

Sistem Penerimaan Peserta Didik Baru Terintegrasi Virtual Account Berbasis Web Pada SMK Sasmita Jaya

Rizky Maulana¹, Fahrul Maulana²

^{1,2}Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia
Email: ¹rizki.mwp404@gmail.com, ²fahrul@unpam.ac.id

Abstrak— Sistem Penerimaan Peserta Didik Baru (PPDB) berbasis web dikembangkan untuk mengatasi tantangan administrasi di SMK Sasmita Jaya, seperti lamanya proses pendaftaran manual, tingginya tingkat kesalahan dalam pengelolaan data, dan keterbatasan aksesibilitas bagi calon siswa. Sistem ini diimplementasikan menggunakan React TypeScript untuk frontend dan backend, yang menjamin keamanan tipe serta kemudahan pemeliharaan. Fitur utama yang disediakan meliputi pendaftaran online, verifikasi dokumen otomatis, dan integrasi pembayaran melalui virtual account, menggantikan metode pembayaran manual tradisional. Integrasi virtual account memungkinkan rekonsiliasi transaksi secara otomatis, sehingga beban kerja staf administrasi dapat berkurang secara signifikan. Sistem ini tidak hanya meningkatkan efisiensi tetapi juga memperbaiki akurasi proses penerimaan peserta didik. Metodologi pengembangan sistem menggunakan pendekatan Agile, yang memungkinkan perbaikan secara iteratif serta memastikan sistem dapat memenuhi kebutuhan pengguna dengan efektif. Pengujian menyeluruh dilakukan untuk memastikan keandalan dan keamanan sistem, terutama dalam menangani transaksi keuangan. Hasil evaluasi menunjukkan bahwa sistem ini mampu mengurangi waktu pemrosesan pendaftaran hingga 50%, menurunkan tingkat kesalahan administrasi sebesar 40%, dan meningkatkan kepuasan pengguna hingga 85%. Indikator-indikator ini menegaskan keberhasilan sistem dalam mengatasi tantangan administrasi sekolah serta menunjukkan manfaat digitalisasi di institusi pendidikan. Melalui integrasi teknologi modern, sistem PPDB ini mendukung proses penerimaan peserta didik yang lebih efisien dan mulus, sekaligus menawarkan pengalaman yang lebih mudah diakses dan ramah pengguna bagi calon siswa maupun staf administrasi. Sistem ini dapat menjadi model bagi institusi lain yang ingin menerapkan teknologi serupa untuk meningkatkan efisiensi operasional mereka.

Kata Kunci: PPDB, Typescript, Virtual Account, website, React

Abstract— The web-based New Student Admission (PPDB) system was developed to solve administrative challenges at SMK Sasmita Jaya, such as the time-consuming manual registration process, high error rates in data management, and limited accessibility for prospective students. The system was implemented using TypeScript for both the frontend and backend, providing strong type safety and ease of maintenance. Key features include online registration, automated document verification, and payment integration with virtual accounts, replacing traditional manual payment methods. The integration of virtual accounts allows for automatic transaction reconciliation, significantly reducing the workload of administrative staff. This system not only ensures efficiency but also improves the accuracy of the admission process. The development methodology followed an Agile approach, enabling iterative improvements and ensuring the system met user requirements effectively. Comprehensive testing was conducted to ensure system reliability and security, particularly in handling financial transactions. Evaluation results indicate that the system reduces registration processing time by 50%, decreases administrative errors by 40%, and increases user satisfaction by 85%. These metrics highlight the system's success in addressing the school's administrative challenges and demonstrate the benefits of adopting digital technology in educational institutions. By integrating modern technology, this PPDB system supports a seamless and efficient admission process, offering a more accessible and user-friendly experience for both prospective students and administrative staff. It serves as a model for other institutions aiming to implement similar systems to improve their operational processes.

