaan sistem informasi pendaftaran santri baru. Hal ini bertujuan untuk memastikan bahwa hasil akhir sesuai dengan ekspektasi dan memenuhi standar sesuai kebutuhan. Berikut langkah-langkah metode *User Acceptance Testing (UAT)* yang dilakukan di Islamic Center Al-Hazza:

1. Perencanaan UAT: Menentukan ruang lingkup, tujuan, dan rencana pengujian.
2. Persiapan Lingkungan: Menyiapkan lingkungan pengujian dengan data dan konfigurasi yang sesuai.
3. Pembuatan Skenario: Mengidentifikasi dan membuat test case berdasarkan fungsionalitas sistem informasi pendaftaran santri baru.
4. Pelaksanaan UAT: Melakukan pengujian sesuai dengan skenario yang telah ditentukan.
5. Pengumpulan dan Analisis Hasil: Mengumpulkan umpan balik dan menganalisis data pengujian.
6. Pembuatan Laporan UAT: Menyusun laporan hasil pengujian dan rekomendasi perbaikan.

Tabel 3.1 Tabel Pengujian User Acceptance Test

NO	Pengujian	Hasil yang diharapkan	Kesimpulan
1	Sistem informasi pendaftaran santri baru mudah diakses	Sistem informasi pendaftaran santri baru mudah diakses dan dijangkau oleh para calon santri	Sesuai atau Tidak Sesuai
2	Fitur yang tersedia sesuai dengan kebutuhan pengguna	Segala fitur yang tersedia sesuai dengan kebutuhan pengguna	Sesuai atau Tidak Sesuai
3	Tampilan sistem informasi pendaftaran santri baru memiliki	Sistem informasi yang dibangun memiliki tampilan yang menarik dan <i>user-friendly</i> untuk menarik	Sesuai atau Tidak Sesuai

	tampilan yang menarik dan <i>user-friendly</i>	calon santri baru dan mudah di gunakan	
4	Segala fitur yang tersedia berjalan dengan baik	Fitur-fitur yang dibangun dapat berjalan sesuai dengan fungsinya	Sesuai atau Tidak Sesuai
5	Sistem informasi yang dibangun berjalan dengan baik dan sesuai dengan harapan	Sistem informasi pendaftaran santri baru dapat berjalan dengan baik dan sesuai harapan pesantren Islamic Center Al-Hazza	Sesuai atau Tidak Sesuai

Dengan langkah-langkah ini, UAT dapat memastikan bahwa Sistem Informasi Pendaftaran Santri Baru di Islamic Center Al-Hazza memenuhi standar dan ekspektasi yang ditetapkan.

3.2.5 Metode Implementasi dan Evaluasi

Dalam mengimplementasikan Sistem Informasi Pendaftaran Santri Baru di Islamic Center Al-Hazza, pendekatan yang digunakan adalah *Extreme Programming (XP)*. Dengan XP memungkinkan perubahan dan penyesuaian yang cepat berdasarkan kebutuhan pengguna. Sementara dalam evaluasi, metode *User Acceptance Testing (UAT)* menjadi pendekatan yang efektif. Dengan UAT, pengguna melakukan pengujian langsung terhadap fitur-fitur yang telah dikembangkan untuk memastikan sesuai dengan kebutuhan mereka. Umpan balik dari UAT menjadi dasar untuk evaluasi.

Dengan menggabungkan pendekatan *Extreme Programming* dalam implementasi dan UAT dalam evaluasi, diharapkan Sistem Informasi Pendaftaran Santri Baru di Islamic Center Al-Hazza dapat dikembangkan dengan cepat, efisien, dan sesuai dengan kebutuhan pengguna.

3.2.6 Lingkungan Pengembangan

Lingkungan pengembangan adalah tempat di mana penelitian atau pengembangan sistem informasi pendaftaran santri baru berlangsung dan termasuk semua peralatan, software, serta sumber daya yang digunakan oleh penulis untuk

merancang, menguji, dan menyusun kode. Ini melibatkan semua komponen yang diperlukan dalam seluruh proses pengembangan sistem informasi pendaftaran santri baru, mulai dari tahap awal hingga penyelesaiannya.

1. Lokasi Penelitian

Penelitian ini berfokus di Islamic Center Al-Hazza karena saat ini belum ada sistem informasi pendaftaran santri baru yang tersedia. Dengan harapan, penelitian ini memberikan kontribusi yang berarti untuk mempermudah dan mempercepat proses pendaftaran santri baru ke pesantren. Ini merupakan langkah maju untuk mendukung efisiensi dan efektivitas manajemen pesantren dalam menerima santri baru dengan lebih terstruktur dan sistematis.

2. Alat dan Bahan

Berbagai perangkat keras dan perangkat lunak digunakan dalam penelitian untuk merancang dan membangun sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza adalah sebagai berikut :

1. Kebutuhan Alat

- a. ASUS ROG Zephyrus M16
- b. Microsoft Windows 11 Pro
- c. RAM 40GB
- d. 11th Gen Intel(R) Core(TM) i9-11900H @ 2.50GHz

2. Kebutuhan Bahan

- a. Text Editor : Visual Studio Code 1.87.2
- b. PHP Version : PHP 8.2.12
- c. Laravel Version : Laravel 10
- d. Database : MySQL
- e. Desain Diagram : Figma

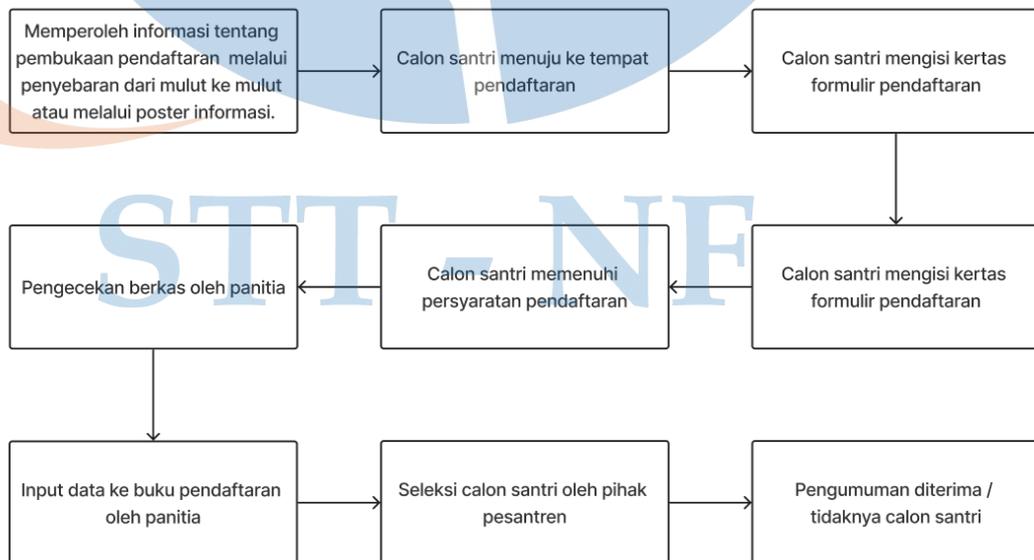
BAB IV IMPLEMENTASI DAN EVALUASI

4.1 Analisis dan Perencanaan Sistem

Pada tahap analisis dan perencanaan sistem, penelitian difokuskan pada masalah pendaftaran santri baru di Islamic Center Al-Hazza yang masih menggunakan metode tradisional. Sebagai langkah awal, dilakukan analisis kebutuhan secara mendalam untuk memahami berbagai tantangan yang ada. Berdasarkan analisis tersebut, diusulkan solusi inovatif berupa pengembangan Sistem Informasi Pendaftaran Santri Baru dengan menggunakan *framework Laravel*.

4.1.1 Analisis Sistem Berjalan

Proses pendaftaran santri baru di Islamic Center Al-Hazza saat ini masih dilakukan secara manual, di mana calon santri dan orang tua harus datang langsung ke pesantren dan mengisi formulir fisik dengan data pribadi, riwayat pendidikan, dan keahlian. Proses ini memiliki beberapa kendala, seperti kesulitan bagi calon santri dan orang tua yang tinggal jauh atau memiliki keterbatasan waktu. Selain itu, pengolahan data manual rentan terhadap kesalahan dan memerlukan waktu lebih lama untuk pengarsipan dan verifikasi.



Gambar 4.1 Alur Pendaftaran Santri Baru Saat Ini

Pada Gambar 4.1 alur pendaftaran santri baru di Islamic Center Al-Hazza dimulai dengan calon santri memperoleh informasi tentang pembukaan pendaftaran. Selanjutnya, calon santri menuju lokasi pendaftaran untuk mengisi formulir dan melengkapi berkas persyaratan. Setelah itu, panitia melakukan pengecekan berkas dan penginputan data calon santri ke dalam buku pendaftaran. Proses berikutnya adalah seleksi calon santri yang dilakukan oleh pihak pesantren, yang kemudian akan memberikan keputusan diterima atau tidaknya calon santri.

Proses pendaftaran santri baru di Islamic Center Al-Hazza saat ini memiliki beberapa kekurangan yang signifikan. Pengisian formulir dan pengecekan berkas secara manual memakan waktu dan rentan terhadap kesalahan manusia. Selain itu, calon santri dan orang tua harus datang langsung ke lokasi pendaftaran, yang mungkin merepotkan terutama bagi mereka yang tinggal jauh. Penginputan data ke dalam buku pendaftaran secara manual juga tidak efisien, menyulitkan dalam pengarsipan dan pencarian data. Oleh karena itu, diperlukan sebuah sistem informasi pendaftaran santri baru yang modern untuk mengatasi masalah-masalah ini, membuat proses pendaftaran menjadi lebih cepat, mudah, dan akurat.

4.1.2 Analisis Permasalahan dan Usulan Perbaikan

Dalam upaya meningkatkan efisiensi dan efektivitas proses pendaftaran santri baru di Islamic Center Al-Hazza, penting untuk terlebih dahulu mengidentifikasi berbagai permasalahan yang ada dalam sistem pendaftaran saat ini. Berikut adalah analisis permasalahan dan usulan perbaikan yang dapat diterapkan:

1. Analisis Permasalahan
 - a. Proses Manual yang Memakan Waktu
Pengisian formulir dan pengecekan berkas secara manual menyebabkan proses pendaftaran menjadi lambat dan tidak efisien.
 - b. Ketidaknyamanan bagi Calon Santri dan Orang Tua
Calon santri dan orang tua harus hadir langsung di lokasi pendaftaran, yang merepotkan terutama bagi yang tinggal jauh atau memiliki keterbatasan waktu.
 - c. Informasi Terbatas

Informasi mengenai pembukaan pendaftaran hanya disebarakan melalui mulut ke mulut atau poster, sehingga tidak semua calon santri dapat menjangkaunya.

d. Pencatatan Data yang Tidak Efisien

Penginputan data secara manual ke dalam buku pendaftaran meningkatkan risiko kesalahan pencatatan dan menyulitkan proses pengarsipan serta pencarian data.

2. Usulan Perbaikan

Sebagai solusi untuk mengatasi permasalahan tersebut, diusulkan implementasi sistem informasi pendaftaran santri baru. Sistem ini memungkinkan calon santri mengisi formulir dan mengunggah berkas secara online, mempercepat dan mempermudah proses pendaftaran. Informasi pendaftaran akan disebarakan melalui website, menjangkau lebih banyak calon santri. Memudahkan proses pengecekan dan penginputan data mengurangi kesalahan manusia dan meningkatkan efisiensi. Sistem ini juga mempercepat proses seleksi dan memudahkan akses pendaftaran dari mana saja, mengurangi ketidaknyamanan bagi calon santri dan orang tua yang berdomisili jauh dari tempat pendaftaran.

4.1.3 Analisis Kebutuhan Sistem

Untuk mengembangkan sistem informasi pendaftaran santri baru yang efektif di Islamic Center Al-Hazza, perlu dilakukan analisis kebutuhan sistem. Analisis ini bertujuan untuk mengidentifikasi berbagai kebutuhan fungsional dan non-fungsional yang harus dipenuhi oleh sistem agar dapat mengatasi permasalahan yang ada dan memenuhi harapan pengguna.

1. Kebutuhan Fungsional

a. Formulir Pendaftaran Online

Sistem harus menyediakan formulir pendaftaran yang dapat diisi secara online oleh calon santri, mencakup data pribadi, riwayat pendidikan, dan keahlian.

b. Unggah Berkas

Calon santri harus dapat mengunggah berkas persyaratan seperti foto, ijazah, dan dokumen pendukung lainnya melalui sistem.

c. Notifikasi dan Pengumuman

Sistem harus mampu mengirim notifikasi melalui email atau SMS kepada calon santri tentang status pendaftaran dan pengumuman hasil seleksi.

d. Manajemen Data Santri

Sistem harus menyediakan fitur untuk mengelola data calon santri, termasuk pencatatan, pengarsipan, dan pencarian data.

e. Proses Seleksi Otomatis

Sistem harus mendukung proses seleksi otomatis berdasarkan kriteria yang telah ditentukan oleh pihak pesantren.

f. Dashboard Admin

Sistem harus memiliki dashboard untuk admin yang memudahkan pengelolaan pendaftaran, pengecekan berkas, dan monitoring proses seleksi.

2. Kebutuhan Fungsional

1. Keamanan Data

Sistem harus memiliki mekanisme keamanan yang kuat untuk melindungi data pribadi calon santri dan mencegah akses tidak sah.

