

## PENERAPAN APLIKASI PENDAFTARAN SISWA BARU UNTUK MENINGKATKAN PENGALAMAN CALON SISWA DI MTS TARBIYATUL MUBTADIIN WILALUNG

Hendra Dwi Kurniawan<sup>1</sup>, Ahmad Abdul Chamid<sup>3</sup>, Esti Wijayanti<sup>3</sup>

<sup>1,2,3</sup> Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muria Kudus  
Email: <sup>1</sup>201951136@std.umk.ac.id, <sup>2</sup>abdul.chamid@umk.ac.id, <sup>3</sup>esti.wijayanti@umk.ac.id

(Naskah masuk: dd mmm yyyy, diterima untuk diterbitkan: dd mmm yyyy)

### Abstrak

Penelitian ini mengembangkan dan menguji aplikasi PPDB MTs Tarbiyatul Mubtadiin Wilalung untuk mempermudah dan meningkatkan efisiensi proses pendaftaran serta pengelolaan data calon siswa. Metode waterfall digunakan dalam pengembangan aplikasi ini, meliputi tahap analisis, desain, implementasi, pengujian, dan implementasi. Aplikasi dilengkapi dengan fitur-fitur penting seperti halaman utama, halaman registrasi calon siswa, halaman akun calon siswa, dan halaman dashboard untuk panitia. Pengujian menggunakan metode blackbox dilakukan terhadap fitur-fitur utama aplikasi. Hasil pengujian menunjukkan bahwa aplikasi berhasil menampilkan halaman-halaman yang diharapkan dan fitur-fitur berfungsi sesuai rencana. Aplikasi ini memberikan kemudahan bagi calon siswa dalam pendaftaran online, memungkinkan panitia mengelola data calon siswa dengan efisien, serta meningkatkan transparansi dan akurasi dalam proses seleksi. Aplikasi ini diharapkan dapat meningkatkan efektivitas dan efisiensi PPDB serta memberikan manfaat bagi semua pihak yang terlibat dalam proses tersebut.

**Kata kunci:** PPDB, waterfall, web

## IMPLEMENTATION OF THE NEW STUDENT REGISTRATION APPLICATION TO IMPROVE PROSPECTIVE STUDENT EXPERIENCE AT MTS TARBIYATUL MUBTADIIN WILALUNG

### Abstract

*This research aims to develop and test the application for the Admission of New Students (PPDB) of MTs Tarbiyatul Mubtadiin Wilalung in order to facilitate and improve the efficiency of the registration process and student data management. The waterfall method was employed in the development, encompassing analysis, design, implementation, testing, and deployment stages. The application includes essential features such as a homepage with registration and login buttons, a news section created by the committee, a registration page for prospective students with comprehensive forms, an account page for students to upload documents and print registration proofs, and a dashboard page for the committee with a table displaying PPDB participants and management features. Blackbox testing was conducted on the main features of the application, including the homepage, student registration page, login, student account page, and dashboard. The test results demonstrated that the application successfully displayed the expected pages and the features functioned as planned. This research produced a PPDB application that can assist MTs Tarbiyatul Mubtadiin Wilalung in the admission process for new students. The application provides convenience for prospective students to register online, enables the committee to manage student data more efficiently, and improves transparency and accuracy in the selection process.*

**Keywords:** PPDB, waterfall, web

## 1. PENDAHULUAN

Perkembangan teknologi dalam beberapa dekade terakhir telah memberikan dampak yang signifikan pada berbagai aspek kehidupan manusia. Revolusi digital telah mengubah cara kita berkomunikasi, bekerja, belajar, dan berinteraksi dengan dunia di sekitar kita (Salamah, 2020).

Kemajuan teknologi telah memberikan banyak manfaat dan memperluas batas-batas kemungkinan yang sebelumnya sulit dijangkau. Dampak terbesar dari perkembangan teknologi adalah kemudahan akses informasi (Hidayat, Muttaqin dan Djamaludin, 2020). Internet telah menjadi sumber pengetahuan global yang tak terbatas. Dengan hanya beberapa kali klik, kita dapat mengakses informasi dari

berbagai sumber, menjelajahi jutaan artikel, buku, dan konten multimedia (Severance, 2018). Kemampuan ini telah mengubah cara kita belajar, melakukan riset, dan mengakses berita dan informasi terkini.

Teknologi juga telah mengubah cara kita berkomunikasi. Dengan adanya *platform* komunikasi seperti email, pesan instan, dan media sosial, kita dapat terhubung dengan orang-orang di seluruh dunia dengan mudah. Jarak dan waktu tidak lagi menjadi penghalang dalam berkomunikasi dan berkolaborasi. Teknologi juga telah memungkinkan pengembangan aplikasi *video call* dan konferensi *online*, memungkinkan pertemuan tatap muka jarak jauh tanpa batasan geografis. Dalam dunia bisnis, perkembangan teknologi juga telah membawa transformasi besar. Otomatisasi proses bisnis (Ozen dan Simsek, 2019), penggunaan sistem manajemen keuangan terintegrasi (Sonata, 2019), dan analisis data yang canggih telah memungkinkan efisiensi yang lebih tinggi (Sholeh, Rachmawati dan Cahyo, 2022) dan pengambilan keputusan yang lebih akurat (Nugroho, 2020). Teknologi juga telah membuka peluang baru dalam pemasaran digital, *e-commerce*, dan pembangunan aplikasi berbasis web atau *mobile*.

Tidak hanya dalam kehidupan sehari-hari, perkembangan teknologi juga telah mengubah dunia industri, seperti manufaktur, transportasi, energi, dan kesehatan. Teknologi canggih seperti kecerdasan buatan, robotika, dan *Internet of Things* (IoT) telah menghadirkan sistem yang lebih efisien, otomatis, dan terhubung. Inovasi ini meningkatkan produktivitas, mengurangi biaya, dan mengoptimalkan penggunaan sumber daya (Tserpes, 2019). Namun, bersamaan dengan manfaatnya, perkembangan teknologi juga menimbulkan tantangan baru. Perlindungan data pribadi, keamanan *cyber*, dan pengaruh sosial media adalah beberapa contoh isu yang muncul seiring dengan kemajuan teknologi (Wiedeman, 2018). Oleh karena itu, penting untuk terus memantau, mengatur, dan mengimplementasikan teknologi dengan bijak, memperhatikan dampaknya pada masyarakat dan lingkungan.

Perkembangan teknologi yang pesat telah memberikan dampak yang signifikan pada sistem PPDB (Penerimaan Peserta Didik Baru) di berbagai lembaga pendidikan, termasuk MTS Tarbiyatul Muhtadiin Wilalung. Teknologi telah membawa perubahan yang positif dan revolusioner dalam proses pendaftaran siswa baru, mempermudah akses, meningkatkan efisiensi, dan meningkatkan pengalaman pengguna secara keseluruhan (Rachmat dan Arman, 2020). Pengaruh utama dari perkembangan teknologi pada sistem PPDB adalah kemunculan aplikasi berbasis web. Dengan adanya aplikasi PPDB berbasis web, calon siswa dan orang tua dapat mengakses dan mengisi *formulir* pendaftaran secara *online*. Hal ini menghilangkan ketergantungan pada proses manual dan memberikan

kemudahan akses yang lebih luas, tidak terbatas oleh batasan geografis atau waktu (Puspita, Alkhalifi dan Basri, 2021). Calon siswa dapat mengisi *formulir* kapan saja dan di mana saja, asalkan terhubung ke internet. Ini juga meminimalisir kesalahan pengisian dan mempercepat proses verifikasi data.

Teknologi juga memungkinkan integrasi dengan sistem basis data yang canggih. Data calon siswa dapat disimpan dan dikelola dengan lebih efisien dan terstruktur (Agustina dan Mandela, 2022). Sistem basis data dapat memberikan kemampuan untuk menyimpan informasi seperti NISN, nama lengkap, tempat lahir, tanggal lahir, email, nomor HP, dan dokumen pendukung lainnya. Penggunaan basis data memungkinkan pencarian dan pengolahan data yang lebih cepat, pengurangan kesalahan manusia, dan menyediakan informasi yang lebih akurat dan terpercaya.

Penggunaan teknologi juga memungkinkan adopsi sistem notifikasi otomatis. Setelah proses pendaftaran selesai, sistem dapat mengirimkan notifikasi secara otomatis kepada calon siswa dan orang tua tentang status pendaftaran mereka, batas waktu pembayaran, atau dokumen yang masih belum lengkap. Notifikasi otomatis ini membantu meningkatkan komunikasi antara pihak sekolah dan calon siswa, serta meminimalisir kesalahan atau keterlambatan informasi.

Pengaruh lainnya adalah kemampuan untuk menyimpan dan mengelola dokumen-dokumen pendukung secara digital. Calon siswa dapat mengunggah *scan* ijazah, akta kelahiran, kartu keluarga, SKHUN, surat keterangan lulus, KIP/PKH/DTKS, surat penugasan, bukti bayar pendaftaran, dan foto melalui aplikasi. Dengan demikian, dokumen-dokumen tersebut dapat dengan mudah diakses, disimpan, dan diproses oleh Panitia PPDB.

Berdasarkan uraian latar belakang yang telah diberikan dan berbagai permasalahan yang teridentifikasi, rumusan masalah dalam penelitian ini meliputi tiga pertanyaan utama. Pertama, bagaimana sistem informasi penerimaan peserta didik baru secara daring di Dinas Pendidikan MTs Tarbiyatul Muhtadiin Wilalung Gajah Demak beroperasi? Kedua, bagaimana langkah-langkah merancang dan mengimplementasikan sistem informasi pendaftaran peserta didik baru di Madrasah Tsanawiyah Tarbiyatul Muhtadiin Wilalung Gajah Demak? Dan ketiga, bagaimana proses serta karakteristik dari sistem informasi pendaftaran peserta didik baru yang diterapkan di MTs Tarbiyatul Muhtadiin Wilalung Gajah Demak?

Berdasarkan penjelasan dalam latar belakang, batasan masalah dalam penelitian ini mencakup beberapa aspek. Pertama, sistem informasi penerimaan peserta didik baru secara daring melibatkan otomatisasi proses pendaftaran, seleksi, dan pengumuman hasil seleksi secara online. Kedua, sistem ini dibangun dengan menggunakan platform

berbasis website. Dan ketiga, fokus utama sistem ini adalah pada proses pendaftaran peserta didik baru secara daring.