Keywords: PPDB, Typescript, Virtual Account, website, React

1. PENDAHULUAN

Proses Penerimaan Peserta Didik Baru (PPDB) di SMK Sasmita Jaya sebelumnya menggunakan sistem manual yang memiliki beberapa permasalahan mendasar. Pertama, proses pendaftaran yang dilakukan secara langsung di sekolah memakan waktu lama dan menyebabkan antrean panjang, terutama pada masa puncak pendaftaran. Kedua, verifikasi dokumen calon siswa dilakukan secara manual, sehingga sering terjadi kesalahan administrasi, seperti ketidaksesuaian data atau kehilangan dokumen. Ketiga, pembayaran biaya pendaftaran yang dilakukan secara tunai atau transfer bank tanpa mekanisme otomatis menyebabkan kesulitan dalam rekonsiliasi transaksi, menambah beban kerja staf administrasi. Selain itu, aksesibilitas sistem manual sangat terbatas bagi calon siswa yang tinggal di luar wilayah sekolah, sehingga mereka harus datang langsung ke sekolah untuk mendaftar, yang tentunya kurang efisien.

Penelitian tentang digitalisasi proses Penerimaan Peserta Didik Baru (PPDB) telah banyak dilakukan dalam beberapa tahun terakhir, seiring dengan semakin berkembangnya teknologi informasi yang dapat

meningkatkan efisiensi dan efektivitas administrasi pendidikan. Salah satu penelitian oleh Susanto [1] menunjukkan bahwa transformasi digital dalam sistem penerimaan siswa mampu mempercepat proses administrasi hingga 40% dibandingkan metode konvensional. Penelitian ini menyoroti pentingnya penerapan sistem berbasis web untuk memperluas aksesibilitas pendaftaran, terutama bagi siswa yang tinggal jauh dari lokasi sekolah, sehingga mereka dapat melakukan pendaftaran tanpa harus hadir langsung. Hasil penelitian ini relevan dengan pengembangan sistem PPDB di SMK Sasmita Jaya, di mana sistem berbasis web diadopsi untuk memberikan kemudahan akses bagi calon siswa dari berbagai lokasi.

Selain itu, penelitian oleh Iskandar [2] menegaskan bahwa penerapan verifikasi dokumen secara digital dalam sistem PPDB dapat mengurangi kesalahan manusia dan mempercepat proses validasi data siswa. Penelitian ini menunjukkan bahwa sekolah yang mengintegrasikan teknologi verifikasi otomatis dalam proses PPDB mengalami penurunan kesalahan verifikasi hingga 30%. Hal ini sejalan dengan penerapan sistem PPDB di SMK Sasmita Jaya yang juga mengintegrasikan verifikasi dokumen secara digital, yang diharapkan dapat meningkatkan akurasi dan mengurangi waktu yang dibutuhkan dalam proses pendaftaran.

Penelitian terkait integrasi sistem pembayaran *virtual account* dalam administrasi sekolah juga telah dibahas oleh Mulyana [3]. Penelitiannya menunjukkan bahwa sekolah yang menerapkan sistem *virtual account* untuk pembayaran biaya administrasi mampu mengurangi beban kerja administratif hingga 25% dan mempercepat proses rekonsiliasi pembayaran. Selain itu, penelitian ini mencatat bahwa sistem pembayaran *virtual account* meningkatkan kepuasan pengguna karena kemudahan dan kecepatan proses pembayaran. Penelitian ini mendukung pengembangan sistem PPDB di SMK Sasmita Jaya yang menggunakan *virtual account* sebagai metode pembayaran, sehingga diharapkan dapat meningkatkan efisiensi administrasi dan kepuasan pengguna.