2. Skalabilitas

Sistem harus mampu menangani jumlah pendaftar yang banyak tanpa menurunkan kinerja.

3. Ketersediaan

Sistem harus dapat diakses secara online 24/7 agar calon santri dapat mendaftar kapan saja.

4. User-Friendly

Antarmuka sistem harus mudah digunakan oleh pengguna, termasuk calon santri dan orang tua yang mungkin tidak terbiasa dengan teknologi.

5. Kompatibilitas

Sistem harus kompatibel dengan berbagai perangkat dan browser untuk memastikan aksesibilitas yang luas.

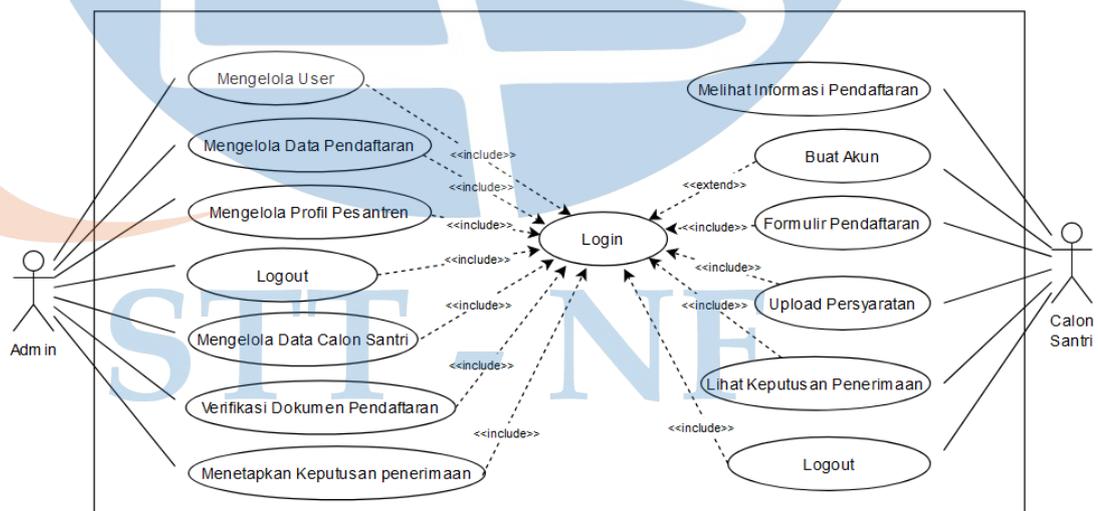
Dengan mengidentifikasi kebutuhan-kebutuhan ini, sistem informasi pendaftaran santri baru dapat dirancang dan dikembangkan secara optimal untuk meningkatkan efisiensi, kenyamanan, dan keandalan proses pendaftaran di Islamic Center Al-Hazza.

4.2 Perancangan Sistem

Perancangan sistem merupakan langkah krusial dalam pengembangan perangkat lunak, yang bertujuan untuk memastikan bahwa solusi yang diusulkan dapat memenuhi kebutuhan pengguna dan mengatasi permasalahan yang telah diidentifikasi di Islamic Center Al-Hazza. Melalui perancangan yang matang dan sistematis dengan menggunakan *Unified Modeling Language (UML)*, diharapkan sistem informasi ini dapat diimplementasikan dengan efektif dan efisien, memberikan manfaat maksimal bagi seluruh pengguna.

4.2.1 Design Use Case Diagram

Use Case Diagram untuk sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza adalah alat visual yang digunakan untuk menggambarkan interaksi antara pengguna dengan sistem, serta fungsionalitas utama yang disediakan oleh sistem tersebut. Dengan menggunakan *Use Case Diagram* dapat mengidentifikasi dan memodelkan skenario-skenario utama dalam proses pendaftaran santri baru di Islamic Center Al-Hazza, sehingga memastikan bahwa sistem dapat memenuhi kebutuhan dan ekspektasi pengguna secara efektif.



Gambar 4.2 Use Case Diagram

Dalam Gambar 4.2 Pada use case diagram ini, digambarkan interaksi antara pengguna dan sistem informasi pendaftaran santri baru yang dikembangkan menggunakan Framework Laravel. Diagram ini mencakup tiga aktor utama: admin dan calon santri.

1. Admin

Bertanggung jawab untuk mengelola data calon santri, mengelola data pendaftaran, memverifikasi dokumen pendaftaran, mengelola profil pesantren, serta menetapkan keputusan penerimaan santri.

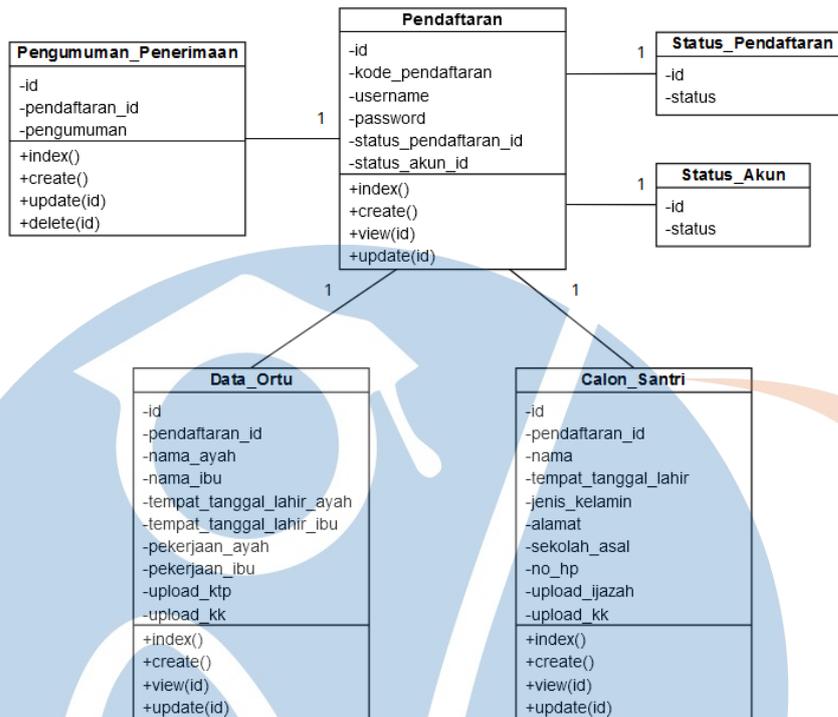
2. Calon Santri

Dapat melihat informasi pendaftaran, membuat akun pendaftaran, mengisi formulir pendaftaran, mengunggah persyaratan pendaftaran, dan melihat keputusan penerimaan santri.

Diagram ini memvisualisasikan bagaimana setiap aktor berinteraksi dengan sistem untuk menyelesaikan tugas-tugas mereka masing-masing. Dengan adanya sistem ini, proses pendaftaran santri menjadi lebih terstruktur dan efisien, mengurangi kesalahan dan mempercepat waktu pendaftaran. Selain itu, sistem ini juga memastikan bahwa semua data yang diperlukan tersimpan dengan baik dan mudah diakses.

4.2.2 Design Class Diagram

Class diagram adalah gambaran visual dari struktur dan interaksi antar kelas dalam sebuah sistem. Dalam konteks sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza, class diagram digunakan untuk mengilustrasikan bagaimana berbagai komponen atau kelas dalam sistem saling terhubung dan berinteraksi satu sama lain.



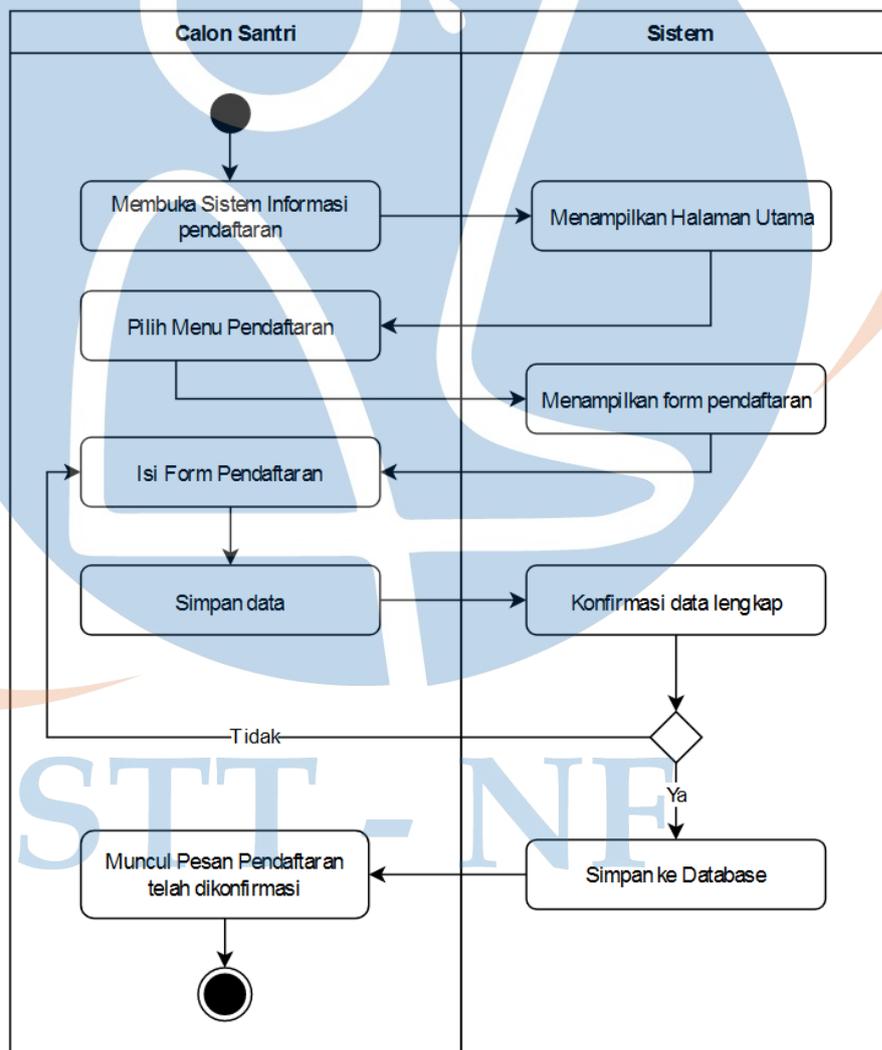
Gambar 4.3 Class Diagram

Pada Gambar 4.3 Class Diagram untuk sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza terdiri dari beberapa kelas yang saling berhubungan untuk mendukung keseluruhan proses pendaftaran. Berikut penjelasan setiap kelas:

1. *Class Pendaftaran*: Berisi informasi akun pendaftaran bagi calon santri, termasuk data login dan kode pendaftaran.
2. *Class Calon_Santri*: Menyimpan informasi biodata dari calon santri, seperti nama, tanggal lahir, alamat, dan lainnya.
3. *Class Data_Ortu*: Menyimpan informasi mengenai orang tua calon santri, termasuk nama dan pekerjaan.
4. *Class Status_Pendaftaran* : Terhubung dengan *class Pendaftaran*, menyimpan status proses pendaftaran, seperti "lulus" atau "tidak lulus".
5. *Class Status_Akun*: Terhubung dengan *class Pendaftaran*, berisi status aktif atau tidaknya akun calon santri.
6. *Class Pengumuman_Penerimaan*: Berelasi dengan *class Pendaftaran*, berisi informasi mengenai kelulusan atau tidaknya calon santri dalam proses pendaftaran.

4.2.3 Design Activity Diagram

Activity Diagram adalah jenis diagram UML yang digunakan untuk memvisualisasikan urutan alur kerja atau serangkaian kegiatan dalam suatu proses. Diagram ini memberikan gambaran yang jelas tentang langkah-langkah yang terlibat dalam suatu tindakan atau aktivitas, serta hubungan antara aktivitas-aktivitas tersebut. Dalam konteks sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza digunakan untuk menggambarkan alur kerja atau aktivitas dalam sistem, menunjukkan urutan langkah-langkah dan keputusan yang terjadi dalam proses pendaftaran.