Berdasarkan latar belakang dan rumusan masalah yang telah dijabarkan, tujuan utama penelitian ini adalah dua hal. Pertama, untuk meningkatkan validasi dan kepraktisan sistem penerimaan peserta didik baru yang menggunakan platform berbasis website di Madrasah Tsanawiyah Tarbiyatul Muhtadain Wilalung Gajah Demak. Kedua, untuk meningkatkan efektivitas penggunaan sistem penerimaan peserta didik baru berbasis website di MTs Tarbiyatul Muhtadain Wilalung Gajah Demak dalam mendukung proses penerimaan peserta didik baru secara lebih efisien.

## 2. Penelitian Terkait

Penelitian oleh (Rugiyana, 2023) dengan judul “Aplikasi PPDB Mandiri *Online* MTs Negeri 4 Gunungkidul dalam Memberikan Pelayanan pada Masyarakat”. Penelitian ini fokus pada pengembangan Aplikasi PPDB Mandiri *Online* untuk meningkatkan pelayanan kepada masyarakat selama periode penerimaan siswa baru di MTs Negeri 4 Gunungkidul. Metode studi kasus eksplorasi dengan pendekatan kualitatif digunakan untuk mendapatkan informasi mengenai kendala dan solusi yang terkait dengan pengembangan aplikasi PPDB. Hasil penelitian menunjukkan bahwa sejumlah pengembangan telah dilakukan dalam aplikasi PPDB Mandiri *Online* di MTs Negeri 4 Gunungkidul, seperti mengamati proses PPDB konvensional, mempelajari alurnya, dan mengadaptasikannya ke dalam bentuk digital untuk memudahkan calon siswa dalam mengumpulkan berkas dan memantau status pendaftaran mereka. Selain itu, fitur-fitur seperti penambahan nilai rapor, skor sertifikat prestasi, dan ranking juga diimplementasikan untuk mengatasi kendala yang dihadapi oleh calon siswa selama proses pendaftaran. Pemanfaatan Aplikasi PPDB Mandiri *Online* ini memberikan kemudahan bagi siswa dan staf guru dalam memantau perkembangan jumlah siswa baru dari masa ke masa.

Penelitian oleh (Yudahana, Riadi dan Elvina, 2023) berjudul “Perancangan Sistem Informasi Pendaftaran Peserta Didik Baru (PPDB) Berbasis Web Menggunakan Metode *Rapid Application Development* (RAD)”. RA Plus Rabbani adalah salah satu Raudhatul Atfal yang menjadi pilihan orang tua dalam mendaftarkan anak-anak mereka ke sekolah tingkat pra sekolah di bawah naungan Kementerian Agama Kabupaten Sikka. Saat ini, sekolah ini belum memanfaatkan sistem informasi sebagai penunjang manajemen sistem, terutama dalam hal pendaftaran. Proses Penerimaan Peserta Didik Baru (PPDB) merupakan proses manajemen tahunan yang jika dilakukan secara konvensional dapat menjadi tidak efektif. Oleh karena itu, diperlukan sistem terkomputerisasi untuk mengelola

PPDB secara efisien. Strategi yang akan diterapkan adalah metode *Rapid Application Development* (RAD) dalam pengembangan sistem. Alat bantu yang digunakan dalam perancangan sistem meliputi *Flowchart* dan *Data Flow Diagram* (DFD) untuk analisis alur sistem, *Entity Relationship Diagram* (ERD) untuk analisis desain *database*, dan *Figma* untuk analisis desain antarmuka sistem. Hasil penelitian menunjukkan bahwa RAD dapat mendukung pengembangan sistem dalam waktu yang lebih singkat dibandingkan dengan metode lain seperti *waterfall*, SDLC, dan *prototype*. Berdasarkan hasil tersebut, dapat disimpulkan bahwa desain sistem yang dibuat dapat memudahkan pengguna dalam menggunakan sistem tersebut, karena rancangannya menarik secara visual dan mudah dipahami oleh berbagai kalangan pengguna.

Penelitian oleh (Agustiana *et al.*, 2022) berjudul “Perancangan Sistem Informasi Penerimaan Peserta Didik Baru Berbasis Web Menggunakan *Framework Laravel*”. Dalam rangka mendukung upaya pemerintah dalam mengendalikan penyebaran virus Covid-19, banyak kegiatan yang sebelumnya dilakukan secara konvensional beralih ke bentuk *online*. Hal yang sama terjadi pada kegiatan Penerimaan Peserta Didik Baru (PPDB), di mana proses pendaftaran yang sebelumnya dilakukan dengan datang langsung ke sekolah, sekarang diubah menjadi *online*. Hal ini dilakukan untuk mengurangi risiko penyebaran virus Covid-19 serta mempercepat proses administrasi yang sebelumnya cenderung lambat karena penggunaan arsip fisik yang rentan hilang atau rusak. Untuk mengatasi tantangan tersebut, telah dibuat perancangan sistem informasi PPDB berbasis web. Metode pengembangan yang digunakan dalam proyek ini mengikuti pendekatan *System Development Life Cycle* (SDLC) dengan model *Waterfall*. Sistem informasi yang dirancang menggunakan desain UML dan *website* yang dikembangkan dengan bahasa pemrograman PHP, *framework Laravel*, *Bootstrap library*, dan *database MySQL*. Dengan adanya sistem informasi PPDB *online* di SD Negeri Nurul Salam ini, calon siswa dapat dengan mudah mengakses informasi dan melakukan proses pendaftaran secara *online*, sehingga proses administrasi PPDB menjadi lebih efektif dan efisien.

Penelitian yang dilakukan oleh (Fajriani dan Romahdoni, 2021) dengan judul “Rancang Bangun Aplikasi PPDB Studi Kasus Pondok Pesantren Diniyyah Putri Lampung”. Proses penerimaan siswa baru di Pondok Pesantren Diniyyah Putri Lampung (DPL) masih menggunakan sistem konvensional yang mengakibatkan calon siswa mengalami kesulitan dalam memperoleh informasi dan melakukan pendaftaran. Proses pendaftaran juga seringkali lambat karena kurangnya integrasi dan pengelolaan data yang efisien. Selain itu, penggunaan arsip fisik dalam proses ini juga rentan mengalami kerusakan atau kehilangan. Untuk

mengatasi permasalahan ini, dilakukan perancangan dan pembangunan aplikasi PPDB berbasis web. Metode yang digunakan dalam pembangunan sistem informasi ini adalah permodelan *Entity Relationship Diagram*. Hasil akhir dari perancangan dan pembangunan sistem informasi ini adalah sebuah web *mobile* yang memberikan kemudahan akses informasi dan proses pendaftaran, sehingga proses penerimaan siswa baru menjadi lebih efektif dan efisien.

Penelitian oleh (Rachmat dan Arman, 2020) dengan judul “Rancang Bangun Aplikasi Penerimaan Peserta Didik Baru Sekolah Swasta Berbasis Web”. Berdasarkan Surat Edaran Menteri Pendidikan dan Kebudayaan Nomor 4 Tahun 2020 yang mengatur pelaksanaan kebijakan pendidikan saat pandemi Covid-19, sekolah diharapkan untuk menyusun mekanisme Penerimaan Peserta Didik Baru (PPDB) yang mematuhi protokol kesehatan guna mencegah penyebaran virus. Hal ini termasuk menghindari kerumunan siswa dan orang tua di lingkungan sekolah. Untuk mendukung upaya pencegahan penyebaran Covid-19 yang dilakukan oleh pemerintah, telah dirancang sebuah aplikasi PPDB daring untuk Sekolah Swasta. Aplikasi ini memungkinkan calon peserta didik untuk mendaftar secara *online* tanpa perlu datang ke sekolah. Dalam penelitian ini, pengembangan sistem menggunakan model *waterfall*, sementara aplikasi dibangun menggunakan *framework PHP* dengan metode *Model, View, dan Controller* (MVC). Selain itu, protokol *Simple Mail Transfer Protocol* (SMTP) digunakan untuk mengirimkan notifikasi email kepada calon peserta didik dan Panitia PPDB.

### 3. Metodologi Penelitian

Metode *waterfall* dipilih sebagai metode pengembangan sistem dalam penelitian ini untuk mengatur dan mengelola langkah-langkah pengembangan aplikasi Penerimaan Peserta Didik Baru (PPDB) secara terstruktur dan berurutan. Metode *waterfall* mengikuti pendekatan linear, di mana setiap fase pengembangan dilakukan secara berurutan dan hanya dimulai setelah fase sebelumnya selesai sepenuhnya. Pendekatan ini sangat berguna dalam pengembangan aplikasi yang membutuhkan perencanaan dan analisis yang matang sebelum memasuki tahap implementasi.

#### 3.1 Analisis Kebutuhan

Dalam tahap analisis kebutuhan, dilakukan pengumpulan informasi dan pemahaman mendalam terhadap kebutuhan pengguna dan sistem yang akan dikembangkan. Tim pengembang melakukan wawancara, observasi, dan studi literatur untuk mengidentifikasi dan menggali informasi mengenai kebutuhan fungsional dan non-fungsional dari aplikasi yang akan dibangun. Selain itu, juga dilakukan identifikasi aktor, proses bisnis, dan aturan-aturan yang terkait. Hasil analisis kebutuhan

ini akan menjadi dasar untuk merancang dan mengembangkan sistem yang sesuai dengan kebutuhan pengguna, sehingga memastikan bahwa aplikasi PPDB yang dibangun akan efektif dan memenuhi tujuan yang diinginkan.

#### 3.2 Perancangan

Dalam tahap perancangan, dilakukan penjabaran rinci mengenai arsitektur sistem dan struktur komponen yang akan dibangun. Tim pengembang menerjemahkan kebutuhan pengguna menjadi desain yang terstruktur dan terorganisir. Dalam hal aplikasi PPDB, tahap perancangan mencakup pembuatan diagram UML seperti *use case diagram*, *class diagram*, dan *activity diagram* untuk menggambarkan interaksi antara komponen sistem, alur proses, dan hubungan antar entitas. Selain itu, desain antarmuka pengguna (*user interface*) juga dibuat agar aplikasi dapat digunakan dengan mudah dan intuitif. Pada tahap ini, perhatian juga diberikan pada aspek keamanan, skalabilitas, dan efisiensi sistem. Hasil dari tahap perancangan ini akan menjadi panduan bagi tim pengembang dalam mengimplementasikan sistem aplikasi PPDB secara lebih terstruktur dan terarah.