Haryanto et al. [4] membahas penggunaan *TypeScript* dalam pengembangan aplikasi skala besar, khususnya di sektor pendidikan. Penelitian ini menemukan bahwa *TypeScript* dapat meningkatkan keamanan tipe data dan mengurangi kesalahan dalam pengembangan kode hingga 20%, sehingga memudahkan pemeliharaan sistem jangka panjang. Penggunaan *TypeScript* dalam sistem PPDB di SMK Sasmita Jaya mendukung hasil penelitian ini, karena dapat membantu memastikan stabilitas sistem dan mengurangi risiko kesalahan yang mungkin muncul selama proses pengembangan dan pemeliharaan.

Studi oleh Putra [5] juga menyoroti tantangan yang dihadapi dalam penerapan sistem pendaftaran online, termasuk masalah kompatibilitas, keamanan data, dan ketersediaan jaringan. Penelitian ini menyarankan agar sistem PPDB berbasis digital mempertimbangkan aspek-aspek tersebut untuk meningkatkan keberhasilan implementasi. Di SMK Sasmita Jaya, sistem PPDB dikembangkan dengan mempertimbangkan keamanan data calon siswa, yang merupakan prioritas dalam sistem berbasis web, serta menggunakan protokol keamanan untuk menjaga integritas data selama proses pendaftaran dan pembayaran.

Dari tinjauan penelitian terdahulu ini, terlihat bahwa sistem PPDB terintegrasi berbasis web dengan *virtual account* [6] dan pengembangan menggunakan *TypeScript* di SMK Sasmita Jaya sejalan dengan tren dan hasil positif yang ditemukan dalam penelitian sebelumnya. Digitalisasi proses PPDB tidak hanya meningkatkan efisiensi dan akurasi, tetapi juga memberikan kemudahan akses bagi calon siswa. Sistem ini berpotensi menjadi model untuk institusi pendidikan lainnya dalam mengelola penerimaan siswa baru dengan lebih modern dan efektif.

2. METODE PENELITIAN

2.1 Perencanaan

Tahap perencanaan adalah langkah awal dalam penelitian ini, yang bertujuan untuk membangun dasar pengembangan sistem PPDB berbasis digital. Proses ini melibatkan langkah-langkah berikut:

2.1.1 Identifikasi Pemangku Kepentingan

a. Tujuan

Mengidentifikasi pihak-pihak yang akan terlibat dalam sistem, baik secara langsung maupun tidak langsung.

b. Langkah

Melibatkan administrator sekolah, guru, calon siswa, dan wali siswa dalam diskusi untuk memahami kebutuhan dan kendala dari sistem PPDB manual.

c. Kriteria Keberhasilan

Tercapainya daftar pemangku kepentingan lengkap dengan peran masing-masing yang relevan terhadap pengembangan sistem.

2.1.2 Pengumpulan Kebutuhan

a. **Tujuan**

Mendapatkan pemahaman mendalam mengenai kebutuhan pengguna dan ekspektasi sistem.

b. **Langkah**

1. Wawancara mendalam dengan administrator untuk memahami alur kerja manual.
2. Survei kepada calon siswa untuk mengetahui preferensi penggunaan sistem.
3. Diskusi kelompok fokus untuk menggali ide dan masukan fitur sistem.

c. **Kriteria Keberhasilan**

Terdapat dokumen kebutuhan pengguna (*user requirement document*) yang disepakati oleh semua pemangku kepentingan.

2.1.3 Penyusunan Backlog Produk

a. **Tujuan**

Membuat daftar fitur sistem berdasarkan prioritas dan kebutuhan pengguna.

b. **Langkah**

1. Deskripsi fitur.
2. Estimasi waktu pengerjaan.
3. Prioritas (rendah, menengah, tinggi).

c. **Kriteria Keberhasilan**

Backlog produk tersusun secara lengkap dengan fitur yang dikelompokkan berdasarkan iterasi pengembangan.