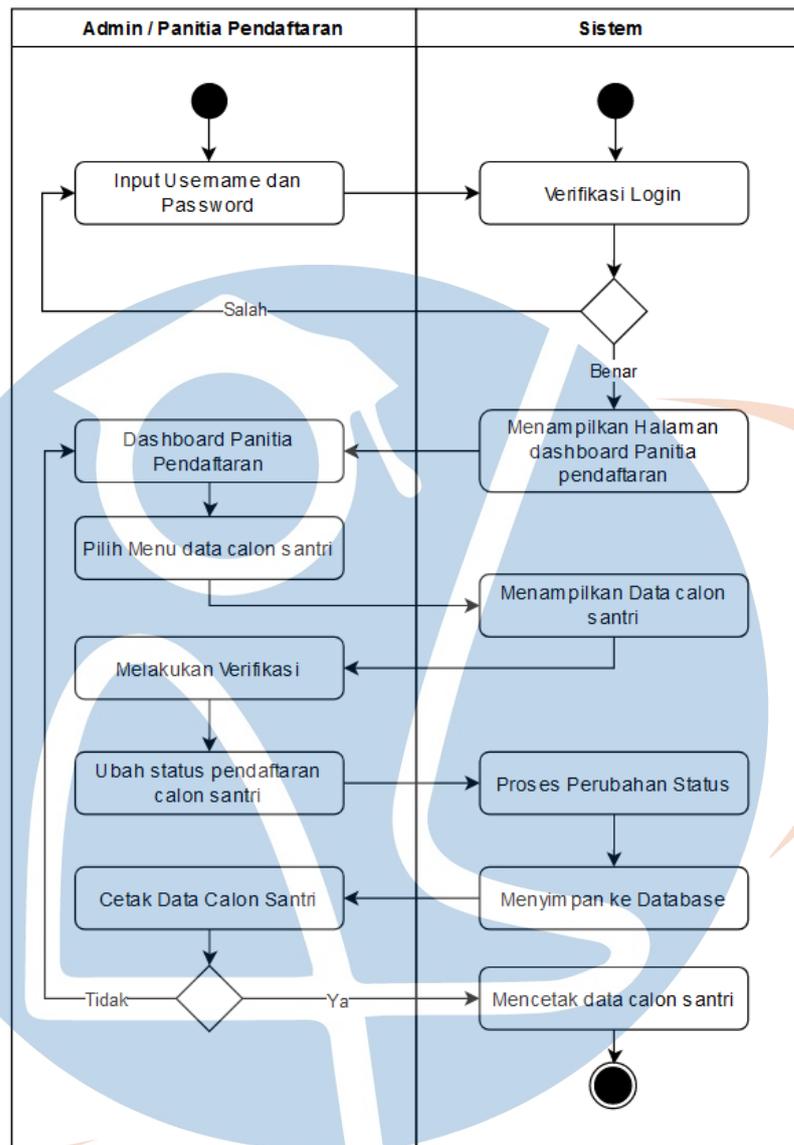


Gambar 4.4 Activity Diagram Calon Santri Melakukan Pendaftaran

Pada gambar 4.4 *activity diagram* calon santri melakukan pendaftaran pendaftaran menggambarkan alur langkah demi langkah yang diikuti calon santri dalam melakukan pendaftaran di sistem informasi pendaftaran santri baru.

1. Membuka Sistem Informasi Pendaftaran: Calon santri mengakses website sistem informasi pendaftaran santri baru melalui browser mereka.
2. Menampilkan Halaman Utama: Sistem menampilkan antarmuka utama yang ramah pengguna dari sistem informasi pendaftaran santri baru.
3. Pilih Menu Pendaftaran: Calon santri memilih opsi pendaftaran dari menu navigasi untuk memulai proses pendaftaran.
4. Menampilkan Form Pendaftaran: Sistem menampilkan formulir pendaftaran yang harus diisi oleh calon santri.
5. Isi Form Pendaftaran: Calon santri mengisi formulir pendaftaran dengan data pribadi, riwayat pendidikan, dan informasi lain yang diperlukan. Pada tahap ini, calon santri juga membuat akun pendaftaran.
6. Simpan Data: Setelah mengisi formulir, calon santri menekan tombol "Simpan Data" untuk mengirimkan informasi pendaftaran mereka.
7. Konfirmasi Data Lengkap?: Sistem secara otomatis memeriksa kelengkapan data yang telah diinput oleh calon santri. Jika semua data lengkap, proses berlanjut; jika ada data yang kurang, sistem mengarahkan calon santri kembali ke formulir untuk melengkapi data yang kurang.
8. Simpan ke Database: Informasi pendaftaran yang telah lengkap disimpan ke dalam database.
9. Muncul Pesan Pendaftaran Telah Dikonfirmasi: Sistem menampilkan pesan konfirmasi kepada calon santri bahwa pendaftaran mereka telah berhasil dan semua data telah tersimpan dengan baik.

Diagram ini mempermudah pemahaman tentang proses pendaftaran, mengurangi potensi kesalahan, dan memastikan bahwa semua data pendaftaran calon santri tersimpan dengan baik dan aman.



Gambar 4.5 Activity Diagram Verifikasi Data oleh Panitia Pendaftaran atau Admin

Pada Gambar 4.5 Activity diagram untuk proses verifikasi data oleh panitia pendaftaran atau admin memberikan pandangan secara visual tentang alur proses verifikasi data dalam sistem informasi pendaftaran santri baru.

1. Input Username dan Password: Panitia atau admin melakukan login dengan memasukkan username dan password.
2. Verifikasi Login: Sistem memverifikasi kredensial login. Jika login berhasil, sistem menampilkan halaman dashboard panitia pendaftaran. Jika gagal, sistem kembali ke menu login.

3. Pilih Menu Data Calon Santri: Admin atau panitia memilih menu data calon santri dari dashboard.
4. Menampilkan Data Calon Santri: Sistem menampilkan data calon santri yang telah terdaftar.
5. Melakukan Verifikasi Data: Panitia memeriksa dan memverifikasi data yang telah diinput oleh calon santri, memastikan keakuratan dan kelengkapan informasi.
6. Ubah Status Pendaftaran Calon Santri: Setelah verifikasi, panitia mengubah status pendaftaran calon santri, misalnya dari "Menunggu Verifikasi" menjadi "Lulus" atau "Tidak Lulus".
7. Proses Perubahan Status: Sistem memproses perubahan status pendaftaran calon santri sesuai dengan keputusan panitia.
8. Menyimpan ke Database: Sistem menyimpan data calon santri yang statusnya telah diubah ke dalam database.
9. Cetak Data Calon Santri: Admin atau panitia memiliki opsi untuk mencetak data calon santri. Jika memilih untuk mencetak, sistem melakukan proses pencetakan data. Jika tidak, sistem kembali ke dashboard panitia pendaftaran.

Activity diagram ini memberikan gambaran yang jelas tentang langkah-langkah yang diambil oleh panitia atau admin dalam memverifikasi data calon santri, mengubah status pendaftaran, dan menyimpan informasi tersebut ke dalam database. Diagram ini memastikan proses verifikasi berjalan dengan baik dan efisien, serta memberikan fleksibilitas untuk mencetak data jika diperlukan.

STT - NF

4.3 Implementasi Sistem Pendaftaran Santri Baru

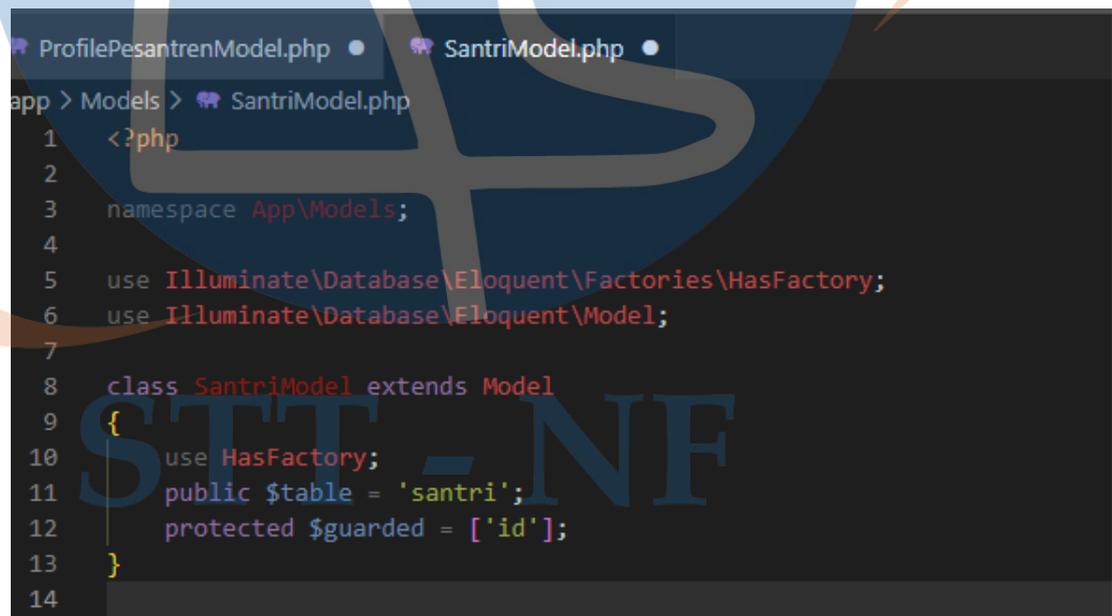
Implementasi sistem pendaftaran santri baru adalah proses penerapan teknologi dan pengkodean untuk sistem informasi di Islamic Center Al-Hazza. Langkah ini merupakan realisasi dari perancangan yang telah dibuat menjadi sistem informasi pendaftaran santri baru yang dikembangkan menggunakan Framework Laravel.

4.3.1 Implementasi Kode

Tahap Implementasi kode dalam pengembangan Sistem Informasi Pendaftaran Santri Baru di Islamic Center Al-Hazza. Dengan menggunakan Framework Laravel, bab ini menguraikan langkah-langkah pengkodean untuk setiap fitur, integrasi komponen sistem, serta teknik MVC yang digunakan.

1. Model

Model merupakan bagian dari arsitektur MVC yang menangani interaksi dengan database. Pada sistem ini, *Model* digunakan untuk mengelola data santri, pendaftaran, detail pendaftaran, dan Profil Pesantren.



```
ProfilePesantrenModel.php • SantriModel.php •
app > Models > SantriModel.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class SantriModel extends Model
9  {
10     use HasFactory;
11     public $table = 'santri';
12     protected $guarded = ['id'];
13 }
14
```

Gambar 4.6 Model Santri

Pada gambar 4.5, *Model* Santri menunjukkan koneksi ke database dengan tabel santri. *Model* ini berperan penting dalam mengelola data yang terkait dengan calon santri.

```
PendaftaranModel.php X
app > Models > PendaftaranModel.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class PendaftaranModel extends Model
9  {
10     use HasFactory;
11     public $table = 'pendaftaran';
12     protected $guarded = ['id'];
13
14     public function detail_pendaftaran() {
15         return $this->hasMany(PendaftaranDetailModel::class, 'pendaftaran_id', 'id');
16     }
17 }
18
```

Gambar 4.7 Model Pendaftaran

Pada gambar 4.7, Model Pendaftaran menunjukkan koneksi ke database dengan tabel pendaftaran. Model ini berperan dalam mengelola data yang terkait dengan proses pendaftaran santri baru.

```
app > Models > PendaftaranDetailModel.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class PendaftaranDetailModel extends Model
9  {
10     use HasFactory;
11     public $table = 'pendaftaran_detail';
12     protected $guarded = ['id'];
13
14     public function pendaftaran()
15     {
16         return $this->belongsTo(PendaftaranModel::class);
17     }
18     public function santri()
19     {
20         return $this->belongsTo(SantriModel::class);
21     }
22 }
23
```

Gambar 4.8 Model Detail Pendaftaran

Pada gambar 4.8, *Model* Detail Pendaftaran menunjukkan koneksi ke database dengan tabel pendaftaran_detail. *Model* ini berfungsi untuk mengelola data pendaftaran setiap calon santri secara detail. Selain itu, *Model* Detail Pendaftaran juga memiliki relasi dengan *Model* Santri dan *Model* Pendaftaran, yang memungkinkan integrasi dan pengelolaan data antara tabel santri, pendaftaran, dan detail pendaftaran.

```
app > Models > ProfilePesantrenModel.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class ProfilePesantrenModel extends Model
9  {
10     use HasFactory;
11     public $table = 'profile_pesantren';
12     public $timestamps = false;
13     protected $guarded = ['id'];
14 }
15
```

Gambar 4.9 Model Profil Pesantren

Pada gambar 4.9, *Model* Profile Pesantren menunjukkan koneksi ke database dengan tabel profil_pesantren. *Model* ini berfungsi untuk mengelola data profil pesantren, yang mencakup informasi penting mengenai pesantren. Dengan koneksi ini, sistem dapat menampilkan, mengubah, dan mengelola informasi profil pesantren

2. View

View dalam arsitektur MVC pada sistem informasi pendaftaran santri bertugas menampilkan informasi kepada pengguna dengan menggunakan data yang diperoleh dari *model*. *View* ini dirancang untuk menghasilkan tampilan antarmuka yang jelas dan *user-friendly*, sehingga memudahkan pengguna untuk berinteraksi dengan aplikasi pendaftaran santri baru di Islamic Center Al-Hazza.

```
<div class="col-lg-12">
  <div class="card">
    <div class="card-body wizard-content">
      <h4 class="card-title">Input Form Pendaftaran</h4>
      <p class="card-subtitle mb-3">Pastikan data yang diisi valid atau sesuai </p>
      <form action="{{ url('santri/pendaftaran') }}/{{ $pendaftaran->id }}" class="validation-wizard wizard-circle mt-5" id
      @csrf
      <!-- Step 1 sampe email -->
      <h6>Data Santri</h6>
      <section>
        <div class="row">
          <div class="col-md-6">
            <div class="mb-3">
              <label class="form-label" for="no_regis"> No Registrasi :
              </label>
              <input type="text" class="form-control" id="no_regis" name="no_regis" readonly value="{{ $user->no_regis }}"
            </div>
          </div>
          <div class="col-md-6">
            <div class="mb-3">
              <label class="form-label" for="nama"> Nama : <span class="text-danger">*</span>
              </label>
              <input type="text" class="form-control required" id="nama" name="nama" value="{{ $user->nama }}" />
            </div>
          </div>
        </div>
        <div class="row">
          <div class="col-md-6">
            <div class="mb-3">
              <label class="form-label" for="nisan"> NISN : <span class="text-danger">*</span>
              </label>
              <input type="text" class="form-control required" id="nisan" name="nisan" value="{{ $user->nisan }}" />
            </div>
          </div>
          <div class="col-md-6">
            <div class="mb-3">
              <label class="form-label" for="alamat"> Alamat : <span class="text-danger">*</span>
              </label>
              <input type="text" class="form-control required" id="alamat" name="alamat" value="{{ $user->alamat }}" />
            </div>
          </div>
        </div>
      </section>
    </div>
  </div>
</div>
```