#### 3.3 Implementasi

Dalam tahap implementasi, konsep dan desain yang telah dibuat dalam tahap perancangan diwujudkan menjadi kode program yang dapat dijalankan. Tim pengembang menggunakan bahasa pemrograman dan *framework* yang sesuai untuk membangun komponen-komponen aplikasi PPDB, seperti pengelolaan *database*, logika bisnis, dan antarmuka pengguna. Selama proses implementasi, terjadi pengkodean, pengujian, dan *debugging* untuk memastikan bahwa aplikasi berfungsi sesuai dengan spesifikasi yang telah ditetapkan. Selain itu, dalam tahap ini juga dilakukan integrasi antara komponen-komponen yang telah dibangun sehingga membentuk sistem aplikasi PPDB yang utuh. Setelah tahap implementasi selesai, aplikasi siap untuk diuji lebih lanjut dalam tahap pengujian sebelum dapat digunakan secara efektif oleh pengguna.

#### 3.4 Pengujian

Dalam tahap pengujian, aplikasi PPDB menjalani serangkaian tes untuk memastikan bahwa semua fungsionalitasnya berjalan dengan baik dan memenuhi persyaratan yang telah ditetapkan. Tim pengujian melakukan pengujian unit untuk menguji setiap komponen dan fungsi secara terisolasi. Selanjutnya, dilakukan pengujian integrasi untuk memastikan bahwa komponen-komponen tersebut bekerja secara sinergis dan berinteraksi dengan baik satu sama lain. Pengujian sistem juga dilakukan untuk menguji kinerja keseluruhan aplikasi dalam situasi nyata, termasuk menangani beban yang tinggi dan skenario penggunaan yang kompleks. Selama

proses pengujian, *bug* dan kesalahan ditemukan, dicatat, dan diperbaiki. Tujuan utama dari tahap ini adalah memastikan bahwa aplikasi PPDB berfungsi dengan baik, responsif, dan dapat diandalkan sebelum dirilis kepada pengguna akhir.

### 3.5 Perawatan dan Dokumentasi

Dalam tahap perawatan dan dokumentasi, tim pengembang aplikasi PPDB bertanggung jawab untuk memelihara dan merawat sistem yang sudah berjalan dengan baik. Ini melibatkan pemantauan kinerja aplikasi, pemecahan masalah, dan peningkatan yang diperlukan untuk menjaga keandalan dan kualitas sistem. Selain itu, dokumentasi yang komprehensif tentang sistem juga disusun dan diperbarui secara berkala. Dokumentasi ini mencakup deskripsi lengkap tentang arsitektur sistem, alur kerja, fungsi dan fitur, serta petunjuk penggunaan yang jelas. Tujuan dari tahap ini adalah memastikan bahwa aplikasi PPDB tetap berjalan dengan baik dalam jangka panjang dan memberikan sumber daya yang diperlukan bagi pengguna dan pengembang yang terlibat dalam pemeliharaan dan pengembangan masa depan.

## 4. PEMBAHASAN

### 4.1 Konfigurasi Lingkungan Pengembangan

Pada tahap ini, persiapan peralatan pengembangan dilakukan untuk memastikan lingkungan pengembangan yang optimal. Beberapa langkah yang perlu dilakukan adalah mengunduh *Visual Studio Code* dari situs resminya dan menjalankan *installer*. Instruksi instalasi akan muncul di layar dan proses instalasi harus diselesaikan.

XAMPP perlu diunduh dari situs resminya dan *installer* juga perlu dijalankan. Selama instalasi, komponen yang akan diinstal harus dipilih dengan hati-hati. Pastikan *Apache*, *MySQL*, dan *PHP* dipilih karena mereka merupakan komponen penting dalam pengembangan aplikasi web. Setelah instalasi XAMPP selesai, konfigurasi *Apache* dan *MySQL* perlu diatur. Folder *root* dan konfigurasi *virtual host*, jika diperlukan, harus ditentukan untuk *Apache*. Untuk *MySQL*, *password* untuk akun *root* perlu diatur dan konfigurasi lainnya harus disesuaikan dengan kebutuhan. Setelah konfigurasi XAMPP selesai, *Apache* dan *MySQL* dapat dijalankan melalui panel kontrol XAMPP.

Untuk membuat aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung, diperlukan XAMPP versi 7 terbaru. Untuk mendapatkan versi ini, dapat mencarinya di folder *XAMPP for Windows* yang tersedia di halaman yang disediakan di SourceForge. Pemilihan versi XAMPP ini dilakukan karena aplikasi PPDB ini menggunakan *PHP 7* sebagai versi *PHP* yang digunakan.

*Visual Studio Code* dapat dengan mudah dibuka setelah diinstal, dan selanjutnya, ekstensi yang diperlukan untuk pengembangan aplikasi web

dapat diinstal. Dalam hal ini, terdapat beberapa ekstensi yang sangat berguna, seperti ekstensi *PHP* dan *MySQL*. Ekstensi *PHP* memungkinkan pengembang untuk menulis dan mengelola kode *PHP* dengan lebih efisien, termasuk fitur-fitur seperti penyorotan sintaks, saran kode, dan *debugging*. Ekstensi *MySQL* memungkinkan pengembang untuk berinteraksi dengan *database MySQL* secara langsung dari dalam lingkungan pengembangan, dengan menyediakan fitur-fitur seperti penjelajahan *database*, eksekusi *query*, dan pemantauan *performa*.

Dalam pembuatan Aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung, ekstensi *Visual Studio Code* menjadi bagian penting dalam meningkatkan produktivitas dan kualitas kode. Dengan menggunakan ekstensi *PHP*, pengembang dapat mengoptimalkan proses penulisan kode *PHP*, mengurangi kesalahan pengetikan, dan meningkatkan pemahaman terhadap kode yang sedang dikembangkan. Selain itu, ekstensi *MySQL* juga memberikan kemudahan dalam berinteraksi dengan *database*, memungkinkan pengembang untuk melihat dan memanipulasi data dengan lebih cepat dan efisien.

Salah satu ekstensi yang sangat membantu proses pembuatan Aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung adalah *PHP IntelliSense*, yang memberikan fitur *intellisense* yang canggih untuk bahasa pemrograman *PHP*. Ekstensi ini membantu dalam penulisan kode yang lebih efisien dan akurat dengan menyediakan saran kode, penyelesaian otomatis, dan dokumentasi yang terperinci.

Ekstensi *Prettier* merupakan alat yang sangat berguna dalam pengembangan aplikasi, terutama dalam hal mengatur tata letak dan *format* kode secara konsisten. Dengan menggunakan ekstensi ini, pengembang dapat dengan mudah menjaga gaya penulisan kode yang seragam dan profesional. *Prettier* bekerja dengan cara menganalisis struktur kode dan secara otomatis mengaplikasikan aturan penulisan yang telah ditentukan. Hal ini membantu menghemat waktu dan usaha dalam mengatur tata letak kode secara manual, sehingga fokus dapat lebih diarahkan pada pengembangan fitur dan logika aplikasi. Selain itu, *Prettier* juga mendukung berbagai bahasa pemrograman, termasuk *PHP*, sehingga dapat digunakan dalam proyek pengembangan aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung. Dengan konsistensi tata letak dan *format* kode yang dihasilkan oleh *Prettier*, tim pengembang dapat bekerja lebih efisien dan menghasilkan kode yang lebih mudah dipelihara dan diperbarui di masa depan.

Ekstensi *PHP IntelliSense* merupakan salah satu tambahan yang sangat bermanfaat dalam pengembangan aplikasi *PHP*. Dengan menggunakan ekstensi ini, pengembang akan merasakan peningkatan fitur penyelesaian kode yang sangat berguna. Fitur ini akan memberikan saran kode secara otomatis saat pengembang mengetik,

sehingga mempercepat proses penulisan kode dan mengurangi risiko kesalahan pengetikan. Selain itu, ekstensi ini juga menyediakan dokumentasi yang lebih baik, memberikan informasi rinci tentang fungsi, kelas, dan metode yang tersedia dalam kode PHP. Dengan adanya dokumentasi yang lengkap dan mudah diakses, pengembang dapat dengan cepat menemukan informasi yang dibutuhkan untuk mengembangkan aplikasi dengan lebih efisien. Selain itu, *PHP IntelliSense* juga menawarkan kemampuan navigasi yang lebih baik. Pengembang dapat dengan mudah melompat antara definisi fungsi, deklarasi variabel, dan penggunaan lainnya dalam kode PHP, sehingga memudahkan dalam memahami dan memodifikasi kode yang ada. Dengan fitur-fitur ini, pengembang dapat menulis kode PHP dengan lebih cepat, akurat, dan efisien, meningkatkan produktivitas dan kualitas dalam pengembangan aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung.

#### 4.2 Penyusunan Database

Proses penyusunan *database* pada Aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung dilakukan dengan bantuan *phpMyAdmin*, yang merupakan salah satu alat administrasi dan manajemen *database* yang berbasis web. Dengan menggunakan *phpMyAdmin*, pengembang dapat dengan mudah mengelola struktur *database*, tabel, dan relasi antara entitas. Pertama, pengembang mengakses *phpMyAdmin* melalui web browser dengan memasukkan informasi *login* yang valid. Setelah masuk, tampilan antarmuka *phpMyAdmin* memberikan berbagai opsi dan fitur yang memungkinkan pengembang untuk bekerja dengan *database*.

Langkah pertama adalah membuat *database* baru yang akan digunakan dalam Aplikasi PPDB. Dalam *phpMyAdmin*, pengembang dapat membuat *database* baru dengan menentukan nama *database* dan mengatur lainnya seperti pengkodean karakter. Setelah *database* dibuat, pengembang dapat memulai pembuatan tabel dengan menentukan nama tabel, kolom-kolom, dan tipe data untuk setiap kolom. Kolom-kolom ini mencerminkan atribut-atribut yang telah ditentukan dalam perancangan *database* sebelumnya. Pengembang juga dapat menetapkan kunci primer dan kunci asing untuk mengatur relasi antara tabel-tabel.

Selanjutnya, pengembang menggunakan *phpMyAdmin* untuk mengelola data dalam tabel-tabel. Mereka dapat menambahkan, mengubah, atau menghapus data dengan mudah melalui antarmuka yang intuitif. Pengembang juga dapat menjalankan *query* SQL untuk melakukan operasi yang lebih kompleks, seperti pemilihan data berdasarkan kriteria tertentu atau penggabungan data dari beberapa tabel.

#### 4.3 Pengaturan Awal CodeIgniter

Langkah awal dalam pengaturan awal *CodeIgniter* adalah dengan mengunduh *framework* tersebut dari situs resmi *CodeIgniter*. Setelah berhasil mengunduh *framework* *CodeIgniter*, tahap selanjutnya melibatkan pengaturan pada beberapa *File* konfigurasi yang ada.