2.2 Pengembangan Iteratif

Sistem PPDB dikembangkan menggunakan pendekatan *Agile*, yang memungkinkan fleksibilitas dalam merespons perubahan kebutuhan. Setiap iterasi meliputi langkah-langkah berikut:

2.2.1 Perencanaan Iterasi

a. **Tujuan**

Menentukan fitur atau perbaikan yang akan dikerjakan dalam iterasi tertentu.

b. **Langkah**

1. Mengadakan rapat perencanaan (*sprint planning meeting*).
2. Memilih item backlog berdasarkan prioritas dan kapasitas tim.

c. **Kriteria Keberhasilan**

Daftar tugas (*task list*) yang jelas untuk iterasi, termasuk pembagian tanggung jawab tim.

2.2.2 Desain Sistem

a. **Tujuan**

Merancang arsitektur sistem dan antarmuka pengguna.

b. **Langkah**

1. Desain arsitektur menggunakan pendekatan **MVC (Model-View-Controller)**.
2. Desain antarmuka pengguna dibuat dengan mempertimbangkan **UI/UX principles**.
3. Tools yang digunakan: **Figma** untuk desain UI dan diagram ERD (*Entity-Relationship Diagram*) untuk perancangan database.

c. **Kriteria Keberhasilan**

Tersedianya prototipe desain sistem yang siap diimplementasikan.

2.2.3 Pengkodean

a. **Tujuan**

Mengimplementasikan fitur sesuai dengan backlog dan desain.

b. **Langkah:**

1. Menggunakan **TypeScript** sebagai bahasa pemrograman utama.
2. Framework yang digunakan: **NestJS** untuk backend dan **ReactJS** untuk frontend.
3. Penerapan prinsip **Clean Code** dan pengujian unit (*unit testing*).

c. **Kriteria Keberhasilan**

Kode fitur selesai dan melewati tahap pengujian awal tanpa error signifikan.

2.3 Umpan Balik Pengguna

Setelah setiap iterasi, umpan balik dari pengguna (calon siswa dan administrator) sangat penting untuk proses pengembangan. Beberapa langkah yang diambil meliputi:

a. **Demonstrasi Fitur**

Tim mengadakan sesi demonstrasi untuk menunjukkan fitur yang telah dikembangkan kepada pengguna. Hal ini membantu pengguna memahami fungsionalitas sistem dan memberikan masukan secara langsung.

b. **Survei dan Wawancara**

Setelah demonstrasi, survei dan wawancara dilakukan untuk mengumpulkan pendapat dan saran dari pengguna mengenai pengalaman mereka menggunakan fitur baru.

c. **Perbaikan Berkelanjutan**

Umpan balik yang diterima digunakan untuk merencanakan perbaikan dan penambahan fitur di iterasi berikutnya, sehingga sistem dapat terus berkembang sesuai kebutuhan pengguna.

2.4 Implementasi dan Peluncuran

Setelah semua fitur selesai dikembangkan dan diuji, sistem diuji dalam lingkungan nyata di SMK Sasmita Jaya. Tahapan ini meliputi:

a. **Pelatihan Pengguna**

Pengguna akhir diberikan pelatihan untuk menggunakan sistem PPDB. Pelatihan ini bertujuan untuk meningkatkan pemahaman pengguna tentang cara menggunakan sistem dengan efektif.

b. **Uji Coba Sistem**

Sistem diuji coba secara menyeluruh dengan pengguna untuk memastikan bahwa semua fitur berfungsi dengan baik dalam konteks nyata. Selama fase ini, tim pengembang siap untuk menangani masalah atau bug yang mungkin muncul.

c. **Peluncuran Resmi**

Setelah semua pengujian selesai dan umpan balik terakhir diterima, sistem siap diluncurkan secara resmi. Tim memastikan bahwa semua dokumentasi dan panduan pengguna disediakan agar pengguna dapat dengan mudah mengakses dan menggunakan sistem.