Gambar 4.10 View Pendaftaran Santri

Pada Gambar 4.10 *View* pendaftaran santri, terdapat kode form yang dirancang untuk mengumpulkan informasi dari calon santri. Form ini memungkinkan calon santri untuk mengisi data pribadi, riwayat pendidikan, dan informasi lain yang diperlukan untuk proses pendaftaran.

```
<table border="0" cellpadding="5" class="mt-4">
  <tr>
    <th>NO REGISTRASI</th>
    <td style="width: 20px;"></td>
    <td> : {{ $user->no_regis }}</td>
  </tr>
  <tr>
    <th>TGL PENDAFTARAN</th>
    <td style="width: 20px;"></td>
    <td> : {{ $detail_pendaftaran->created_at }}</td>
  </tr>
  <tr>
    <th>NISN</th>
    <td style="width: 20px;"></td>
    <td> : {{ $user->nisn }}</td>
  </tr>
  <tr>
    <th>NAMA LENGKAP</th>
    <td style="width: 20px;"></td>
    <td> : {{ $user->nama }}</td>
  </tr>
  <tr>
    <th>JENIS KELAMIN</th>
    <td style="width: 20px;"></td>
    <td> : {{ $user->jenis_kelamin }}</td>
  </tr>
  <tr>
    <th>TEMPAT, TANGGAL LAHIR</th>
    <td style="width: 20px;"></td>
    <td> : {{ $user->tempat_lahir }} {{ $user->tgl_lahir }}</td>
  </tr>
  <tr>
    <th>NAMA ORANGTUA / WALI</th>
    <td style="width: 20px;"></td>
    <td> : </td>
  </tr>
  <tr>
    <th style="text-align: center;">AYAH</th>

```

Gambar 4.11 View Detail Santri

Pada Gambar 4.11 View detail santri, terdapat kode yang bertanggung jawab untuk menampilkan tabel yang berisi informasi detail mengenai santri. Tabel ini menampilkan data seperti nama, alamat, tanggal lahir, riwayat pendidikan, dan informasi lainnya yang relevan terkait santri tersebut. Hal ini memungkinkan pengguna, seperti panitia pendaftaran atau administrator, untuk melihat informasi santri secara lengkap dan terstruktur dalam satu tampilan, memudahkan dalam proses manajemen dan verifikasi data.

```

7       <div class="col-9">
8         <h4 class="fw-semibold mb-8">Dashboard</h4>
9         <nav aria-label="breadcrumb">
10          <ol class="breadcrumb">
11            <li class="breadcrumb-item">
12              <a class="text-muted text-decoration-none" href="{{ url('/admin') }}">Home</a>
13            </li>
14            <li class="breadcrumb-item" aria-current="page">Dashboard</li>
15          </ol>
16        </nav>
17      </div>
18    </div>
19  </div>
20 </div>
21
22 <div class="row">
23   <div class="col-sm-6 col-lg-4 col-xl-2">
24     <a href="{{ url('/admin/siswa') }}" class="p-4 text-center bg-primary-subtle card shadow-none rounded-2">
25       
26       <p class="fw-semibold text-primary mb-1">Calon Santri</p>
27       <h4 class="fw-semibold text-primary mb-0">{{ $data_siswa->count() }}</h4>
28     </a>
29   </div>
30   <div class="col-sm-6 col-lg-4 col-xl-2">
31     <a href="{{ url('/admin/pendaftaran') }}" class="p-4 text-center bg-warning-subtle card shadow-none rounded-2">
32       
33       <p class="fw-semibold text-warning mb-1">Pendaftaran</p>
34       <h4 class="fw-semibold text-warning mb-0">{{ $data_pendaftaran->count() }}</h4>
35     </a>
36   </div>
37   <div class="col-sm-6 col-lg-4 col-xl-2">
38     <a href="{{ url('/admin/pendaftaran') }}" class="p-4 text-center bg-info-subtle card shadow-none rounded-2">
39       
40       <p class="fw-semibold text-info mb-1">Lulus</p>
41       <h4 class="fw-semibold text-info mb-0">{{ $total_lulus->count() }}</h4>
42     </a>
43   </div>
44   <div class="col-sm-6 col-lg-4 col-xl-2">
45     <a href="{{ url('/admin/pendaftaran') }}" class="p-4 text-center bg-danger-subtle card shadow-none rounded-2">
46       
47       <p class="fw-semibold text-danger mb-1">Tidak Lulus</p>
48       <h4 class="fw-semibold text-danger mb-0">{{ $total_tidak_lulus->count() }}</h4>

```

Gambar 4.12 View Dashboard Admin

Pada Gambar 4.12 *view dashboard admin*, terdapat tampilan halaman utama yang menampilkan informasi dasar mengenai pendaftaran santri baru di Islamic Center Al-Hazza. Halaman ini dilengkapi dengan berbagai menu navigasi yang memudahkan admin untuk mengelola data pendaftaran, mengakses informasi santri, serta melakukan verifikasi dan pengelolaan dokumen pendaftaran.

```

<div class="table-responsive mt-3">
  <table id="zero_config" class="table border table-bordered text-nowrap align-middle">
    <thead>
      <!-- start row -->
      <tr>
        <th class="text-center">Thn Angkatan</th>
        <th class="text-center">Gelombang</th>
        <th class="text-center">Kuota</th>
        <th class="text-center">Total Pendaftar</th>
        <th class="text-center">Status</th>
        <!-- <th class="text-center">Tgl Dibuat</th -->
        <th class="text-center">Ops</th>
      </tr>
      <!-- end row -->
    </thead>
    <tbody>
      @foreach ($data_pendaftaran as $pendaftaran)
        <!-- start row -->
        <tr class="text-center">
          <td>Tahun {{ $pendaftaran->tahun_angkatan }}</td>
          <td>Gelombang Ke-{{ $pendaftaran->gelombang }}</td>
          <td>{{ $pendaftaran->kuota }} Orang</td>
          <td>{{ $pendaftaran->detail_pendaftaran->count() }} Orang</td>
          <td>
            @if ($pendaftaran->tutup == 0)
              <span class="badge bg-success-subtle text-success fw-bold">Buka</span>
            @else
              <span class="badge bg-danger-subtle text-danger fw-bold">Tutup</span>
            @endif
          </td>
          <!-- <td>{{ $pendaftaran->created_at }}</td -->
          <td>
            <a href="{{ url('/admin/pendaftaran') }}/{{ $pendaftaran->id }}" class="mb-1 badge font-m

```

Gambar 4.13 View Kelola Pendaftaran

Pada Gambar 4.13 *view* kelola pendaftaran di admin, terdapat kode untuk menampilkan halaman yang memuat fitur-fitur seperti pembukaan gelombang pendaftaran. Halaman ini biasanya mencakup daftar calon santri yang mendaftar, status pendaftaran mereka, serta opsi untuk mengelola setiap entri pendaftaran dan pengelolaan status penerimaan.

3. Controller

Controller berfungsi sebagai pengatur alur logika aplikasi yang menerima input dari pengguna melalui *view*. Setelah itu, *controller* mengolah data yang dibutuhkan dari *model* dan mengembalikan hasilnya kembali ke *view* untuk ditampilkan kepada pengguna.

```

12 class AdminController extends Controller
13 {
14     public function index()
15     {
16
17         $total_lulus = PendaftaranDetailModel::where('status', 1)->get();
18         $total_tidak_lulus = PendaftaranDetailModel::where('status', 2)->get();
19
20         return view('admin.index', [
21             'plugins' => '',
22             'menu_master' => 'false',
23             'menu' => 'dashboard',
24             'judul' => 'Dashboard',
25             'pesantren' => ProfilePesantrenModel::first(),
26             'data_santri' => SantriModel::all(),
27             'data_pendaftaran' => PendaftaranModel::all(),
28             'total_lulus' => $total_lulus,
29             'total_tidak_lulus' => $total_tidak_lulus,
30         ]);
31     }
32
33     public function profile()
34     {
35         // dd(ProfilePesantrenModel::first());
36         return view('admin.profile', [
37             'plugins' => '',
38             'menu_master' => 'false',
39             'menu' => 'profile',
40             'judul' => 'Profile Pesantren',
41             'pesantren' => ProfilePesantrenModel::first(),
42         ]);
43     }
44
45     public function profile_foto(Request $request)
46     {
47         if ($request->file('foto')) {
48             if ($request->foto_lama) {
49                 if ($request->foto_lama != 'favicon.png') {
50                     Storage::delete('assets/files/' . $request->foto_lama);
51                 }
52             }
53             $data['foto'] = str_replace('assets/files/', '', $request->file('foto')->store('assets/files'));
54         }
55
56         ProfilePesantrenModel::where('id', $request->id)
57             ->update($data);
58         return redirect('/admin/profile')->with('pesan', "
59         <script>
60         Swal.fire(

```

Gambar 4.14 Controller Admin

Pada Gambar 4.14 *Controller* admin dalam sistem pendaftaran santri baru di Islamic Center Al-Hazza menyediakan berbagai fungsionalitas yang krusial bagi pengelola sistem. Salah satu fungsionalitas utamanya adalah *dashboard* admin yang dirancang untuk menampilkan informasi penting seperti statistik pendaftaran, status terkini dari proses penerimaan santri, dan ringkasan aktivitas terbaru dalam sistem. Selain itu, *controller* ini juga mengelola manajemen profil admin, memungkinkan mereka untuk mengakses dan mengubah informasi profil mereka melalui formulir yang disediakan. Hal ini termasuk operasi untuk mengambil data profil dari *model*, menampilkan form *edit* profil, serta menyimpan perubahan yang diperlukan oleh admin untuk menjaga akurasi dan kelengkapan informasi.

```

12 class AdminPendaftaranController extends Controller
13 {
14     /**
15      * Display a listing of the resource.
16      */
17     public function index()
18     {
19         return view('admin.pendaftaran.index', [
20             'plugins' => '
21                 <link rel="stylesheet" href="' . url('/assets/template') . '/dist/assets/l
22                 <script src="' . url('/assets/template') . '/dist/assets/libs/datatables.n
23                 <link rel="stylesheet" href="' . url('/assets/template') . '/dist/assets/l
24                 <script src="' . url('/assets/template') . '/dist/assets/libs/prismjs/pris
25             ',
26             'menu_master' => 'false',
27             'menu' => 'pendaftaran',
28             'judul' => 'Data Pendaftaran',
29             'pesantren' => ProfilePesantrenModel::first(),
30             'data_pendaftaran' => PendaftaranModel::all()->sortByDesc('id')
31         ]);
32     }
33
34     /**
35      * Show the form for creating a new resource.
36      */
37     public function create()
38     {
39         //
40     }
41
42     /**
43      * Store a newly created resource in storage.
44      */
45     public function store(Request $request)
46     {
47         $data = [
48             'tahun_angkatan' => $request->tahun_angkatan,
49             'gelombang' => $request->gelombang,
50             'kuota' => $request->kuota,
51             'tutup' => 0
52         ];
53
54         PendaftaranModel::create($data);
55         return redirect('/admin/pendaftaran')->with('pesan', "
56             <script>
57                 Swal.fire(
58                     {
59                         title: 'Berhasil',
60                         text: 'Data disimpan'

```

Gambar 4.15 Controller Admin Pendaftaran

Pada Gambar 4.15 *Controller* admin pendaftaran memiliki fungsi penting dalam mengelola data pendaftaran. *Controller* ini menyediakan fungsionalitas bagi admin untuk membuka gelombang pendaftaran serta melihat, memverifikasi, dan mengelola data pendaftaran calon santri. Hal ini termasuk menampilkan daftar pendaftar, memverifikasi dokumen, mengubah status penerimaan, dan menyimpan perubahan ke dalam sistem.

```

14 class SantriController extends Controller
15 {
16     public function index()
17     {
18         $notifikasi = PendaftaranDetailModel::where('santri_id', session()->get('id'))
19             ->where('notif', 0)
20             ->get();
21
22         $cek_lulus = PendaftaranDetailModel::where('santri_id', session()->get('id'))
23             ->where('status', 1)
24             ->first();
25
26         return view('santri.index', [
27             'plugins' => '
28                 <link rel="stylesheet" href="'. url('assets/template/dist') . '/assets/libs/prismjs/themes/prism-oka
29                 <script src="'. url('assets/template/dist') . '/assets/libs/prismjs/prism.js"></script>
30                 <script src="'. url('assets/template/dist') . '/assets/js/widget/ui-card-init.js"></script>
31             ',
32             'menu' => 'dashboard',
33             'judul' => 'Dashboard',
34             'user' => SantriModel::firstWhere('id', session()->get('id')),
35             'pesantren' => ProfilePesantrenModel::first(),
36             'notifikasi' => $notifikasi,
37             'cek_lulus' => $cek_lulus,
38             'data_pendaftaran' => PendaftaranModel::where('tutup', 0)->get(),
39             'detail_pendaftaran' => PendaftaranDetailModel::where('santri_id', session()->get('id'))->get()
40         ]);
41     }
42 }