*File* konfigurasi ini perlu disesuaikan agar sesuai dengan kebutuhan aplikasi yang sedang dikembangkan. Dalam *File* konfigurasi tersebut, pengembang dapat mengatur berbagai pengaturan penting seperti pengaturan *database*, pengaturan URL, pengaturan sesi, dan masih banyak lagi.

Salah satu *File* yang perlu diatur adalah *config.php*. Pada *File* ini, terdapat beberapa konfigurasi penting yang perlu disesuaikan, seperti pengaturan URL *base*, pengaturan *time zone*, dan konfigurasi enkripsi. Pengaturan ini akan mempengaruhi cara aplikasi berinteraksi dengan pengguna dan lingkungan pengembangan.

Selanjutnya, pengaturan dilakukan pada *File* *database.php*. *File* ini berisi konfigurasi untuk koneksi ke *database* yang akan digunakan oleh aplikasi. Pengembang perlu menentukan pengaturan seperti *hostname*, nama pengguna, kata sandi, dan nama *database* yang akan digunakan. Dengan mengonfigurasi *File* ini dengan benar, aplikasi dapat terhubung ke *database* yang sesuai dan dapat mengakses data yang diperlukan.

Pengaturan pada *File* *routes.php* juga penting dalam *CodeIgniter*. *File* ini menentukan bagaimana URL aplikasi akan diarahkan ke fungsi-fungsi dalam *controller*. Pengembang dapat menambahkan aturan-aturan rute khusus untuk mengatur alur navigasi dan menghubungkan URL dengan aksi-aksi yang tepat dalam aplikasi.

Pengaturan awal juga melibatkan pengaturan *File* *.htaccess*. *File* ini berperan penting dalam mengonfigurasi URL dan aturan-aturan server. Pengembang dapat menggunakan *File* *.htaccess* untuk mengatur *string* URL, mengaktifkan atau menonaktifkan fitur-fitur server seperti kompresi *gzip*, mengatur kebijakan *cache*, dan sebagainya.

#### 4.3 Pembuatan Model

Pada aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung, tahap pembuatan model adalah salah satu langkah penting dalam pengembangan aplikasi. Model bertanggung jawab untuk mengelola data dan logika bisnis aplikasi. Dalam aplikasi ini, beberapa model yang dibuat adalah *Auth\_model*, *Berita\_model*, *Files\_model*, *Panitia\_model*, dan *Siswa\_model*.

#### 4.4 Auth Model

*Auth\_model* bertanggung jawab untuk mengelola proses autentikasi dan otorisasi pengguna dalam aplikasi. Hal ini meliputi verifikasi *login*, pengelolaan sesi pengguna, dan pengaturan hak akses. Pada model ini terdapat beberapa fungsi yang digunakan untuk melakukan operasi pada tabel

database. Fungsi pendaftaran digunakan untuk menyimpan data pendaftaran ke dalam tabel yang ditentukan. Fungsi *get\_peserta* digunakan untuk mendapatkan data peserta berdasarkan ID tertentu. Selain itu, terdapat juga fungsi *auth\_siswa* dan *auth\_panitia* yang digunakan untuk melakukan proses autentikasi siswa dan panitia. Kedua fungsi ini akan memeriksa kecocokan *username* dan *password* yang dimasukkan oleh pengguna dengan data yang ada dalam tabel terkait. Jika data cocok, maka fungsi akan mengembalikan data pengguna tersebut. Jika tidak cocok, maka akan ditampilkan pesan kesalahan yang sesuai.

#### 4.5 Berita Model

*Berita\_model* berperan dalam pengelolaan data berita pada aplikasi. Model ini dapat memiliki metode untuk menambah, mengubah, menghapus, dan mengambil data berita dari database. Selain itu, model ini juga bisa memiliki metode untuk mengambil berita terbaru atau berita dengan kriteria tertentu. Model ini memiliki beberapa fungsi yang masing-masing memiliki tujuan tertentu. Fungsi *insert\_berita* digunakan untuk menyimpan data berita baru ke dalam tabel 'berita'. Fungsi *update\_berita* digunakan untuk memperbarui data berita berdasarkan ID tertentu. Fungsi *delete\_post* digunakan untuk menghapus berita berdasarkan ID tertentu. Fungsi *get\_berita* digunakan untuk mengambil semua data berita dari tabel 'berita'. Fungsi *getnew\_berita* digunakan untuk mengambil data berita secara terurut berdasarkan ID terbaru. Fungsi *single\_post* digunakan untuk mengambil data berita berdasarkan *slug* yang diberikan.

#### 4.6 Files Model

*Files\_model* digunakan untuk mengelola data *File* pada aplikasi. Model ini dapat memiliki fungsi untuk mengunggah *File*, menyimpan *metadata File* ke database, dan melakukan operasi lain terkait *File* seperti menghapus atau mengambil *File* berdasarkan kriteria tertentu. Model ini memiliki beberapa fungsi yang masing-masing memiliki tujuan tertentu. Fungsi *insert\_File* digunakan untuk menyimpan *File* ke dalam tabel yang ditentukan. Fungsi *check\_File* digunakan untuk memeriksa keberadaan *File* berdasarkan ID siswa dan kriteria tertentu, seperti *File* ijazah, pas foto, dan *File* pembayaran. Fungsi *update\_document* digunakan untuk memperbarui dokumen berdasarkan ID siswa.

#### 4.7 Panitia Model

*Panitia\_model* bertugas untuk mengelola data panitia dalam proses PPDB. Model ini bisa memiliki metode untuk menambah, mengubah, menghapus, dan mengambil data panitia dari database. Selain itu, model ini juga bisa menyediakan fungsi untuk mengambil daftar panitia berdasarkan peran tertentu atau kriteria lain yang relevan. Model ini memiliki beberapa fungsi yang masing-masing memiliki

tujuan tertentu. Fungsi *set\_status* digunakan untuk mengatur status siswa berdasarkan ID siswa dan status yang diberikan. Fungsi ini juga memberikan pesan sukses atau *error* terkait dengan pengaturan status. Fungsi *countall\_peserta* digunakan untuk menghitung jumlah total peserta yang terdaftar. Fungsi *count\_diterima* digunakan untuk menghitung jumlah peserta yang diterima. Fungsi *count\_ditolak* digunakan untuk menghitung jumlah peserta yang ditolak. Selain itu, model ini juga memiliki fungsi *set\_ppdb* untuk mengatur status PPDB (buka atau tutup) berdasarkan ID status, dan fungsi *ppdb\_status* untuk mendapatkan status PPDB.

#### 4.8 Siswa Model

*Siswa\_model* berfungsi untuk mengelola data siswa yang terkait dengan proses PPDB. Model ini bisa memiliki metode untuk menambah, mengubah, menghapus, dan mengambil data siswa dari database. Model ini memiliki beberapa fungsi yang masing-masing memiliki tujuan tertentu. Fungsi *get\_siswa* digunakan untuk mendapatkan data siswa berdasarkan ID. Fungsi *getall\_siswa* digunakan untuk mendapatkan semua data siswa yang belum memiliki status atau status "none" dan sudah mengunggah *File* pembayaran. Fungsi *getsiswa\_diterima* digunakan untuk mendapatkan semua data siswa yang sudah diterima. Fungsi *getsiswa\_ditolak* digunakan untuk mendapatkan semua data siswa yang ditolak. Selain itu, model ini juga memiliki fungsi *update\_proFile* untuk memperbarui profil siswa berdasarkan ID, dan fungsi *delete\_siswa\_ditolak* untuk menghapus data siswa yang ditolak.

#### 4.9 Pembuatan Controller

Setelah model dibuat, tahap berikutnya adalah pembuatan *Controller*. *Controller* berperan sebagai penghubung antara model dan tampilan (*view*), serta bertanggung jawab untuk mengatur logika aplikasi. Dalam konteks aplikasi ini, beberapa *controller* yang dibuat adalah *ControllerAuth*, *ControllerDashboard*, *ControllerHome*, dan *ControllerPendaftaran*.

*ControllerAuth* bertanggung jawab untuk mengelola proses autentikasi pengguna dalam aplikasi. Tugasnya mencakup hal-hal seperti menampilkan halaman *login*, memeriksa keabsahan kredensial pengguna, dan mengelola sesi pengguna setelah berhasil *login*. Selain itu, *ControllerAuth* juga dapat memiliki fungsi-fungsi lain seperti *logout*, registrasi, dan pengaturan hak akses pengguna. Dalam konstruktor kelas *ControllerAuth*, terdapat pemanggilan *helper form*, *library form\_validation*, dan model *auth\_model* yang diperlukan untuk menjalankan fungsi-fungsi dalam kelas ini.

Metode *index* digunakan untuk menangani halaman *login*. Pertama, aturan validasi *form* diatur menggunakan *set\_rules()* dari *library form\_validation*. Jika validasi *form* gagal, halaman

*login (signin.php)* akan ditampilkan dengan memuat variabel data seperti judul halaman. Selain itu, *session* yang ada akan dihancurkan menggunakan *sess\_destroy()* untuk memastikan tidak ada sesi yang tersimpan. Jika validasi *form* berhasil, fungsi *secure\_input()* akan dipanggil untuk mengamankan input dari *form*. Kemudian, data *username* dan *password* yang telah diamankan akan digunakan untuk melakukan autentikasi siswa dan autentikasi panitia menggunakan fungsi *auth\_siswa()* dan *auth\_panitia()* dari model *auth\_model*. Jika autentikasi siswa berhasil, *session* akan diset dengan informasi yang diperlukan dan diarahkan ke halaman *dashboard*. Jika autentikasi panitia berhasil, *session* akan diset dengan informasi yang diperlukan dan diarahkan ke halaman *dashboard*. Jika autentikasi gagal, pengguna akan diarahkan kembali ke halaman *login*.

Metode *register\_panitia\_ppdb* digunakan untuk menangani pendaftaran panitia PPDB. Aturan validasi *form* diatur menggunakan *set\_rules()* dari *library\_form\_validation*. Jika validasi *form* gagal, halaman pendaftaran panitia (*panitia.php*) akan ditampilkan dengan memuat variabel data seperti judul halaman. Jika validasi *form* berhasil, fungsi *secure\_input()* akan dipanggil untuk mengamankan input dari *form*. Kemudian, data nama, NUPTK, dan *password* yang telah diamankan akan digunakan untuk melakukan pendaftaran panitia menggunakan fungsi *pendaftaran()* dari model *auth\_model*. Setelah pendaftaran berhasil, pesan sukses akan disimpan dalam *session\_flashdata* dan pengguna akan diarahkan ke halaman *login*.