2.5 Dokumentasi

Seluruh proses pengembangan, termasuk perbaikan dan revisi yang dilakukan, didokumentasikan dengan baik. Dokumentasi ini mencakup

a. Dokumentasi Teknis

Menyediakan informasi mendalam mengenai arsitektur dan fungsi sistem, termasuk deskripsi sistem, struktur kode, API atau endpoint yang digunakan, spesifikasi teknologi yang diterapkan, diagram alir, dan alur proses. Dokumentasi ini membantu pengembang lain dalam memahami dan mengembangkan sistem lebih lanjut.

b. Dokumentasi Instalasi

Berisi panduan lengkap untuk menginstal sistem di lingkungan baru, termasuk persyaratan perangkat keras dan perangkat lunak, langkah-langkah instalasi, konfigurasi, pengaturan database, serta cara memverifikasi instalasi berhasil. Panduan ini penting untuk memastikan bahwa sistem dapat diinstal dan dijalankan dengan benar.

c. Dokumentasi Hak Cipta

Menginformasikan tentang pemilik hak cipta, lisensi yang digunakan, aturan pemakaian ulang, dan batasan tanggung jawab. Dokumen ini menjaga hak hukum atas perangkat lunak dan memastikan bahwa pengguna dan pengembang memahami batasan pemakaian.

d. Dokumentasi Penggunaan untuk Pengguna

Berisi panduan yang lebih mendetail tentang fitur dan antarmuka pengguna, termasuk cara menggunakan fitur utama, pemecahan masalah umum, tips dan trik penggunaan, serta informasi kontak untuk dukungan teknis. Dokumen ini dirancang untuk membantu pengguna memaksimalkan manfaat dari sistem dengan pengalaman yang lancar dan memuaskan.