```

Gambar 4.16 Controller Santri

Pada Gambar 4.16 *Controller* santri pada sistem pendaftaran santri baru memiliki fungsionalitas penting yang disesuaikan dengan kebutuhan santri. Fungsi utama dari *controller* ini mencakup Dashboard Santri yang bertanggung jawab untuk menampilkan informasi terkait proses pendaftaran yang sedang diikuti oleh santri, termasuk detail status pendaftaran serta kemajuan proses penerimaan. Selain itu, notifikasi pengumuman memungkinkan santri untuk menerima informasi terbaru mengenai pengumuman penerimaan, termasuk pembukaan gelombang pendaftaran berikutnya. Menu-menu penting dalam *controller* ini menyediakan akses santri terhadap fitur-fitur yang diperlukan dalam sistem, seperti pengelolaan profil pendaftaran. Dengan menggunakan *controller* santri ini, santri dapat dengan mudah mengakses dan mengelola informasi yang relevan dengan proses pendaftaran.

```

168 public function store_pendaftaran(Request $request, PendaftaranModel $pendaftaran)
169 {
170     $data = [
171         'nama' => $request->nama,
172         'nisn' => $request->nisn,
173         'nis' => $request->nis,
174         'jurusan' => $request->jurusan,
175         'jenis_kelamin' => $request->jenis_kelamin,
176         'tempat_lahir' => $request->tempat_lahir,
177         'tgl_lahir' => $request->tgl_lahir,
178         'agama' => $request->agama,
179         'anak_ke' => $request->anak_ke,
180         'jumlah_saudara' => $request->jumlah_saudara,
181         'hobi' => $request->hobi,
182         'cita_cita' => $request->cita_cita,
183         'hp' => $request->hp,
184         'email' => $request->email,
185         'jenis_tempat_tinggal' => $request->jenis_tempat_tinggal,
186         'alamat' => $request->alamat,
187         'desa' => $request->desa,
188         'kecamatan' => $request->kecamatan,
189         'kabupaten' => $request->kabupaten,
190         'provinsi' => $request->provinsi,
191         'pos' => $request->pos,
192         'jarak' => $request->jarak,
193         'transportasi' => $request->transportasi,
194         'no_kk' => $request->no_kk,
195         'kepala_kk' => $request->kepala_kk,
196         'nama_ayah' => $request->nama_ayah,
197         'nik_ayah' => $request->nik_ayah,
198         'tahun_lahir_ayah' => $request->tahun_lahir_ayah,
199         'pekerjaan_ayah' => $request->pekerjaan_ayah,
200         'penghasilan_ayah' => $request->penghasilan_ayah,
201         'pendidikan_ayah' => $request->pendidikan_ayah,
202         'nama_ibu' => $request->nama_ibu,
203         'nik_ibu' => $request->nik_ibu,
204         'tahun_lahir_ibu' => $request->tahun_lahir_ibu,
205         'pekerjaan_ibu' => $request->pekerjaan_ibu,
206         'penghasilan_ibu' => $request->penghasilan_ibu,
207         'pendidikan_ibu' => $request->pendidikan_ibu,
208         'sekolah_asal' => $request->sekolah_asal,
209         'jenjang_sekolah' => $request->jenjang_sekolah,
210         'npsn_sekolah' => $request->npsn_sekolah,
211         'foto' => $request->foto_lama,
212     ];
213
214     if ($request->file('foto')) {
215         if ($request->gambar_lama) {
216             if ($request->gambar_lama != 'default.png') {
217                 Storage::delete('assets/files/' . $request->gambar_lama);
218             }
219         }
220         $data['foto'] = str_replace('assets/files/', '', $request->file('foto')->store('assets/files'));

```

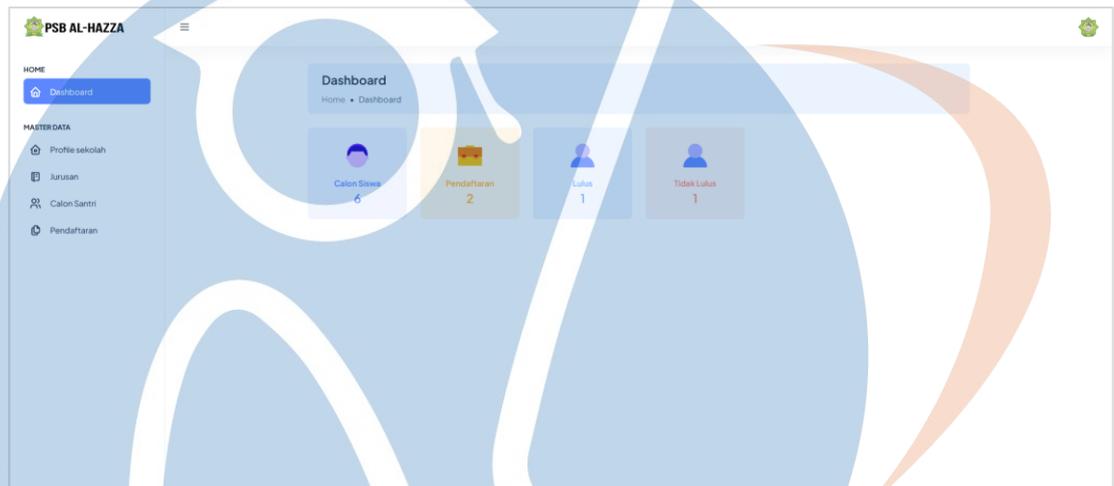
Gambar 4.17 Controller Pendaftaran

Pada Gambar 4.17 *Controller* Pendaftaran bertanggung jawab untuk mengelola form pendaftaran calon santri yang harus diisi. *Controller* ini menangani proses validasi input dari form, penyimpanan data pendaftaran ke dalam database, serta interaksi antara *view* pendaftaran santri dengan *model* data calon santri. Selain itu, *controller* ini juga memastikan bahwa semua langkah dalam proses pendaftaran berjalan lancar dan sesuai dengan aturan yang telah ditetapkan.

4.3.2 Implementasi Antarmuka

Antarmuka adalah tampilan dan mekanisme interaksi yang memungkinkan pengguna berinteraksi dengan sistem, menginput data, serta menerima informasi secara efektif dan efisien. Dalam implementasi antarmuka sistem, peneliti akan menampilkan antarmuka admin dan antarmuka untuk calon santri.

1. *Dashboard Admin*

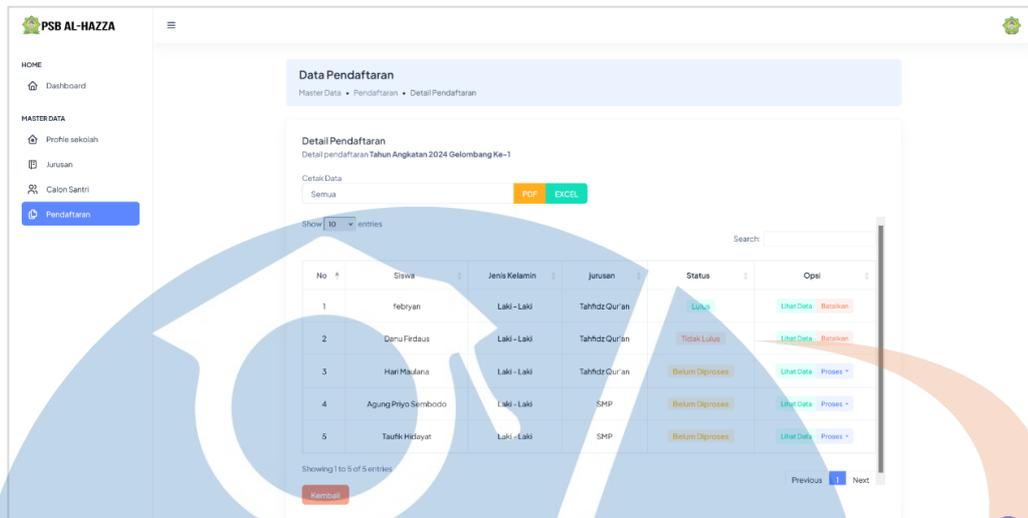


Gambar 4.18 Dashboard Admin

Pada gambar 4.18 menunjukkan dashboard admin yang berisi informasi umum jumlah pendaftaran dan menu yang tersedia.

STT - NF

2. Kelola Pendaftaran



The screenshot displays the 'Data Pendaftaran' management interface. It includes a sidebar with navigation options like 'Dashboard', 'Profil sekolah', 'Jurusan', 'Calon Santri', and 'Pendaftaran'. The main content area shows a table of registrants with the following data:

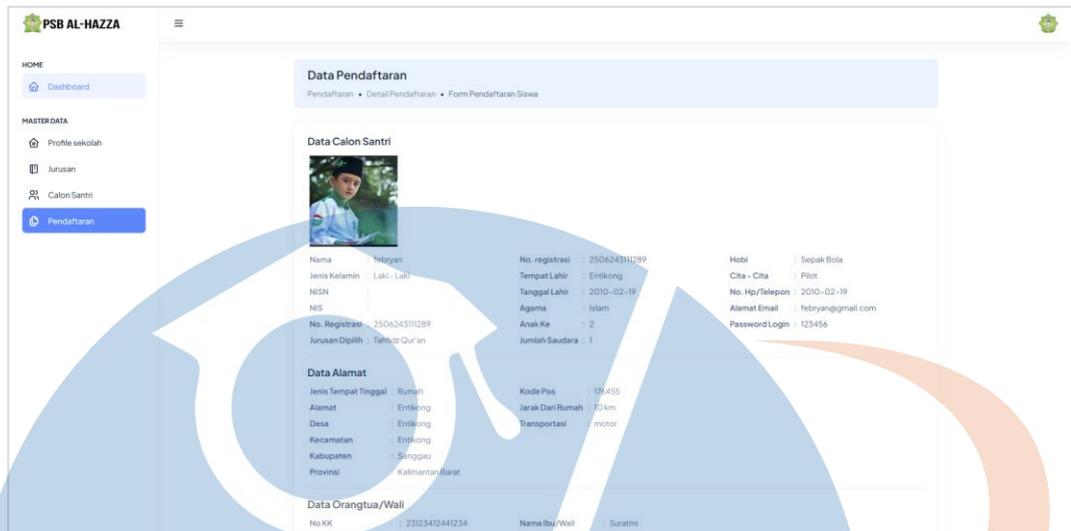
No	Nama	Jenis Kelamin	Jurusan	Status	Opsi
1	Febryan	Laki-Laki	Tahfidz Qur'an	Lulus	Lihat Data, Batalkan
2	Danu Firdaus	Laki-Laki	Tahfidz Qur'an	Tidak Lulus	Lihat Data, Batalkan
3	Hari Mas'ana	Laki-Laki	Tahfidz Qur'an	Belum Diproses	Lihat Data, Proses
4	Agung Priyo Sembodo	Laki-Laki	SMP	Belum Diproses	Lihat Data, Proses
5	Taufik Hidayat	Laki-Laki	SMP	Belum Diproses	Lihat Data, Proses

Gambar 4.19 Halaman Kelola Pendaftaran

Pada Gambar 4.19 ditunjukkan halaman Kelola Pendaftaran, di mana admin dapat melihat calon santri yang mendaftar serta status pendaftaran mereka. Admin juga dapat mengubah status pendaftaran dengan mengklik tombol opsi dan memilih "lulus" atau "tidak lulus, serta admin juga dapat mencetak data pendaftaran dengan menekan tombol PDF atau Excel."

STT - NF

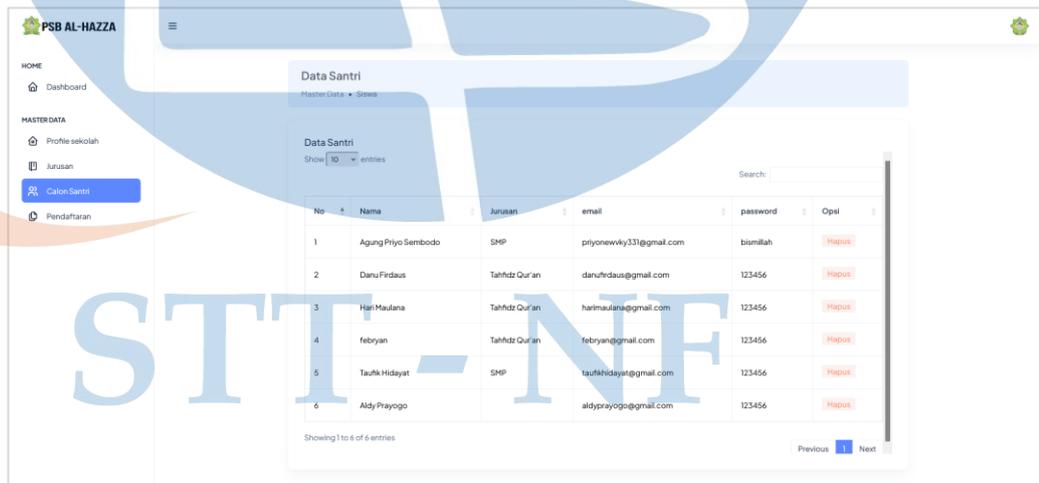
3. Detail Pendaftar / Calon santri



Gambar 4.20 Detail Calon Santri

Pada Gambar 4.20, halaman detail calon santri dapat admin akses dengan mengklik tombol "Lihat Data" di salah satu pendaftar untuk melihat informasi calon santri secara lengkap.