Metode *logout* digunakan untuk melakukan proses *logout*. Sesi yang ada akan dihancurkan menggunakan *sess\_destroy()* dan pengguna akan diarahkan kembali ke halaman *login*.

*ControllerDashboard* mengatur tampilan dan logika bisnis pada halaman *dashboard* aplikasi. Fungsinya termasuk menampilkan data penting seperti statistik, grafik, atau informasi terkini. Selain itu, *ControllerDashboard* juga mengelola aksi-aksi pengguna seperti mengirim permintaan data, mengubah pengaturan, atau memproses tindakan khusus lainnya.

Pada konstruktor kelas *ControllerDashboard*, dilakukan beberapa tindakan awal, seperti memeriksa *session userdata* masuk. Jika tidak ada *session userdata* yang masuk, pengguna akan diarahkan ke halaman *home*. Konstruktor juga membuat fungsi *secure\_input()* untuk membersihkan dan melindungi input pengguna dari serangan seperti *Cross-Site Scripting (XSS)*. Selain itu, beberapa paket/model yang diperlukan oleh *controller dashboard*, seperti *siswa\_model*, *panitia\_model*, *Files\_model*, dan *berita\_model*, juga dimuat di dalam konstruktor.

Fungsi *index()* digunakan untuk menampilkan halaman utama *dashboard*. Logika dalam fungsi ini berbeda tergantung pada akses pengguna. Jika

pengguna memiliki akses 2 (siswa), maka halaman yang ditampilkan akan berisi *form* untuk melengkapi data dan mengunggah dokumen yang diperlukan. Data yang dimasukkan akan divalidasi dan disimpan ke dalam database jika validasi berhasil. Jika pengguna bukan siswa, halaman utama *dashboard* akan menampilkan informasi dan fitur-fitur yang sesuai. Seperti halnya pada fungsi sebelumnya, data yang dimasukkan akan divalidasi dan disimpan ke dalam database jika validasi berhasil.

Selanjutnya, terdapat fungsi *update\_berita(\$id)* yang digunakan untuk mengupdate berita berdasarkan ID yang diberikan sebagai parameter. Fungsi ini juga melakukan validasi input pengguna dan menyimpan data berita yang telah diupdate ke dalam database jika validasi berhasil.

Terdapat beberapa fungsi lainnya seperti *ppdb()*, *ppdb\_diterima()*, dan *ppdb\_ditolak()* yang masing-masing digunakan untuk menampilkan data peserta yang mendaftar pada PPDB (Penerimaan Peserta Didik Baru), data peserta yang diterima pada PPDB, dan data peserta yang ditolak pada PPDB. Fungsi-fungsi tersebut akan memuat halaman *view* dan mengirimkan data yang diperlukan ke *view* tersebut.

*ControllerHome* mengelola tampilan halaman utama (*home*) aplikasi. Tugasnya meliputi menampilkan informasi umum, berita terbaru, dan fitur-fitur penting lainnya. Selain itu, *ControllerHome* juga mengelola aksi-aksi pengguna seperti mengirim pesan, menghubungi sekolah, atau mengakses fitur-fitur lain yang terkait dengan halaman utama.

Dalam konstruktor kelas *ControllerHome*, terdapat pemanggilan paket/model yang diperlukan, yaitu *berita\_model* dan *panitia\_model*. Tindakan ini dilakukan saat kelas pertama kali dipanggil.

Fungsi *index()* digunakan untuk menampilkan halaman utama aplikasi web. Pada fungsi ini, beberapa data seperti judul halaman, berita terbaru, dan status PPDB (Penerimaan Peserta Didik Baru) dimasukkan ke dalam variabel *\$data*. Nilai-nilai tersebut diperoleh melalui pemanggilan fungsi dari model *berita\_model* dan *panitia\_model*. Setelah itu, halaman utama ditampilkan dengan memuat tiga *view* yang terpisah, yaitu *header*, *home*, dan *footer*.

Terdapat fungsi *single\_post(\$slug)* yang digunakan untuk menampilkan halaman *single post* berdasarkan *slug* yang diberikan sebagai parameter. Pada fungsi ini, data seperti judul halaman, berita terbaru, dan data *post* yang sesuai dengan *slug* diperoleh melalui pemanggilan fungsi dari model *berita\_model*. Data tersebut dimasukkan ke dalam variabel *\$data* dan halaman *single post* ditampilkan dengan memuat *view* yang sesuai.

*ControllerPendaftaran* bertanggung jawab atas proses pendaftaran dalam aplikasi PPDB. Tugasnya mencakup menampilkan *formulir* pendaftaran, menerima input pengguna, memvalidasi data, dan menyimpan data pendaftaran ke dalam database



menggunakan model yang sesuai. Selain itu, *ControllerPendaftaran* juga dapat menyediakan fungsi-fungsi lain seperti melihat status pendaftaran, mengunduh *formulir*, atau mengelola aksi-aksi terkait dengan pendaftaran.

Dalam konstruktor kelas *ControllerPendaftaran*, terdapat pemanggilan *helper form*, *library form\_validation*, serta model *auth\_model* dan *Files\_model*. Tindakan ini dilakukan saat kelas pertama kali dipanggil.

Fungsi *index()* digunakan untuk menampilkan *form* pendaftaran jika validasi gagal, atau melakukan proses pendaftaran jika validasi berhasil. Pada fungsi ini, terdapat pengaturan aturan validasi untuk setiap input *form* yang ada di *view* menggunakan *set\_rules()* dari *library form\_validation*. Jika validasi gagal, halaman *form* pendaftaran ditampilkan dengan memuat *view auth/pendaftaran*. Jika validasi berhasil, dilakukan proses pendaftaran dengan melakukan sanitasi data yang diperoleh dari *form* menggunakan fungsi *secure\_input()*.

Data yang sudah divalidasi dan di sanitasi tersebut disimpan dalam variabel *\$data* dalam bentuk *array*. Selanjutnya, dilakukan pemanggilan fungsi dari model *auth\_model* dan *Files\_model* untuk menyimpan data pendaftar ke dalam tabel dan menyimpan *id* siswa ke tabel *File* sebagai referensi. Selain itu, kode pendaftaran juga disimpan dalam *session* untuk membuat laporan sementara, dan pengguna diarahkan ke halaman *download* laporan sementara.

Fungsi *cetak(\$id)* digunakan untuk mencetak laporan sementara. Pada fungsi ini, pertama kali dilakukan pemeriksaan apakah *session* kode sudah ada. Jika tidak, pengguna akan diarahkan kembali ke halaman utama. Jika *session* kode tersedia, data peserta diambil dari tabel *tb\_peserta* menggunakan fungsi dari model *auth\_model* dan disimpan dalam variabel *\$data*. Selanjutnya, *view auth/cetak-sementara* ditampilkan dengan memuat data peserta yang telah diambil.

#### 4.10 Pembuatan View

Dalam aplikasi web menggunakan kerangka kerja *CodeIgniter*, *view* digunakan untuk menampilkan tampilan halaman kepada pengguna. Pada aplikasi yang sedang dibahas, terdapat beberapa *view* yang digunakan untuk menampilkan berbagai halaman seperti halaman utama, halaman pendaftaran, dan halaman cetak laporan sementara.

Proses pembuatan *view* pada aplikasi melibatkan penggunaan *Markup HTML*, dan terkadang juga dapat melibatkan penggunaan *CSS* dan *JavaScript* untuk mengatur tampilan dan interaksi dengan pengguna. Semua komponen-komponen tersebut digabungkan menjadi tampilan web yang dinamis dan terstruktur. Berikut adalah nama komponen-komponen yang digunakan.

Ada 11 *File view* yang digunakan dalam Aplikasi PPDB MTs Tarbiyatul Mubtadiin

Wilalung. *File view* tersebut dipanggil oleh *controller* yang sudah dibahas sebelumnya. *File view* tersebut antara lain:

a. *File auth/cetak-sementara.php*

*View* ini digunakan untuk menampilkan laporan sementara setelah proses pendaftaran berhasil. Halaman ini berisi informasi identitas peserta yang telah terdaftar. *File* ini berisi kode HTML untuk tampilan laporan sementara.

b. *File auth/panitia.php*

*View* ini digunakan untuk menampilkan halaman *login* bagi panitia. Halaman ini berisi *form* untuk memasukkan *username* dan *password* panitia. *File* ini berisi kode HTML untuk tampilan halaman *login*.

c. *File auth/pendaftaran.php*

*View* ini digunakan untuk menampilkan *form* pendaftaran. Halaman ini berisi elemen-elemen *form* yang sesuai dengan aturan validasi yang telah ditentukan. Data pada *form* akan dikirim ke *controller* saat pengguna menekan tombol *submit*. *File* ini berisi kode HTML untuk tampilan halaman pendaftaran.

d. *File auth/signin.php*

*View* ini digunakan untuk menampilkan halaman *login* bagi peserta. Halaman ini berisi *form* untuk memasukkan email dan *password* peserta. *File* ini berisi kode HTML untuk tampilan halaman *login* peserta.

e. *File dashboard/daftar\_diterima.php*

*View* ini digunakan untuk menampilkan daftar peserta yang diterima pada halaman *dashboard*. Halaman ini berisi tabel dengan informasi peserta yang telah diterima. *File* ini berisi kode HTML untuk tampilan daftar peserta diterima.

f. *File dashboard/daftar\_ditolak.php*

*View* ini digunakan untuk menampilkan daftar peserta yang ditolak pada halaman *dashboard*. Halaman ini berisi tabel dengan informasi peserta yang telah ditolak. *File* ini berisi kode HTML untuk tampilan daftar peserta ditolak.

g. *File dashboard/daftar\_ppdb.php*

*View* ini digunakan untuk menampilkan daftar peserta PPDB pada halaman *dashboard*. Halaman ini berisi tabel dengan informasi peserta yang mendaftar. *File* ini berisi kode HTML untuk tampilan daftar peserta PPDB.

h. *File dashboard/dashboard.php*

*View* ini digunakan untuk menampilkan halaman utama *dashboard*. Halaman ini berisi ringkasan informasi, seperti jumlah peserta, berita terbaru, dan status PPDB. *File* ini berisi kode HTML untuk tampilan halaman utama *dashboard*.

i. *File dashboard/template/footer.php*

*View* ini merupakan bagian dari *template* untuk halaman *dashboard*. *Template footer* ini berisi elemen-elemen *footer*, seperti link ke skrip *JavaScript* dan penutup *tag* `<body>` dan `<html>`. *File* ini berisi kode HTML untuk tampilan *footer*.

j. *File dashboard/template/header.php*

View ini merupakan bagian dari *template* untuk halaman *dashboard*. *Template header* ini berisi elemen-elemen header, seperti *tag* <head>, judul halaman, dan link ke *stylesheet*. *File* ini berisi kode HTML untuk tampilan *header*.

k. *File dashboard/template/sidebar.php*

View ini merupakan bagian dari *template* untuk halaman *dashboard*. *Template sidebar* ini berisi menu navigasi yang terletak di sisi kiri halaman. *File* ini berisi kode HTML untuk tampilan *sidebar*.