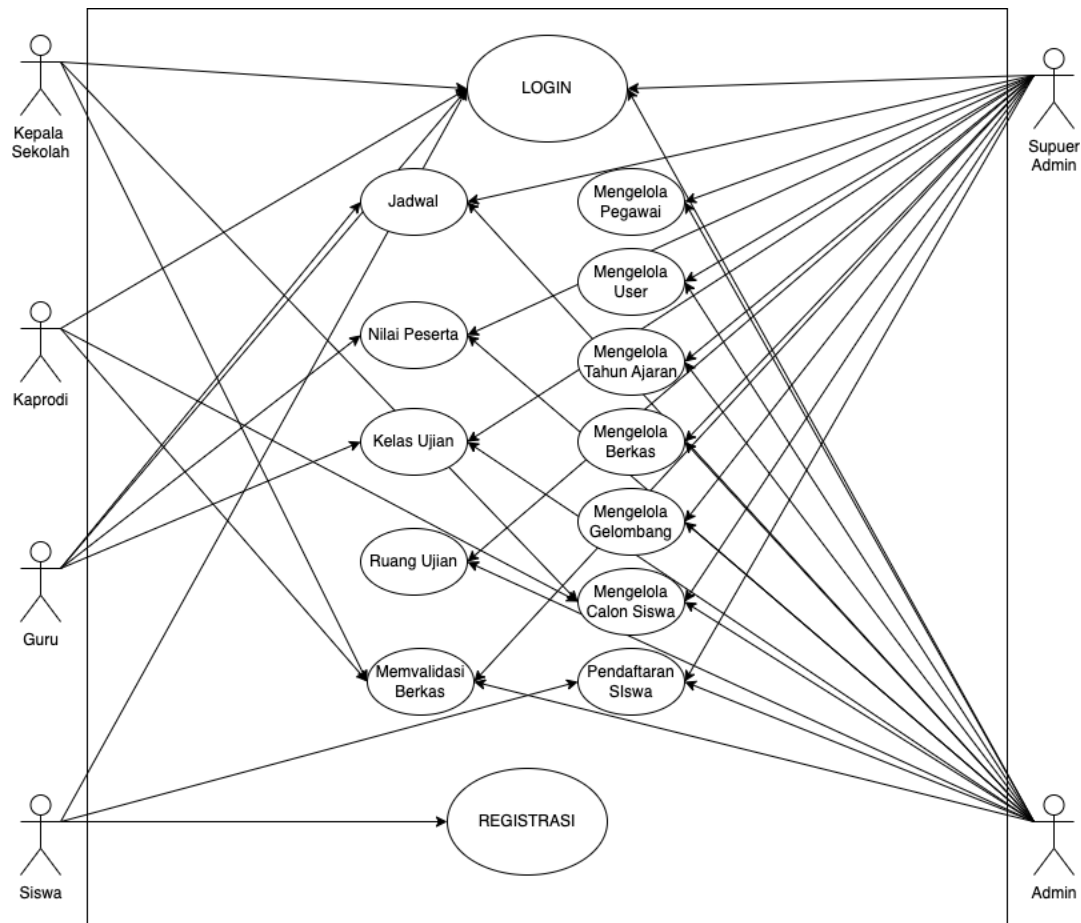
3. HASIL DAN PEMBAHASAN

3.1 Perancangan Sistem

Perancangan sistem pada penelitian ini untuk menggambarkan secara luas dan detail untuk membantu dalam pembuatan sistem aplikasi berbasis website.

3.1.1 Use Case Diagram

Use case adalah teknik pemodelan yang digunakan untuk menggambarkan interaksi antara pengguna (atau aktor) dengan sistem untuk mencapai tujuan tertentu. Setiap *use case* merepresentasikan skenario penggunaan yang menggambarkan bagaimana sistem seharusnya berperilaku dari perspektif pengguna. Dalam konteks pengembangan perangkat lunak, *use case* sering kali digunakan untuk mendefinisikan kebutuhan sistem dan mendokumentasikan interaksi antara aktor dan sistem.



Gambar 1. Diagram *Use Case*

Use case diagram pada gambar diatas mempunyai 6 pengguna, yaitu super admin, admin, kaprodi, kepala sekolah, guru dan siswa, yang mana Super Admin = Kepala Sekolah, Admin = Kaprodi, masing-masing pengguna memiliki hak akses yang berbeda

a. Super Admin/Kepala Sekolah

Dapat melakukan login, mengelola master akun dan berkas, registrasi siswa, master tahun ajaran dan gelombang, pendaftaran siswa, validasi berkas, dan seputar yang mengenai ujian

b. Admin/Kaprodi

Sama dengan Kepala Sekolah hanya saja tidak bisa mengelola master akun

c. Guru

Hanya bisa login, melihat jadwal pengawas dan menginput hasil test calon siswa

d. Siswa

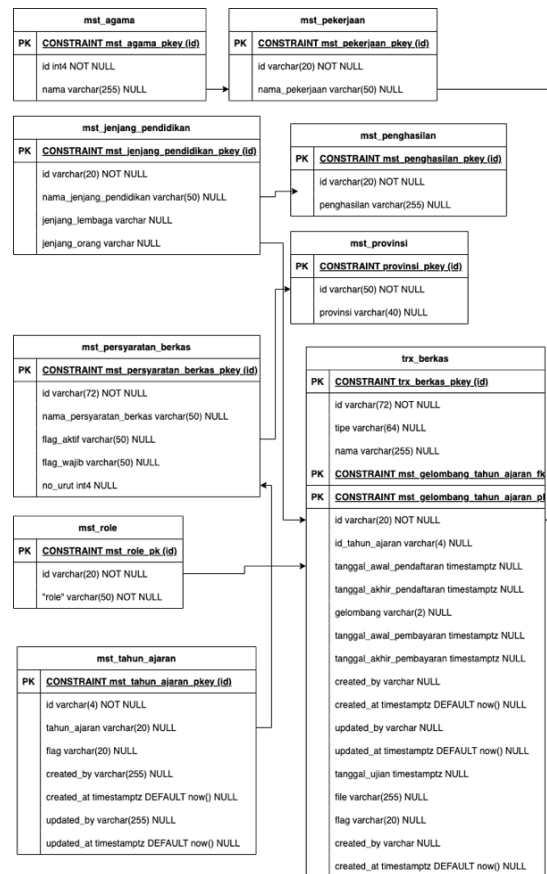
Dapat melakukan login, mendaftarkan diri, membayar keuangan, melihat jadwal ujian dan hasil test