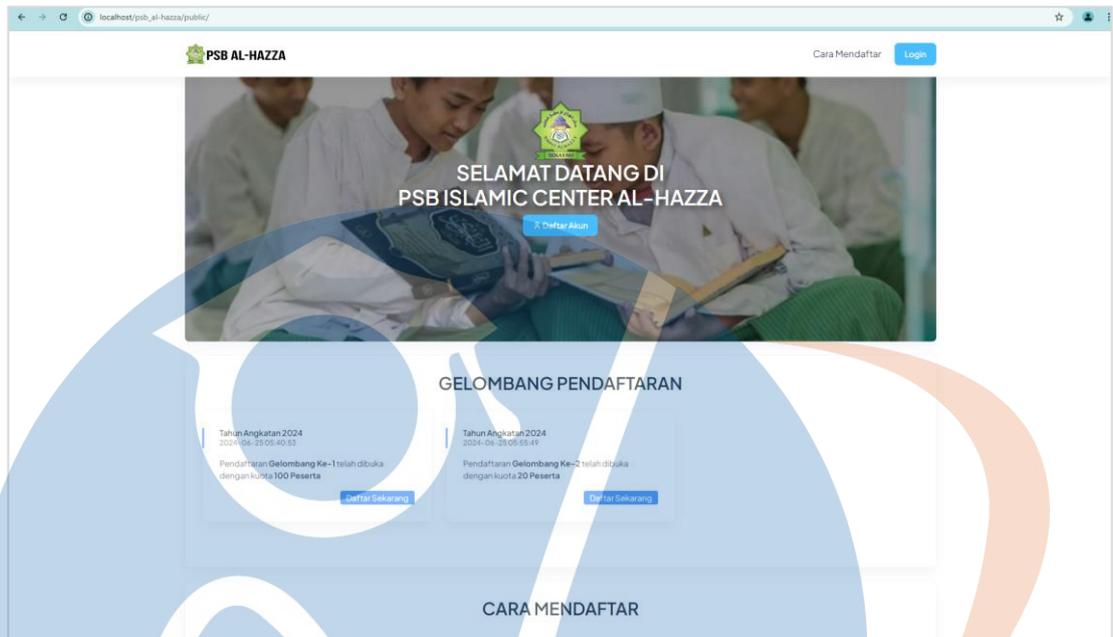
4. Kelola Calon Santri



Gambar 4.21 Halaman Kelola Calon Santri

Pada Gambar 4.21 ditunjukkan halaman Kelola Calon Santri, di mana admin dapat melihat daftar akun calon santri yang mendaftar. Admin juga dapat menghapus akun calon santri pendaftaran dengan mengklik tombol "Hapus"

5. Halaman *Landing Page*

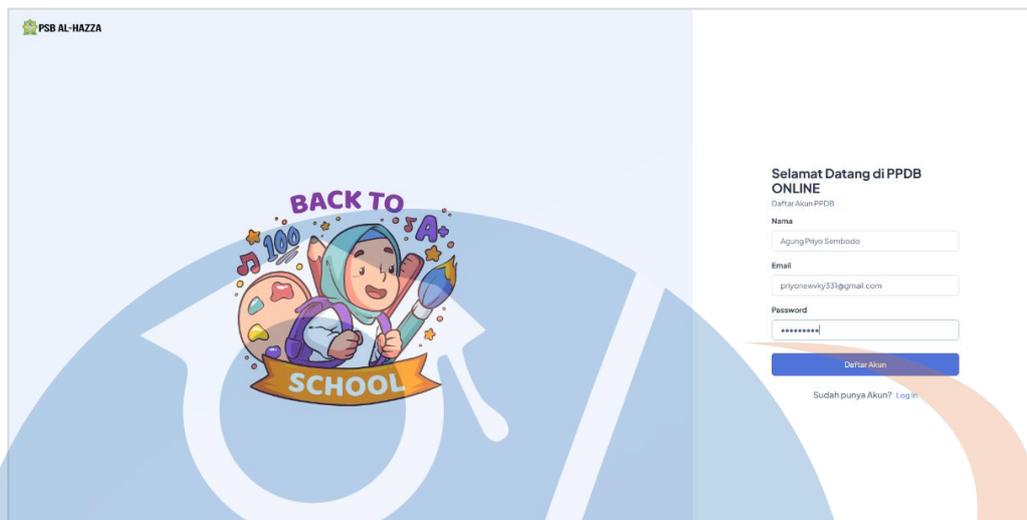


Gambar 4.22 Halaman Landing Page

Pada Gambar 4.22, halaman landing page adalah tempat di mana para calon santri pertama kali dapat melihat dan mengetahui informasi pendaftaran. Halaman ini berisi mengenai proses pendaftaran, serta kontak yang dapat dihubungi untuk informasi lebih lanjut. Selain itu, halaman ini juga menyediakan tautan langsung untuk memulai proses pendaftaran online.

STT - NF

6. Halaman Pembuatan Akun Pendaftaran



PSB AL-HAZZA

BACK TO SCHOOL

Selamat Datang di PPDB ONLINE
Daftar Akun-PPDB

Nama
Agung Phyo Sembodo

Email
pryonevkiy33@gmail.com

Password

Daftar Akun

Sudah punya Akun? [Log in](#)

Gambar 4.23 Halaman Pembuatan Akun Pendaftaran

Pada Gambar 4.23 menunjukkan halaman pembuatan akun yang akan digunakan calon santri untuk login dan mengikuti proses pendaftaran online.

7. Halaman Form Pendaftaran Santri



PSB AL-HAZZA

HOME
Dashboard

SISWA MENU
Profile
Pendaftaran

Input Form Pendaftaran
Pastikan data yang diisi valid atau sesuai

1 Data Siswa 2 Data Alamat 3 Data Orang Tua/Wali 4 Data Sekolah

No Registrasi : 250624311289

Jurusan : Pilih

Tempat Lahir : *

Agama : *

Jumlah Saudara : *

Cita Cita : *

Email : *

Nama : * febyan

Jenis Kelamin : * Pilih

Tanggal Lahir : * hh / bb / tttt

Anak Ke : *

Hobi : *

No HP : *

Foto : *
Telusuri... Tidak ada berkas dipilih.

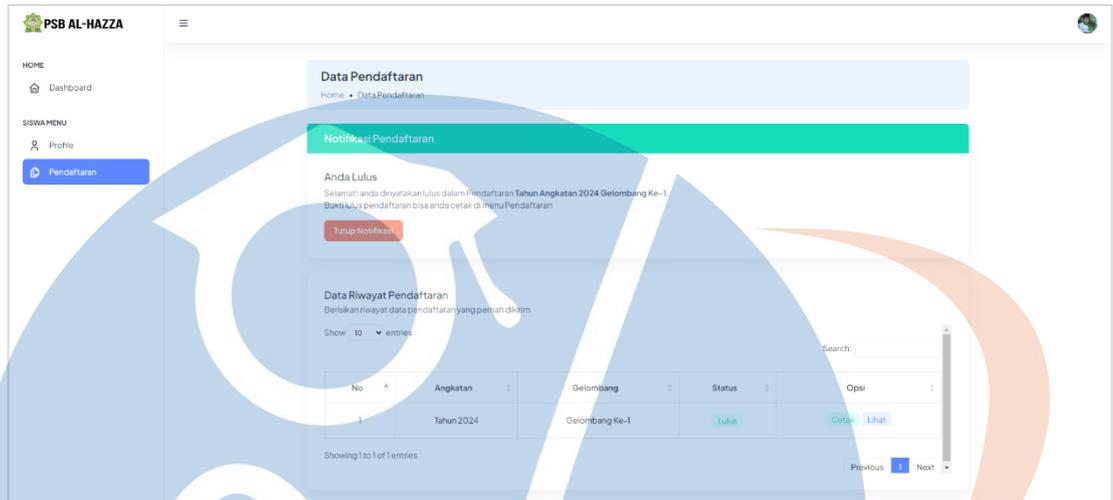
Previous Next

Gambar 4.24 Halaman Form Pendaftaran

Pada Gambar 4.24, halaman form pendaftaran adalah tempat di mana para calon santri dapat mengikuti proses pendaftaran online dengan mengisi data diri mereka secara lengkap. Halaman ini menyediakan berbagai kolom untuk informasi

pribadi, data orang tua, dan data sekolah sebelumnya serta mengunggah dokumen yang diperlukan.

8. Halaman Pendaftaran Santri



Gambar 4.25 Gambar Halaman Pendaftaran

Pada Gambar 4.25, halaman pendaftaran santri memungkinkan calon santri untuk memantau proses pendaftarannya dan menerima pengumuman mengenai kelulusannya. Di halaman ini, calon santri juga dapat mencetak bukti pendaftarannya dengan mengklik tombol cetak. Hal ini memberikan kemudahan bagi calon santri untuk mengikuti dan mengelola proses pendaftaran.

4.4 Pengujian Sistem

Pengujian sistem merupakan serangkaian langkah yang dilakukan untuk menguji dan menilai sistem yang telah dirancang, dengan tujuan memastikan bahwa aplikasi beroperasi sesuai dengan persyaratan yang telah ditentukan.

4.4.1 Black Box Testing

Dalam penelitian ini, pengujian *black box* dilakukan dengan menguji kinerja sistem berdasarkan spesifikasi yang telah dirancang dan membandingkannya dengan hasil yang dicapai, dengan fokus pada pengujian fungsionalitas fitur. Skenario pengujian dan prosedur uji diperoleh dari rencana pengujian *black box* yang telah dibuat sebelumnya. Berikut ini adalah hasil uji coba yang dilakukan menggunakan pendekatan *black box testing*.

Tabel 4.1 Pengujian Black Box Test

No	Fitur	Proses	Hasil yang diharapkan	Hasil tes
1	<i>Login</i>	Pengguna memasukkan username dan password	Pengguna berhasil <i>login</i>	Berhasil
2	<i>Dashboard</i>	Pengguna berhasil login, setelah itu sistem menampilkan halaman <i>dashboard</i>	Pengguna berhasil melihat <i>dashboard</i>	Berhasil
3	Kelola data pendaftaran	Admin berhasil login, setelah itu klik menu pendaftaran	Admin berhasil mengelola data pendaftaran	Berhasil
4	Melihat data calon santri	Admin berhasil login, setelah itu klik menu pendaftaran kemudian klik tombol lihat data disalah satu data	Admin dapat melihat data calon santri	Berhasil
5	Ubah status pendaftaran	Admin berhasil login, setelah itu klik tombol “lulus” atau “tidak lulus” di halaman pendaftaran	Admin dapat merubah status verifikasi pendaftaran	Berhasil
6	Cetak data calon santri	Admin berhasil login, setelah itu klik menu pendaftaran, kemudian klik PDF atau Excel untuk mencetak data calon santri	Admin dapat mencetak data calon santri	Berhasil
7	Kelola Profil Pesantren	Admin berhasil login, setelah itu klik menu profil pesantren	Admin berhasil mengelola profil pesantren	Berhasil
8	Pembuatan akun Pendaftaran	Calon santri membuka halaman pendaftaran ,kemudian menekan tombol Daftar akun	Calon santri dapat melakukan pendaftaran akun dengan mengisi nama,email & password	Berhasil
9	Pendaftaran	Calon santri berhasil login, kemudian	Calon santri dapat melakukan	Berhasil

No	Fitur	Proses	Hasil yang diharapkan	Hasil tes
		mengisi form pendaftaran	pendaftaran pesantren dengan mengisi form	
10	Notifikasi Pengumuman	Calon santri berhasil login, pada halaman dashboard terdapat notifikasi pengumuman	Calon santri dapat melihat pengumuman pendaftaran “lulus” atau “tidak lulus” di <i>dashboard</i>	Berhasil
11	Cetak bukti pendaftaran oleh santri	Calon santri berhasil login, setelah itu klik menu pendaftaran, kemudian klik opsi cetak	Calon santri dapat mencetak bukti pendaftaran	Berhasil
12	<i>Logout</i>	Pengguna menekan tombol logout pada menu utama sistem informasi	Pengguna berhasil <i>logout</i>	Berhasil

Pada Tabel 4.1 Pengujian *Black Box*, dari 12 pengujian yang telah direncanakan, hasil uji coba aplikasi secara keseluruhan sesuai dengan yang diharapkan.

Pengujian black box untuk sistem informasi pendaftaran santri baru melibatkan 12 skenario pengujian yang dilaksanakan oleh pengembang, kemudian dapat dinilai tingkat keberhasilannya:

- Persentase Keberhasilan = $\frac{\text{Jumlah pengujian yang berhasil}}{\text{Total pengujian}} \times 100\%$
- Persentase Keberhasilan = $\frac{12}{12} \times 100\%$
- Persentase Keberhasilan = 100%

Kesimpulannya, seluruh pengujian berhasil diselesaikan oleh sistem. Dari hasil uji black box pada 12 fitur yang diuji, disimpulkan bahwa semua fitur (100%) berfungsi dengan baik dan sesuai dengan standar yang diharapkan.

4.4.2 User Acceptance Test

Pengujian UAT adalah serangkaian tes yang dilakukan oleh pengguna untuk menghasilkan dokumentasi sebagai bukti keberhasilan atau kegagalan sistem yang telah dikembangkan. Jika hasil pengujian berjalan sesuai kebutuhan pengguna, aplikasi dapat diterapkan. Berikut ini adalah hasil dari respon staff Islamic Center Al-Hazza dan calon santri terhadap pengujian yang telah disiapkan.