Seluruh *view* tersebut digunakan untuk menampilkan tampilan halaman kepada pengguna sesuai dengan kebutuhan aplikasi. Masing-masing *view* berisi kode HTML yang membentuk tampilan halaman dengan elemen-elemen yang sesuai, seperti *form*, tabel, dan informasi.

#### 4.11 Uji Coba Fitur Halaman Utama

Pengujian fitur Halaman Utama atau *landing page* pada Aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung telah dilakukan dan menghasilkan hasil yang sukses. Fitur-fitur pada halaman utama berfungsi dengan baik dan sesuai dengan rencana yang telah ditetapkan.



Gambar 1. Halaman utama

Tombol ke halaman pendaftaran berhasil berfungsi dengan baik. Ketika tombol tersebut ditekan, pengguna diarahkan ke halaman pendaftaran yang sesuai. Proses pengujian menjamin bahwa tombol tersebut dapat mengarahkan pengguna ke tempat yang benar sehingga memudahkan calon siswa untuk mendaftar dalam program PPDB.

Tombol ke halaman *login* juga berfungsi sebagaimana mestinya. Pengujian memastikan bahwa tombol ini mengarahkan pengguna ke halaman *login* yang tepat, memungkinkan mereka untuk mengakses akun pribadi mereka. Fitur ini penting karena memisahkan antara calon siswa yang ingin mendaftar dan pengguna yang sudah memiliki akun dalam sistem.

Tampilan berita yang dibuat oleh panitia juga teruji dengan baik. Pengujian memastikan bahwa berita-berita yang dibuat oleh panitia dapat ditampilkan dengan benar pada halaman utama. Dengan demikian, pengguna dapat melihat informasi terkini mengenai PPDB MTs Tarbiyatul Muhtadiin Wilalung secara langsung dari halaman utama.

Hasil pengujian yang sukses menunjukkan bahwa fitur-fitur pada halaman utama aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung berfungsi dengan baik dan sesuai dengan rencana. Hal ini memberikan keyakinan bahwa pengguna akan memiliki pengalaman yang baik saat menggunakan aplikasi ini. Pengujian yang dilakukan memastikan bahwa semua fungsi pada halaman utama berjalan sesuai dengan harapan, sehingga memastikan kualitas dan kesesuaian aplikasi dengan kebutuhan dan tujuan pengguna.

#### 4.12 Uji Coba Fitur Registrasi Calon Siswa

Pengujian fitur Halaman Registrasi Calon Siswa pada Aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung telah dilakukan dan hasilnya sangat sukses. Halaman tersebut dapat ditampilkan dengan baik dan semua fiturnya berfungsi sesuai dengan rencana yang telah ditetapkan.



Gambar 2. Halaman registrasi calon siswa

Fitur utama pada halaman registrasi calon siswa adalah formulir pendaftaran. Pengujian memastikan bahwa formulir tersebut memiliki semua elemen yang diperlukan, seperti *textfield* untuk nama depan, nama belakang, NISN, nomor *handphone*, alamat, *email*, *password*, *date picker* untuk tanggal lahir, dan *dropdown* untuk memilih jenis kelamin. Semua elemen formulir ini berfungsi dengan baik dan memungkinkan calon siswa untuk mengisi informasi pribadi mereka dengan mudah.

Instruksi pada halaman registrasi juga telah dinilai komprehensif dan mudah dipahami oleh para pendaftar. Pengujian menunjukkan bahwa instruksi yang disediakan dengan jelas menjelaskan langkah-langkah yang harus diikuti dalam proses pendaftaran. Hal ini penting untuk memastikan bahwa calon siswa dapat mengisi formulir dengan benar dan memahami apa yang diharapkan dari mereka.

Selama pengujian, semua fitur pada halaman registrasi calon siswa bekerja dengan baik dan sesuai dengan rencana. Hal ini memberikan keyakinan bahwa proses pendaftaran dalam aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung dapat berjalan lancar dan efisien. Pengujian yang sukses ini memastikan bahwa calon siswa dapat mengakses dan menggunakan halaman registrasi dengan mudah, serta mengisi semua informasi yang diperlukan dengan benar.

#### 4.13 Uji Coba Fitur Pasca Registrasi

Selain fitur-fitur yang telah disebutkan sebelumnya, setelah calon siswa menekan tombol “Daftar” pada halaman registrasi, pengujian juga berhasil menunjukkan bahwa halaman pemberitahuan “Sukses Mendaftar” muncul dengan baik. Halaman ini memberikan konfirmasi kepada calon siswa bahwa pendaftaran mereka berhasil dilakukan.

Pengujian membuktikan bahwa halaman pemberitahuan “Sukses Mendaftar” menyediakan tombol untuk berpindah ke halaman *login* aplikasi. Fitur ini memungkinkan calon siswa untuk segera melanjutkan ke proses *login* setelah selesai mendaftar. Tombol ini berfungsi dengan baik dan mengarahkan calon siswa ke halaman *login* tanpa masalah.

Fungsi cetak PDF Bukti Pendaftaran Sementara berfungsi dengan baik. Setelah calon siswa mengunduh bukti pendaftaran sementara, mereka dapat membukanya dalam format PDF. Tombol “Download” berfungsi sesuai harapan, dan setelah ditekan, browser secara otomatis menampilkan menu *print* bawaan.



Gambar 3. Menu *print* Bukti Pendaftaran Sementara

Melalui menu *print* bawaan browser, pengguna memiliki opsi untuk menyimpan bukti pendaftaran sebagai dokumen PDF di perangkat mereka. Mereka juga dapat langsung mencetak bukti pendaftaran menggunakan printer yang terhubung ke perangkat mereka. Opsi-opsi ini memberikan fleksibilitas kepada pengguna dalam memilih cara yang paling sesuai untuk menyimpan dan menggunakan bukti pendaftaran sementara.

#### 4.14 Uji Coba Fitur *Login* sebagai Calon Siswa

Pengujian fitur *login* sebagai calon siswa pada aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung berjalan dengan baik. Data calon siswa yang sebelumnya didaftarkan melalui halaman registrasi dapat digunakan untuk *login* ke dalam sistem dengan sukses.

Pada halaman *login*, terdapat form yang meminta calon siswa untuk memasukkan email dan *password* yang telah mereka daftarkan sebelumnya. Pengujian dilakukan dengan menggunakan data calon siswa yang telah terdaftar sebelumnya. Setelah mengisi formulir *login* dengan benar, calon siswa

dapat menekan tombol “*Login*” untuk memverifikasi kredensial mereka.

Hasil pengujian menunjukkan bahwa fitur *login* berfungsi dengan baik. Sistem dapat memvalidasi informasi *login* yang dimasukkan oleh calon siswa dengan data yang tersimpan dalam database. Jika informasi yang dimasukkan cocok dengan data yang terdaftar, calon siswa akan diarahkan ke halaman utama *dashboard* sebagai pengguna yang terotentikasi.



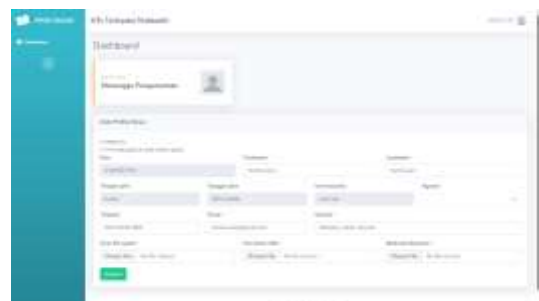
Gambar 4. Halaman *login*

Pengujian juga memverifikasi bahwa ketika calon siswa gagal memasukkan kredensial yang benar, sistem memberikan pemberitahuan yang sesuai, seperti pesan kesalahan yang menunjukkan bahwa kombinasi email dan *password* tidak valid. Link untuk kembali ke halaman utama juga sudah berfungsi sebagaimana mestinya.

#### 4.15 Uji Coba Fitur Halaman Pendaftaran Calon Siswa

Proses pengujian fitur halaman pendaftaran calon siswa pada aplikasi PPDB MTs Tarbiyatul Muhtadiin Wilalung berjalan dengan baik. Data calon siswa yang sebelumnya didaftarkan berhasil disimpan dalam database dan dapat ditampilkan kembali dengan benar.

Pada halaman akun calon siswa, terdapat informasi pribadi calon siswa seperti nama, NISN, nomor *handphone*, alamat, dan email. Pengujian dilakukan dengan memverifikasi bahwa data yang ditampilkan sesuai dengan yang telah *diinput* saat pendaftaran. Data calon siswa yang tersimpan dalam database dapat ditemukan dan ditampilkan dengan akurat.



Gambar 5. Halaman akun calon siswa

Pengujian juga mencakup fitur *upload scan* ijazah, pas foto 4x6, dan bukti pembayaran. Calon siswa dapat mengunggah *file-file* tersebut sesuai dengan persyaratan pendaftaran. Proses pengujian melibatkan pengunggahan *file-file* tersebut dengan tipe file yang diizinkan (misalnya PDF atau gambar)

dan memverifikasi bahwa file berhasil diunggah dan disimpan dalam server.

Selain fitur-fitur yang telah disebutkan sebelumnya, pengujian juga mencakup fitur konfirmasi sebelum menyimpan data pada halaman akun calon siswa. Fitur ini dirancang untuk memberikan pengguna kesempatan untuk memeriksa kembali data yang telah diinput sebelum benar-benar menyimpannya.

Setelah pengguna mengisi atau mengubah informasi pada halaman akun, misalnya mengganti nomor telepon atau alamat, dan menekan tombol "Simpan", muncul sebuah modal yang berisi pesan konfirmasi. Pesan konfirmasi tersebut mengingatkan pengguna bahwa data yang diinput akan disimpan dan sulit untuk diubah setelahnya. Pengguna diberikan pilihan untuk memastikan bahwa data yang mereka masukkan sudah benar sebelum melanjutkan proses penyimpanan.