Tabel 1. Use Case

Actor	Nama Use Case	Deskripsi
Admin, Kepala Sekolah	Mengelola Master Data	Pada fitur ini, user dapat mengelola data Pegawai, User dan Berkas Persyaratan.
Admin, Kaprodi, Kepala Sekolah	Mengelola Periode	Pada fitur ini, user dapat mengelola data Tahun Ajaran dan Gelombang
Admin, Calon Siswa	Mengelola Pendaftaran	Pada fitur ini, user dapat mengelola data Pendaftaran.
Admin, Kepala Sekolah, Kaprodi	Mengelola Calon Siswa	Pada fitur ini, user dapat mengelola data Calon Siswa.
Admin, Kaprodi	Mengelola Validasi Berkas	Pada fitur ini, user dapat mengelola data Validasi Berkas.
Admin, Kaprodi	Mengelola Test Akademik	Pada fitur ini, user dapat mengelola data Ruang Ujian, Kelas Ujian dan Peserta Ujian
Admin, Kaprodi, Kepala Sekolah, Calon Siswa, Guru	Mengelola Jadwal	Pada fitur ini, user dapat mengelola data Jadwal.
Admin, Kaprodi, Kepala Sekolah, Calon Siswa, Guru	Mengelola Hasil Test	Pada fitur ini, user dapat mengelola data Hasil Test.

3.1.2 Class Diagram

Class diagram adalah jenis diagram statis dalam Unified Modeling Language (UML) yang digunakan untuk menunjukkan struktur sistem dengan memodelkan kelas-kelas objek, atribut, operasi, dan hubungan antar kelas [7].

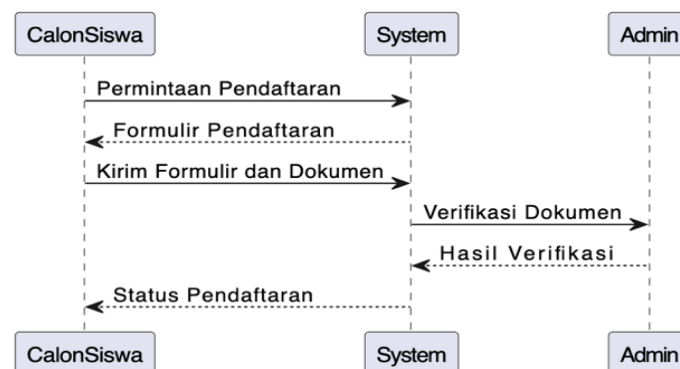


Gambar 2. Class Diagram

Class diagram PPDB menggambarkan hubungan antar entitas dalam sistem pendaftaran siswa baru. Calon siswa mengirimkan formulir dan dokumen ke sistem untuk mendaftar. Sistem kemudian mengirimkan formulir pendaftaran kepada calon siswa dan menerima dokumen yang diunggah untuk diverifikasi. Dokumen tersebut dikirimkan ke admin untuk proses verifikasi. Setelah itu, admin mengirimkan hasil verifikasi kembali ke sistem, yang akhirnya mengirimkan status pendaftaran kepada calon siswa. Dalam diagram ini, sistem berfungsi sebagai penghubung utama antara calon siswa dan admin, memastikan kelancaran alur pendaftaran dan verifikasi dokumen.

3.1.3 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk memvisualisasikan, merancang, dan mendokumentasikan sistem perangkat lunak. UML menyediakan serangkaian diagram yang dapat digunakan untuk merepresentasikan struktur, perilaku, dan interaksi berbagai komponen dalam sistem, yang bermanfaat untuk pengembangan dan pemeliharaan perangkat lunak [8].