1. UAT Admin

Tabel 4.2 Hasil User Acceptance Test (Admin)

No	Fitur	Hasil yang diharapkan	Berhasil	Tidak Berhasil
1	<i>Login</i>	Pengguna berhasil <i>login</i>	7	-
2	<i>Dashboard</i>	Pengguna berhasil melihat <i>dashboard</i>	7	-
3	Kelola data pendaftaran	Admin berhasil mengelola data pendaftaran	6	1
4	Melihat data calon santri	Admin dapat melihat data calon santri	7	-
5	Ubah status pendaftaran	Admin dapat merubah status verifikasi pendaftaran	7	-
6	Cetak data calon santri	Admin dapat mencetak data calon santri	6	1
7	Kelola Profil Pesantren	Admin berhasil mengelola profil pesantren	7	-
8	<i>Logout</i>	Pengguna berhasil <i>logout</i>	7	-

Uji coba UAT Admin melibatkan 7 responden, yang menguji sistem dengan memberikan penilaian terhadap keberhasilan atau kegagalan setiap fitur yang ada. Berdasarkan hasil pengujian ini, dapat ditentukan apakah sistem memenuhi kebutuhan dan harapan pengguna sebelum diterapkan secara penuh.

2. UAT Calon Santri / Orang tua

Tabel 4.3 Hasil User Acceptance Test (Calon santri)

No	Fitur	Hasil yang diharapkan	Berhasil	Tidak Berhasil
1	<i>Login</i>	Pengguna berhasil <i>login</i>	5	-
2	<i>Dashboard</i>	Pengguna berhasil melihat <i>dashboard</i>	5	-
3	Pembuatan akun Pendaftaran	Calon santri dapat melakukan pendaftaran akun dengan mengisi nama, email & password	5	-
4	Pendaftaran	Calon santri dapat melakukan pendaftaran pesantren dengan mengisi form	5	-
5	Notifikasi Pengumuman	Calon santri dapat melihat pengumuman pendaftaran “lulus” atau “tidak lulus” di <i>dashboard</i>	5	-
6	Cetak bukti pendaftaran oleh santri	Calon santri dapat mencetak bukti pendaftaran	4	1
7	<i>Logout</i>	Pengguna berhasil <i>logout</i>	5	-

Uji coba UAT untuk calon santri atau orang tua melibatkan 5 responden, yang juga memberikan penilaian berdasarkan pengalaman mereka dalam menggunakan sistem. Berdasarkan hasil pengujian ini, dapat ditentukan apakah sistem dapat berjalan dengan baik memenuhi kebutuhan dan harapan pengguna sebelum diterapkan secara penuh.

4.5 Evaluasi Sistem

Untuk mengevaluasi sistem informasi pendaftaran santri baru yang telah dikembangkan, penulis melakukan perhitungan dengan memberikan persentase keberhasilan kepada setiap fitur yang dikembangkan berdasarkan jumlah responden.

4.5.1 Evaluasi Fitur Pada Admin

Evaluasi fitur pada admin yang melibatkan 7 responden berfokus pada penilaian sederhana mengenai keberhasilan atau kegagalan masing-masing fitur. Setiap responden mengevaluasi apakah fitur tersebut berfungsi dengan baik atau tidak. Hasil evaluasi ini memberikan gambaran jelas mengenai efektivitas sistem, dan membantu mengidentifikasi fitur yang memerlukan perbaikan.

Tabel 4.4 Hasil Perhitungan Persentase Keberhasilan Fitur (Admin)

No	Fitur	Berhasil	Tidak Berhasil	Persentase Keberhasilan
1	<i>Login</i>	7	-	$\frac{7}{7} \times 100\% = 100\%$
2	<i>Dashboard</i>	7	-	$\frac{7}{7} \times 100\% = 100\%$
3	Kelola data pendaftaran	6	1	$\frac{6}{7} \times 100\% = 86\%$
4	Melihat data calon santri	7	-	$\frac{7}{7} \times 100\% = 100\%$
5	Ubah status pendaftaran	7	-	$\frac{7}{7} \times 100\% = 100\%$
6	Cetak data calon santri	6	1	$\frac{6}{7} \times 100\% = 86\%$
7	Kelola Profil Pesantren	7	-	$\frac{7}{7} \times 100\% = 100\%$
8	<i>Logout</i>	7	-	$\frac{7}{7} \times 100\% = 100\%$

Setelah menjalankan perhitungan pada user acceptance test, analisis yang diperoleh untuk setiap pernyataan adalah:

1. Analisis fitur *login*

Dari tabel 4.4 terlihat bahwa total keberhasilan untuk fitur *login* adalah 7 dari 7 responden, dengan persentase keberhasilan adalah $\frac{7}{7} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur *login* berjalan dengan baik dan sesuai dengan yang diharapkan.

2. Analisis fitur *dashboard*

Dari tabel 4.4 terlihat bahwa total keberhasilan untuk fitur *dashboard* adalah 7 dari 7 responden, dengan persentase keberhasilan adalah $\frac{7}{7} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur *dashboard* berjalan dengan baik dan sesuai dengan yang diharapkan.

3. Analisis fitur kelola data pendaftaran

Dari tabel 4.4 terlihat bahwa total keberhasilan untuk fitur kelola data pendaftaran adalah 6 dari 7 responden, dengan persentase keberhasilan adalah $\frac{6}{7} \times 100\% = 86\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur kelola data pendaftaran berjalan dengan baik dan sesuai dengan yang diharapkan.

4. Analisis fitur melihat data calon santri

Dari tabel 4.4 terlihat bahwa total keberhasilan untuk fitur melihat data calon santri adalah 7 dari 7 responden, dengan persentase keberhasilan adalah $\frac{7}{7} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur melihat data calon santri berjalan dengan baik dan sesuai dengan yang diharapkan.

5. Analisis fitur ubah status pendaftaran

Dari tabel 4.4 terlihat bahwa total keberhasilan untuk fitur ubah status pendaftaran adalah 7 dari 7 responden, dengan persentase keberhasilan adalah $\frac{7}{7} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur ubah status pendaftaran berjalan dengan baik dan sesuai dengan yang diharapkan.

6. Analisis fitur cetak data calon santri

Dari tabel 4.4 terlihat bahwa total keberhasilan untuk fitur cetak data calon santri adalah 6 dari 7 responden, dengan persentase keberhasilan adalah $\frac{6}{7} \times 100\% = 86\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur cetak data calon santri berjalan dengan baik dan sesuai dengan yang diharapkan.

7. Analisis fitur kelola profil pesantren

Dari tabel 4.4 terlihat bahwa total keberhasilan untuk fitur kelola profil pesantren adalah 7 dari 7 responden, dengan persentase keberhasilan adalah $\frac{7}{7} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur kelola profil pesantren berjalan dengan baik dan sesuai dengan yang diharapkan.

8. Analisis fitur *logout*

Dari tabel 4.4 terlihat bahwa total keberhasilan untuk fitur *logout* adalah 7 dari 7 responden, dengan persentase keberhasilan adalah $\frac{7}{7} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur *logout* berjalan dengan baik dan sesuai dengan yang diharapkan.

9. Analisis kegagalan fitur pada admin

Dari tabel 4.4 terlihat bahwa terdapat 2 pengujian yang tidak berhasil oleh satu responden yang sama yaitu kegagalan dalam memuat halaman pada fitur pengelolaan data pendaftaran dan fitur pencetakan data calon santri. Setelah investigasi lebih lanjut, diketahui bahwa responden tersebut mengalami kesulitan mengakses kedua halaman tersebut karena menggunakan perangkat handphone. Dengan demikian, dapat disimpulkan bahwa fungsionalitas pada admin untuk fitur pengelolaan data pendaftaran dan fitur pencetakan data calon santri tidak selalu berjalan dengan baik ketika diakses menggunakan perangkat handphone.

4.5.2 Evaluasi Fitur Pada Calon Santri

Evaluasi fitur pada *user* calon santri yang melibatkan 5 responden berfokus pada penilaian sederhana mengenai keberhasilan atau kegagalan masing-masing fitur. Setiap responden mengevaluasi apakah fitur tersebut berfungsi dengan baik atau tidak. Hasil evaluasi ini memberikan gambaran jelas mengenai efektivitas sistem, dan membantu mengidentifikasi fitur yang memerlukan perbaikan.

Tabel 4.5 Hasil Perhitungan Persentase Keberhasilan Fitur (Calon santri)

No	Fitur	Berhasil	Tidak Berhasil	Persentase Keberhasilan
1	<i>Login</i>	5	-	$\frac{5}{5} \times 100\% = 100\%$
2	<i>Dashboard</i>	5	-	$\frac{5}{5} \times 100\% = 100\%$
3	Pembuatan akun Pendaftaran	5	-	$\frac{5}{5} \times 100\% = 100\%$
4	Pendaftaran	5	-	$\frac{5}{5} \times 100\% = 100\%$
5	Notifikasi Pengumuman	5	-	$\frac{5}{5} \times 100\% = 100\%$
6	Cetak bukti pendaftaran	4	1	$\frac{4}{5} \times 100\% = 80\%$
7	<i>Logout</i>	5	-	$\frac{5}{5} \times 100\% = 100\%$

1. Analisis fitur *login*

Dari tabel 4.5 terlihat bahwa total keberhasilan untuk fitur *login* adalah 5 dari 5 responden, dengan persentase keberhasilan adalah $\frac{5}{5} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur *login* berjalan dengan baik dan sesuai dengan yang diharapkan.

2. Analisis fitur *dashboard*

Dari tabel 4.5 terlihat bahwa total keberhasilan untuk fitur *dashboard* adalah 5 dari 5 responden, dengan persentase keberhasilan adalah $\frac{5}{5} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur *dashboard* pada *user* calon santri berjalan dengan baik dan sesuai dengan yang diharapkan.

3. Analisis fitur pembuatan akun pendaftaran

Dari tabel 4.5 terlihat bahwa total keberhasilan untuk pembuatan akun pendaftaran adalah 5 dari 5 responden, dengan persentase keberhasilan adalah $\frac{5}{5} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur pembuatan akun pendaftaran berjalan dengan baik dan sesuai dengan yang diharapkan.

4. Analisis fitur pendaftaran

Dari tabel 4.5 terlihat bahwa total keberhasilan untuk fitur pendaftaran adalah 5 dari 5 responden, dengan persentase keberhasilan adalah $\frac{5}{5} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur pendaftaran santri berjalan dengan baik dan sesuai dengan yang diharapkan.

5. Analisis fitur notifikasi pengumuman

Dari tabel 4.5 terlihat bahwa total keberhasilan untuk fitur notifikasi pengumuman adalah 5 dari 5 responden, dengan persentase keberhasilan adalah $\frac{5}{5} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur notifikasi pengumuman berjalan dengan baik dan sesuai dengan yang diharapkan.

6. Analisis fitur cetak bukti pendaftaran

Dari tabel 4.5 terlihat bahwa total keberhasilan untuk fitur cetak bukti pendaftaran adalah 4 dari 5 responden, dengan persentase keberhasilan adalah $\frac{4}{5} \times 100\% = 80\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur cetak bukti pendaftaran berjalan dengan baik dan sesuai dengan yang diharapkan.

7. Analisis fitur *logout*

Dari tabel 4.5 terlihat bahwa total keberhasilan untuk fitur *logout* adalah 5 dari 5 responden, dengan persentase keberhasilan adalah $\frac{5}{5} \times 100\% = 100\%$. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur *logout* berjalan dengan baik dan sesuai dengan yang diharapkan.

8. Analisis kegagalan fitur pada calon santri

Dari tabel 4.5 terlihat bahwa terdapat 1 pengujian yang tidak berhasil oleh satu responden, yaitu kegagalan dalam memuat halaman cetak bukti pendaftaran. Setelah ditanyakan lebih lanjut kepada responden, diketahui bahwa halaman cetak bukti pendaftaran gagal dimuat karena responden menggunakan jaringan wi-fi umum ketika melakukan pengujian. Dengan demikian, dapat disimpulkan bahwa fungsionalitas fitur cetak bukti pendaftaran bagi calon santri tidak selalu berjalan lancar saat diakses menggunakan wi-fi umum.

4.5.3 Evaluasi Hasil Pengujian

Hasil akhir dari keseluruhan pengujian dapat diketahui berdasarkan nilai yang diberikan oleh para responden. Nilai ini kemudian dapat dihitung dengan cara menghitung jumlah seluruh keberhasilan dan jumlah seluruh pengujian. Persentase keberhasilan diperoleh dengan membagi jumlah keberhasilan dengan jumlah pengujian, kemudian dikalikan dengan 100%. Metode ini memberikan gambaran jelas mengenai efektivitas dan fungsionalitas sistem berdasarkan umpan balik dari para pengguna.