Fitur konfirmasi sebelum menyimpan data ini memiliki beberapa keunggulan yang signifikan. Fitur ini meminimalisir kesalahan input data karena pengguna memiliki kesempatan untuk memeriksa kembali data mereka sebelum disimpan. Ini membantu mencegah kesalahan yang mungkin terjadi karena kesalahan pengetikan atau kelalaian.

Fitur ini juga memberikan transparansi kepada pengguna tentang konsekuensi menyimpan data. Dengan menyampaikan bahwa data yang disimpan sulit untuk diubah, pengguna lebih mawas diri dan berhati-hati dalam mengisi informasi mereka. Hal ini memastikan bahwa data yang tersimpan adalah data yang akurat dan valid.

Fitur konfirmasi ini juga membantu meningkatkan kepercayaan pengguna terhadap aplikasi. Pengguna merasa bahwa aplikasi memberikan perlindungan dan perhatian terhadap data mereka. Ini dapat membangun kepercayaan bahwa data pribadi mereka akan dikelola dengan baik dan sesuai dengan privasi dan keamanan yang diperlukan.



Gambar 6. Konfirmasi sebelum menyimpan data

Hasil pengujian menunjukkan bahwa fitur *upload* berfungsi sebagaimana mestinya. Data yang diunggah oleh calon siswa, seperti *scan* ijazah, pas foto 4x6, dan bukti pembayaran, dapat disimpan dengan benar dalam database atau direktori yang ditentukan. Pengujian juga memverifikasi bahwa setelah mengunggah file, calon siswa dapat melihat

dan mencetak bukti pendaftaran yang berisi informasi pribadi mereka dan file-file yang telah diunggah.

Setelah calon siswa berhasil menyimpan data pada halaman akun, pengujian telah menunjukkan bahwa data yang diunggah sebelumnya dapat ditampilkan dengan benar. Misalnya, calon siswa dapat mengunggah *scan* ijazah, pas foto 4x6, dan bukti pembayaran. Setelah data tersebut tersimpan, calon siswa dapat melihatnya kembali saat mengakses halaman akun mereka.

Pada halaman akun calon siswa, terdapat bagian yang menampilkan data yang telah diunggah sebelumnya, seperti *scan* ijazah, pas foto, dan bukti pembayaran. Data tersebut ditampilkan dengan jelas dan tersusun rapi, sehingga calon siswa dapat dengan mudah melihatnya. Ini memungkinkan calon siswa untuk memverifikasi bahwa data yang mereka unggah telah tersimpan dengan benar.

Pengujian juga telah membuktikan bahwa calon siswa dapat mencetak bukti pendaftaran setelah menyimpan data. Setelah melihat data yang telah diunggah dan memastikan keakuratannya, calon siswa dapat menggunakan fitur cetak yang disediakan pada halaman akun mereka. Tombol "Cetak Bukti Pendaftaran" atau fungsi serupa akan memungkinkan calon siswa untuk menghasilkan dokumen bukti pendaftaran yang siap dicetak.



Gambar 7. Menu *print* Bukti Pendaftaran

Fitur ini sangat penting dalam memastikan bahwa calon siswa memiliki salinan fisik bukti pendaftaran mereka. Dengan memiliki bukti pendaftaran yang tercetak, calon siswa dapat dengan mudah menyimpannya sebagai referensi dan melibatkan mereka dalam proses seleksi dan administrasi yang mungkin terjadi selanjutnya.

#### 4.16 Uji Coba Fitur *Login* sebagai Panitia PPDB

Pada pengujian ulang halaman *login* menggunakan akun panitia PPDB, hasilnya menunjukkan bahwa fitur tersebut berfungsi dengan baik. Proses pengujian melibatkan penggunaan akun panitia yang telah didaftarkan sebelumnya dalam sistem.

Saat mengakses halaman *login*, panitia PPDB dapat memasukkan informasi *login* yang benar, yaitu *username* dan *password* yang telah ditentukan saat pendaftaran akun. Pengujian telah membuktikan

bahwa sistem dapat mengenali dan memvalidasi akun panitia dengan tepat.

Setelah mengisi informasi *login*, pengujian menunjukkan bahwa proses autentikasi berjalan dengan lancar. Halaman akan mengautentikasi informasi yang dimasukkan oleh panitia PPDB dan memberikan akses ke halaman berikutnya jika akun valid. Namun, jika informasi *login* tidak sesuai, sistem memberikan pesan kesalahan yang sesuai, memberi tahu panitia PPDB tentang kesalahan *login* yang terjadi.

Hasil pengujian juga menunjukkan bahwa setelah berhasil *login*, panitia PPDB dapat mengakses halaman yang ditujukan khusus untuk mereka. Hal ini memastikan bahwa akses panitia dibatasi hanya untuk fitur dan informasi yang relevan dengan peran dan tanggung jawab mereka dalam proses PPDB.



Gambar 8. Halaman *login* sebagai panitia PPDB

Pengujian ini menunjukkan bahwa fitur halaman *login* untuk akun panitia PPDB berfungsi dengan baik. Dengan demikian, panitia PPDB dapat mengakses sistem dengan aman dan dapat melaksanakan tugas-tugas yang diberikan kepada mereka. Keberhasilan pengujian ini menunjukkan bahwa pengelolaan akses pengguna dalam aplikasi PPDB MTs Tarbiyatul Mubtadiin Wilalung telah diimplementasikan dengan baik.

#### 4.17 Uji Coba Fitur Penerimaan Calon Siswa

Setelah berhasil *login*, panitia PPDB akan diberikan tampilan *dashboard* yang menyajikan berbagai informasi terkait proses penerimaan peserta PPDB. *Dashboard* ini memberikan kemudahan bagi panitia dalam mengelola dan memantau proses pendaftaran.



Gambar 9. Halaman *dashboard* panitia PPDB

Pada *sidebar dashboard* terdapat menu navigasi yang mempermudah panitia PPDB untuk berpindah antara halaman-halaman terkait. Menu ini mencakup opsi untuk mengakses halaman

*dashboard* utama, melihat daftar peserta PPDB secara keseluruhan, melihat daftar peserta yang diterima, dan melihat daftar peserta yang ditolak. Dengan adanya menu navigasi ini, panitia dapat dengan mudah mengakses dan memantau data peserta PPDB sesuai dengan kebutuhan mereka.

Halaman *dashboard* pada aplikasi PPDB MTs Tarbiyatul Mubtadiin Wilalung menyajikan informasi dan fitur penting bagi panitia PPDB. Pada halaman ini, terdapat beberapa komponen yang meliputi tombol untuk menutup PPDB, tiga *card* yang menampilkan total peserta PPDB, total peserta diterima, dan total peserta ditolak, serta menu berita yang berisi tombol tambah postingan dan tabel berita dengan tombol edit dan hapus pada setiap baris datanya.

Tombol untuk menutup PPDB memberikan fitur kepada panitia PPDB untuk mengakhiri proses penerimaan pendaftaran. Dengan mengeklik tombol ini, panitia dapat menutup akses pendaftaran sehingga tidak ada lagi calon siswa yang dapat mendaftar melalui aplikasi.

Tiga *card* yang disajikan pada halaman *dashboard* memberikan informasi singkat mengenai jumlah peserta PPDB secara keseluruhan, jumlah peserta yang diterima, dan jumlah peserta yang ditolak. *Card* ini memberikan gambaran cepat tentang perkembangan pendaftaran dan memudahkan panitia dalam melihat angka-angka penting terkait proses PPDB.

Menu berita memberikan akses kepada panitia PPDB untuk mengelola informasi yang ditampilkan di halaman utama aplikasi. Terdapat tombol tambah postingan yang memungkinkan panitia untuk membuat dan mempublikasikan berita terbaru. Tabel berita menampilkan daftar berita yang telah diposting sebelumnya, dan setiap baris datanya dilengkapi dengan tombol edit dan hapus. Fitur ini memungkinkan panitia untuk mengedit atau menghapus berita yang sudah ada, sehingga mereka dapat memperbarui dan mengatur konten informasi yang ingin disampaikan kepada pengguna aplikasi.



Gambar 10. *Form* detail berita untuk pembuatan dan edit berita

Proses pengujian halaman *dashboard* telah dilakukan dan hasilnya berhasil. Setiap komponen pada halaman *dashboard* berfungsi dengan baik, termasuk tombol untuk menutup PPDB, tiga *card* yang menampilkan data peserta, menu berita dengan

tombol tambah postingan, dan tabel berita dengan tombol edit dan hapus. Penggunaan fitur-fitur tersebut mempermudah panitia dalam mengelola dan memantau proses PPDB serta mengatur informasi yang disampaikan melalui halaman utama aplikasi.

Halaman peserta PPDB terdiri dari tabel daftar peserta PPDB yang menyajikan informasi penting seperti NISN (Nomor Induk Siswa Nasional), nama, jenis kelamin, nomor telepon, tanggal registrasi, bukti pembayaran, dan status pendaftaran. Tabel ini memberikan pandangan yang jelas dan terstruktur mengenai calon siswa yang mendaftar, memudahkan panitia dalam mengelola data dan mengambil keputusan terkait penerimaan peserta PPDB.

Pengujian fitur ini menunjukkan bahwa halaman peserta PPDB panitia PPDB berfungsi dengan baik. Informasi yang relevan dan penting tersedia dalam tampilan tabel yang jelas dan terstruktur. Panitia dapat melakukan navigasi antar halaman dengan mudah menggunakan menu *sidebar* yang disediakan. Fitur ini membantu panitia dalam mengelola dan memantau proses penerimaan peserta PPDB dengan lebih efisien dan efektif.



Gambar 11. Halaman *dashboard* panitia PPDB

Selain fitur yang telah disebutkan sebelumnya, pada halaman peserta PPDB, terdapat fungsi tambahan yang memungkinkan panitia untuk melihat data lengkap calon siswa dengan lebih detail. Ketika panitia melakukan klik pada NISN pada tabel daftar peserta PPDB, akan muncul sebuah modal yang berisi informasi lengkap mengenai calon siswa yang terkait.

Modal ini menampilkan semua data yang telah diinputkan oleh calon siswa pada halaman registrasi. Informasi yang ditampilkan meliputi nama lengkap, NISN, jenis kelamin, nomor telepon, alamat, email, tanggal lahir, serta bukti pembayaran yang telah diunggah. Dengan adanya modal ini, panitia PPDB dapat memeriksa data calon siswa secara rinci dan memverifikasi keaslian informasi yang diberikan.