• Pengujian Fitur Berhasil pada Admin	= 54
• Pengujian Fitur Tidak Berhasil pada Admin	= 2
• Pengujian Fitur Berhasil pada Calon Santri	= 34
• Pengujian Fitur Tidak Berhasil pada Calon Santri	= 1
Total Keberhasilan Pengujian	= 54 + 34 = 88
Total Seluruh Pengujian	= 54 + 2 + 34 + 1 = 91

Pada perhitungan di atas, dapat diketahui bahwa dari seluruh 91 pengujian, bahwa 88 pengujian yang dilakukan responden berhasil. Persentase keberhasilan sistem dapat dilihat dari perhitungan berikut ini:

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah pengujian yang berhasil}}{\text{Total pengujian}} \times 100\%$$

$$\text{Persentase Keberhasilan} = \frac{88}{91} \times 100\%$$

$$\text{Persentase Keberhasilan} = 96,7\%$$

Berdasarkan perhitungan di atas, dapat diketahui bahwa persentase keberhasilan sistem berdasarkan pengujian fungsionalitas fitur adalah 96,7%. Dengan demikian, dapat disimpulkan bahwa Sistem Informasi Pendaftaran Santri Baru di Islamic Center Al-Hazza berjalan dengan baik dan sesuai dengan kebutuhan yang diharapkan pengguna.

4.5.4 Evaluasi Kegagalan

Pada hasil akhir pengujian User Acceptance Test (UAT), dari total 91 pengujian yang dilakukan, ditemukan tiga kegagalan dengan tingkat kegagalan sebesar 3,3%. Kegagalan ini terbagi menjadi dua dari sisi admin dan satu dari sisi calon santri. Penjelasan rinci mengenai kegagalan tersebut adalah sebagai berikut:

1. Kelola Pendaftaran (Admin): Admin mengalami kesulitan dalam mengakses halaman kelola pendaftaran saat menggunakan perangkat handphone. Kendala ini disebabkan oleh tampilan antarmuka yang tidak sepenuhnya responsif pada perangkat mobile, sehingga menghambat admin dalam melakukan pengelolaan pendaftaran dengan efisien.
2. Cetak Data Calon Santri (Admin): Kesulitan terjadi ketika admin mencoba mencetak data calon santri menggunakan perangkat handphone. Masalah ini muncul karena format dan fungsi pencetakan tidak dioptimalkan untuk perangkat mobile, yang menyebabkan ketidakmampuan admin untuk mencetak data dengan lancar.
3. Cetak Bukti Pendaftaran (Calon Santri): Calon santri mengalami kegagalan dalam memuat halaman cetak bukti pendaftaran. Hal ini terjadi karena responden menggunakan jaringan Wi-Fi umum yang tidak stabil selama pengujian, mengakibatkan kegagalan memuat halaman tersebut.

Untuk mengatasi masalah ini, direkomendasikan untuk meningkatkan responsivitas antarmuka sistem pada perangkat mobile, khususnya dalam fitur kelola pendaftaran dan pencetakan data. Selain itu, perlu dilakukan pengujian tambahan untuk memastikan fungsi cetak dapat berjalan dengan baik pada berbagai jenis jaringan internet.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

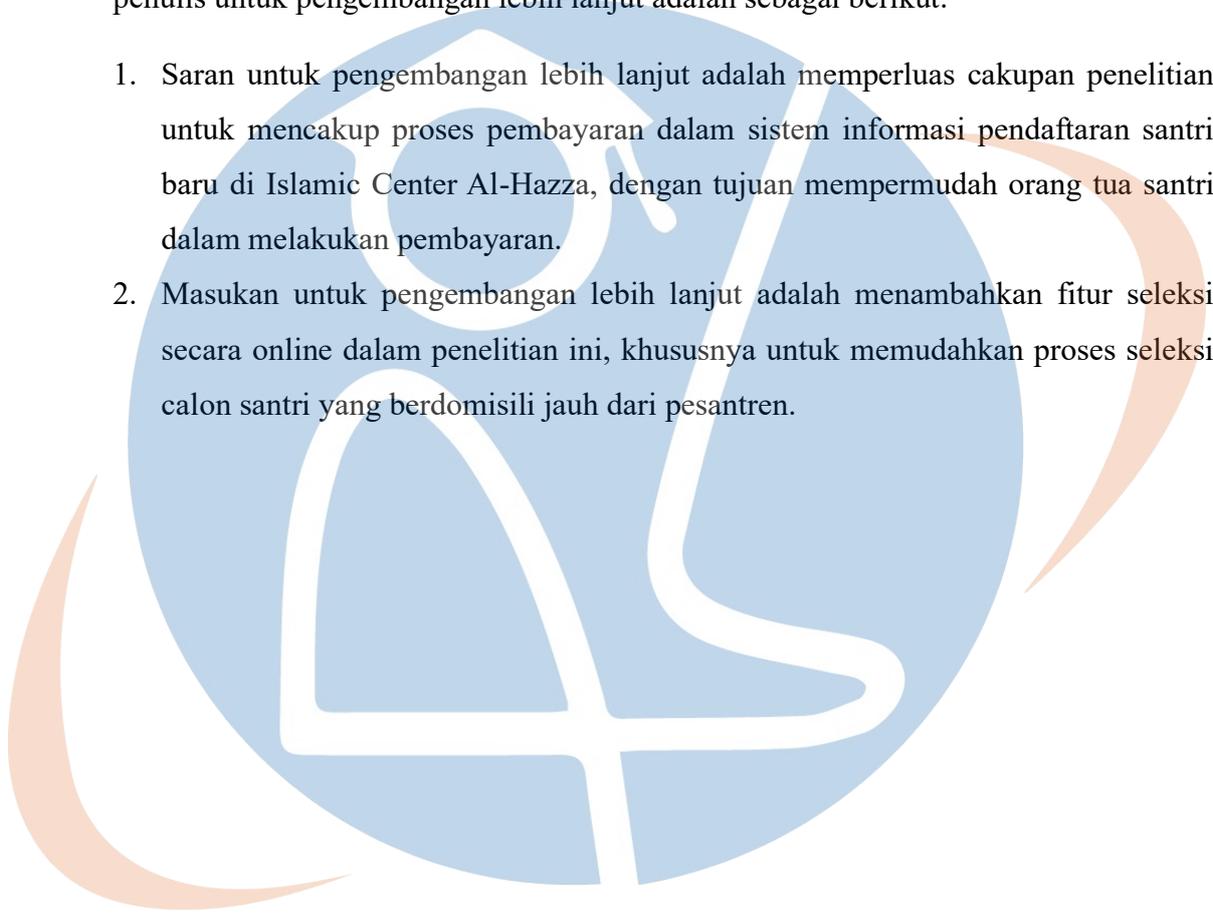
Berdasarkan hasil penelitian tugas akhir berjudul “Rancang Bangun Sistem Informasi Pendaftaran Santri Baru Menggunakan Framework Laravel: Studi Kasus di Islamic Center Al-Hazza,” dapat disimpulkan beberapa hal berikut:

1. Rancang bangun sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza menggunakan framework Laravel mencakup beberapa tahapan penting. Pertama, dilakukan analisis terhadap sistem yang sedang berjalan dan identifikasi permasalahan yang ada. Selanjutnya, dilakukan analisis kebutuhan secara menyeluruh untuk menentukan fitur-fitur yang diperlukan. Teknologi yang dipilih untuk pengembangan sistem adalah *framework Laravel*, yang memanfaatkan konsep *Model-View-Controller (MVC)* serta menggunakan metode *extreme programming*. Penggunaan Diagram UML memberikan gambaran visual mengenai struktur dan hubungan antarkomponen dalam sistem. Implementasi rancangan ini dilakukan melalui pengembangan sistem informasi dengan *Laravel*, diikuti oleh tahap pengujian yang komprehensif, termasuk *Black Box Testing* dan *User Acceptance Test (UAT)*. Keseluruhan proses ini dirancang untuk memastikan bahwa sistem informasi pendaftaran santri baru yang dikembangkan sesuai dengan kebutuhan dan beroperasi dengan baik sesuai harapan.
2. Berdasarkan hasil dari *User Acceptance Test (UAT)* yang mencapai persentase 96,7%, ini menunjukkan bahwa sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza berjalan sesuai dengan harapan. Evaluasi ini mengkonfirmasi bahwa sistem yang dibangun dapat beroperasi dengan baik dan memenuhi harapan serta kebutuhan yang telah ditetapkan.

5.2 Saran

Berdasarkan hasil dari penelitian rancang bangun sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza, terdapat beberapa saran yang dapat dipertimbangkan sebagai panduan untuk pengembangan lebih lanjut. Masukan dari penulis untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Saran untuk pengembangan lebih lanjut adalah memperluas cakupan penelitian untuk mencakup proses pembayaran dalam sistem informasi pendaftaran santri baru di Islamic Center Al-Hazza, dengan tujuan mempermudah orang tua santri dalam melakukan pembayaran.
2. Masukan untuk pengembangan lebih lanjut adalah menambahkan fitur seleksi secara online dalam penelitian ini, khususnya untuk memudahkan proses seleksi calon santri yang berdomisili jauh dari pesantren.



STT - NF

DAFTAR PUSTAKA

- [1] A. D. Mutiara* and A. Alimni, “Lembaga Pendidikan Pesantren,” *JIM J. Ilm. Mhs. Pendidik. Sej.*, vol. 8, no. 2, pp. 426–439, 2023, doi: 10.24815/jimps.v8i2.23921.
- [2] A. A. Irawan and N. Neneng, “Sistem Informasi Penerimaan Siswa Baru Berbasis Web (Studi Kasus Sma Fatahillah Sidoharjo Jati Agung Lampung Selatan),” *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 1, no. 2, pp. 245–253, 2021, doi: 10.33365/jatika.v1i2.620.
- [3] E. B. Pratama and A. Hendini, “Implementasi Extreme Programming Pada Perancangan SIMRS (Sistem Informasi Manajemen Rumah Sakit),” *J. Khatulistiwa Inform.*, vol. 10, no. 2, pp. 107–112, 2022, doi: 10.31294/jki.v10i2.14159.
- [4] H. Rasikhah and A. R. Adriansyah, “Perancangan dan Implementasi Booking System Lapangan menggunakan Framework MVC berbasis Web,” *J. Inform. Terpadu*, vol. 8, no. 1, pp. 08–12, 2022, doi: 10.54914/jit.v8i1.384.
- [5] E. Febriani, S. Adisuwiryo, and D. Savitri, “Design of sales information system based on website at Amonyu Shop,” *AIP Conf. Proc.*, vol. 2485, no. 1, pp. 124–132, 2023, doi: 10.1063/5.0104929.
- [6] A. L. I. Susanto and Y. Rahardja, “Perancangan dan Implementasi Peminjaman Ruang FTI UKSW Salatiga Berbasis Web Menggunakan Framework Laravel,” *J. Tek. Inform. dan Sist. Inf.*, vol. 9, no. 4, pp. 2720–2735, 2022, [Online]. Available: <http://jurnal.mdp.ac.id>
- [7] M. R. Hanif and K. Panji, “Analisis Dan Perancangan Sistem Informasi Pembayaran Sekolah Berbasis Extreme Programming Menggunakan Framework MVC,” *J. Inform. Terpadu*, vol. 9, no. 1, pp. 60–67, 2023, [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/JIT>
- [8] A. Rinjani and S. Munir, “Perancangan Sistem Pengelola Inventaris berbasis Web menggunakan Framework MVC,” *J. Inform. Terpadu*, vol. 8, no. 1, pp. 01–07, 2022, doi: 10.54914/jit.v8i1.377.
- [9] P. T. I. Permana and A. N. Sihanato, “Implementasi Arsitektur MVC Dalam

- Pengembangan Aplikasi Customer Relationship Portal,” *J. Teknol. Inf.*, vol. 10, no. 1, pp. 50–57, 2024, doi: 10.52643/jti.v10i1.2344.
- [10] R. Hermiati, A. Asnawati, and I. Kanedi, “Pembuatan E-Commerce Pada Raja Komputer Menggunakan Bahasa Pemrograman Php Dan Database Mysql,” *J. Media Infotama*, vol. 17, no. 1, pp. 54–66, 2021, doi: 10.37676/jmi.v17i1.1317.
- [11] T. Ardiansah, Y. Rahmanto, and Z. Amir, “Penerapan Extreme Programming Dalam Sistem Informasi Akademik SDN Kuala Teladas,” *J. Inf. Technol. Softw. Eng. Comput. Sci.*, vol. 1, no. 2, pp. 44–51, 2023, [Online]. Available: <https://doi.org/10.58602/itsecs.v1i2.25>
- [12] T. Arianti, A. Fa’izi, S. Adam, and Mira Wulandari, “Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram Uml (Unified Modelling Language),” *J. Ilm. Komput. ...*, vol. 1, no. 1, pp. 19–25, 2022, [Online]. Available: <https://journal.polita.ac.id/index.php/politati/article/view/110/88>
- [13] R. Maha, P. Umbu, S. M. Nalle, P. A. Saputra, and G. Feoh, “Implementasi Use Case Diagram Dan Activity Diagram Pada Perancangan Aplikasi Pembelajaran Bahasa Inggris Berbasis Android Di Sdn 5 Gulingan,” vol. 6, pp. 407–418, 2023.
- [14] A. S. Hardiansyah, Meri Hendayani, Ian Amukti Herlambang, Andhika Nove Rezki, “Implementasi Black Box Testing Pada Website,” *J. Ilmu Komput. dan Sci.*, vol. 1, no. 01, pp. 135–148, 2022.
- [15] I. Wahyudi and F. Alameka, “Analisis Blackbox Testing Dan User Acceptance Testing Terhadap Sistem Informasi Solusimedsosku,” *J. Teknosains Kodepena*, vol. 04, no. 01, pp. 1–9, 2023.

STT - NF