Fitur ini sangat berguna bagi panitia PPDB dalam melakukan pengecekan dan validasi data calon siswa. Mereka dapat memastikan bahwa data yang dimasukkan oleh calon siswa sesuai dengan persyaratan yang ditetapkan. Selain itu, fitur ini juga membantu panitia dalam mengambil keputusan yang tepat terkait penerimaan peserta PPDB berdasarkan informasi yang tersedia.

Dalam pengujian, fungsi ini telah berhasil diuji dan berjalan dengan baik. Ketika panitia melakukan klik pada NISN, modal dengan data lengkap calon siswa tampil dengan benar. Hal ini memastikan bahwa panitia dapat mengakses dan memeriksa informasi calon siswa secara detail, memudahkan mereka dalam proses pengelolaan dan evaluasi pendaftaran.



Gambar 12. Detail data calon siswa

Selain fitur-fitur sebelumnya, pada tabel daftar peserta PPDB di halaman peserta PPDB, terdapat kemampuan untuk melihat bukti pembayaran yang diunggah oleh calon siswa dengan lebih detail. Ketika panitia melakukan klik pada gambar bukti pembayaran pada tabel, gambar tersebut akan terbuka dalam tab baru dengan resolusi penuh.

Fitur ini memungkinkan panitia untuk memeriksa bukti pembayaran calon siswa secara jelas dan mendetail. Dengan membuka gambar bukti pembayaran dalam resolusi penuh, panitia dapat melakukan pengecekan lebih lanjut terhadap bukti tersebut. Mereka dapat memverifikasi keaslian dan kejelasan informasi pada bukti pembayaran yang diunggah, memastikan bahwa calon siswa telah memenuhi persyaratan pembayaran yang ditetapkan.

Pengujian fitur ini telah dilakukan dan hasilnya memuaskan. Ketika panitia PPDB melakukan klik pada gambar bukti pembayaran, gambar tersebut berhasil terbuka dalam tab baru dengan resolusi penuh. Hal ini memudahkan panitia dalam mengevaluasi dan memeriksa bukti pembayaran calon siswa dengan lebih teliti.

Dengan adanya fitur ini, panitia PPDB dapat mengelola proses pendaftaran dengan lebih efisien dan akurat. Mereka dapat melihat data peserta PPDB secara rinci, termasuk bukti pembayaran yang telah diunggah, sehingga mempermudah pengambilan keputusan terkait status pendaftaran peserta.



Gambar 13. Tampilan bukti pembayaran

Halaman peserta PPDB pada aplikasi PPDB MTs Tarbiyatul Mubtadiin Wilalung dilengkapi dengan fitur tombol terima atau tolak calon siswa. Pada setiap baris data calon siswa yang terdaftar, terdapat tombol “Terima” dan “Tolak” yang memungkinkan panitia untuk memproses status pendaftaran calon siswa tersebut. Ketika tombol terima atau tolak ditekan, muncul sebuah modal yang mengkonfirmasi aksi yang akan dilakukan.

Modal konfirmasi ini memberikan kesempatan kepada panitia untuk memastikan keputusan yang diambil. Dalam modal tersebut, panitia diberikan informasi terkait nama calon siswa dan diminta untuk memilih apakah akan menerima atau menolak pendaftaran calon siswa tersebut. Dengan adanya modal konfirmasi ini, panitia dapat memperhatikan kembali data dan informasi calon siswa sebelum mengambil keputusan yang final.



Gambar 14. Modal konfirmasi terima peserta PPDB

Setelah memilih tindakan yang diinginkan, panitia dapat menekan tombol “Ya” atau “Batal” pada modal konfirmasi. Aksi ini akan mengubah status pendaftaran calon siswa menjadi diterima atau ditolak sesuai dengan keputusan yang diambil. Dengan adanya fitur ini, panitia dapat dengan mudah mengelola proses seleksi calon siswa dan memberikan respons yang tepat terhadap setiap pendaftaran.

Proses pengujian fitur tombol terima atau tolak calon siswa pada halaman peserta PPDB telah dilakukan dan berjalan dengan baik. Setiap tombol merespons dengan benar terhadap aksi yang dilakukan, dan modal konfirmasi muncul untuk memastikan keputusan yang diambil oleh panitia. Fitur ini memberikan fleksibilitas dan kendali kepada panitia dalam mengelola proses seleksi calon siswa, sehingga memastikan keputusan yang diambil

berdasarkan pertimbangan yang matang dan data yang akurat.

Halaman “Peserta Diterima” dan “Peserta Ditolak” pada aplikasi PPDB MTs Tarbiyatul Mubtadiin Wilalung adalah halaman yang menampilkan tabel peserta berdasarkan keputusan yang telah ditetapkan oleh panitia. Pada halaman “Peserta Diterima”, ditampilkan daftar calon siswa yang telah diterima dalam proses seleksi PPDB. Sedangkan pada halaman “Peserta Ditolak”, ditampilkan daftar calon siswa yang tidak lolos seleksi dan pendaftarannya ditolak.

Tabel peserta pada kedua halaman ini berisi informasi penting seperti NISN, nama, jenis kelamin, nomor telepon, dan tanggal registrasi calon siswa. Selain itu, terdapat juga kolom yang menunjukkan bukti pembayaran dan status pendaftaran calon siswa. Tampilan tabel ini memberikan gambaran yang jelas tentang calon siswa yang telah diterima atau ditolak oleh panitia.

Pada halaman “Peserta Ditolak”, terdapat juga tombol “Hapus” pada setiap baris data calon siswa yang memungkinkan panitia untuk menghapus data pendaftaran calon siswa yang ditolak. Tombol ini memunculkan konfirmasi sebelum penghapusan dilakukan untuk memastikan keputusan yang diambil oleh panitia.

Proses pengujian fitur halaman “Peserta Diterima” dan “Peserta Ditolak” telah dilakukan dan berjalan dengan baik. Tabel peserta berhasil menampilkan data calon siswa sesuai dengan keputusan panitia, baik yang diterima maupun ditolak. Tombol “Hapus” pada halaman “Peserta Ditolak” berfungsi dengan benar dan memunculkan konfirmasi sebelum penghapusan data dilakukan. Fitur ini memberikan kemudahan bagi panitia dalam mengelola data calon siswa yang telah diterima maupun ditolak dalam proses seleksi PPDB.

## 5. KESIMPULAN

Aplikasi ini berhasil memfasilitasi proses penerimaan peserta didik baru dengan efisien dan efektif. Fitur-fitur yang terdapat dalam aplikasi, seperti halaman utama, halaman registrasi calon siswa, halaman akun calon siswa, halaman dashboard panitia, dan halaman peserta diterima dan ditolak, berjalan dengan baik dan sesuai dengan rencana. Selain itu, fitur-fitur tambahan seperti konfirmasi sebelum menyimpan data, tampilan data lengkap calon siswa dalam modal, dan tampilan gambar bukti pembayaran dalam tab baru juga berfungsi dengan baik. Dengan demikian, aplikasi PPDB MTs Tarbiyatul Mubtadiin Wilalung dapat menjadi solusi yang efisien dan efektif dalam proses penerimaan peserta didik baru di sekolah tersebut.

## DAFTAR PUSTAKA

Agustina, M. dan Mandela, D. (2022) “Aplikasi Penerimaan Peserta Didik Baru (PPDB) Berbasis

Website Pada SMAN 1 Gunung Megang Menggunakan Konsep Model-View-Controller (MVC),” *Jurnal Ilmiah Matrik*, 24(3), hal. 292–301.

ch.ebscohost.com/login.aspx?direct=true&db=f5h&AN=128438643&site=eds-live&scope=site.

Hidayat, T., Muttaqin, M. dan Djamaludin, D. (2020) “Sistem Informasi Penerimaan Peserta Didik Baru Online Berbasis Website di Yayasan Pendidikan Arya Jaya Sentika,” *Komputika : Jurnal Sistem Komputer*, 9(1), hal. 7–14. Tersedia pada: <https://doi.org/10.34010/komputika.v9i1.2750>.

Nugroho, B. (2020) “Sistem Pendukung Keputusan Pemilihan Laptop dengan Metode Analytical Hierarchy Process menggunakan Platform Android,” *Jurnal Wira Pratika*, 7(1), hal. 15–25.

Ozen, F. dan Simsek, M.A. (2019) “Realization of A Building Automation System Using PLC and SCADA,” *International Journal of Engineering and Innovative Research*, 1(1), hal. 28–34.

Puspita, K., Alkhalifi, Y. dan Basri, H. (2021) “Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru Berbasis Website Dengan Metode Spiral,” *Paradigma - Jurnal Komputer dan Informatika*, 23(1), hal. 35–42. Tersedia pada: <https://doi.org/10.31294/p.v23i1.10434>.

Rachmat, N. dan Arman, M. (2020) “Rancang Bangun Aplikasi Penerimaan Peserta Didik Baru Sekolah Swasta Berbasis Web,” *Jurnal Teknologi Informasi Mura*, 12(02), hal. 99–110. Tersedia pada: <https://doi.org/10.32767/jti.v12i02.1065>.

Salamah, S. (2020) “Peran Guru Pai Dalam Pembentukan Karakter Siswa Di Era Revolusi Industri 4.0,” *SCAFFOLDING: Jurnal Pendidikan Islam dan Multikulturalisme*, 2(1), hal. 26–36. Tersedia pada: <https://doi.org/10.37680/scaffolding.v2i1.281>.

Severance, C.R. (2018) “Python for Everybody (Coursera course),” hal. 245. Tersedia pada: <https://books.trinket.io/pfe/index.html>.

Sholeh, M., Rachmawati, R.Y. dan Cahyo, E.N. (2022) “Penerapan Regresi Linear Ganda Untuk Memprediksi Hasil Nilai Kuesioner Mahasiswa Dengan Menggunakan Python,” *Jurnal Dinamika Informatika*, 11(1), hal. 13–24. Tersedia pada: <https://jdi.upy.ac.id/index.php/jdi/article/view/124>.

Sonata, F.- (2019) “Pemanfaatan UML (Unified Modeling Language) Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer-To-Customer,” *Jurnal Komunika : Jurnal Komunikasi, Media dan Informatika*, 8(1), hal. 22. Tersedia pada: <https://doi.org/10.31504/komunika.v8i1.1832>.

Tserpes, K. (2019) “stream-MSA: A microservices’ methodology for the creation of short, fast-paced, stream processing pipelines,” *ICT Express*, 5(2), hal. 146–149. Tersedia pada: <https://doi.org/10.1016/j.icte.2019.04.001>.

Wiedeman, R. (2018) *Gray Hat., New York*. Tersedia pada: <https://oralroberts.idm.oclc.org/login?url=https://sear>