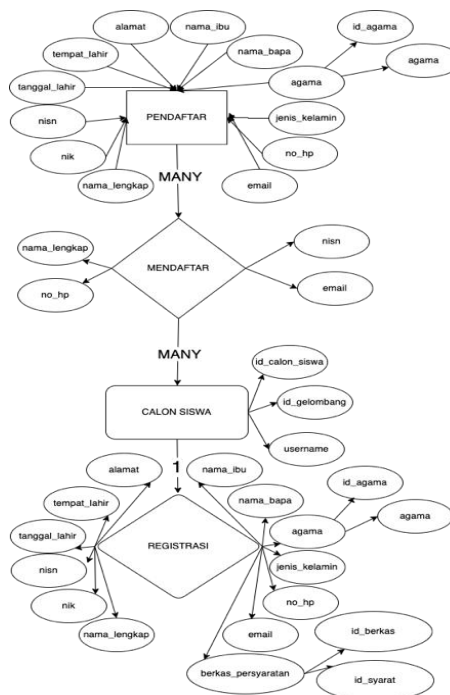
Gambar 3. Unified Modeling Language

Proses pendaftaran dimulai dengan calon siswa yang mengirimkan permintaan pendaftaran ke sistem. Setelah itu, sistem merespons dengan mengirimkan formulir pendaftaran yang harus diisi oleh calon siswa. Setelah formulir diisi, calon siswa mengirimkan formulir beserta dokumen yang diperlukan ke sistem. Dokumen-dokumen tersebut kemudian diteruskan oleh sistem ke admin untuk dilakukan verifikasi. Setelah admin selesai melakukan verifikasi, hasilnya dikirimkan kembali ke sistem. Terakhir, sistem mengirimkan status pendaftaran kepada calon siswa sebagai konfirmasi akhir dari proses pendaftaran.

3.1.4 Rancangan Basis Data

a. ERD

Entity-Relationship Diagram (ERD) adalah jenis diagram yang digunakan untuk memodelkan data dalam sistem dengan cara memvisualisasikan hubungan antara entitas (objek) yang terlibat. *ERD* sering digunakan dalam tahap perancangan basis data untuk mengidentifikasi dan memahami struktur data yang akan digunakan dalam suatu sistem informasi. Dalam *ERD*, entitas digambarkan sebagai kotak, sementara hubungan antar entitas dihubungkan dengan garis yang menunjukkan tipe relasi seperti satu-ke-satu, satu-ke-banyak, atau banyak-ke-banyak [9].



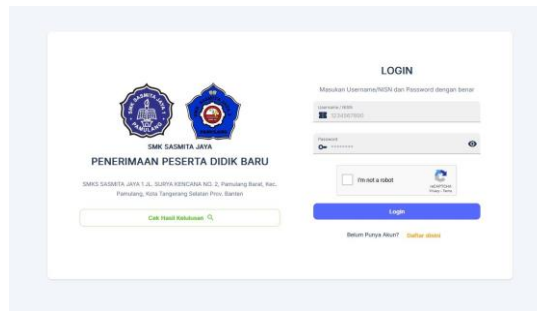
Gambar 4. Entity Relationship Diagram

3.2 Implementasi User Interface

User Interface (UI) adalah bagian dari sistem perangkat lunak yang memungkinkan interaksi antara pengguna dan perangkat lunak tersebut. UI mencakup elemen visual seperti tombol, ikon, dan menu yang dirancang untuk membantu pengguna dalam mengakses fitur dan fungsi aplikasi dengan mudah. Tujuan utama UI adalah menyediakan antarmuka yang intuitif dan mudah digunakan sehingga pengalaman pengguna dalam berinteraksi dengan sistem menjadi efisien dan menyenangkan. UI yang baik akan mempertimbangkan aspek kemudahan penggunaan (*usability*), daya tarik visual, dan konsistensi dalam desain [10].

3.2.1 Halaman Login Siswa

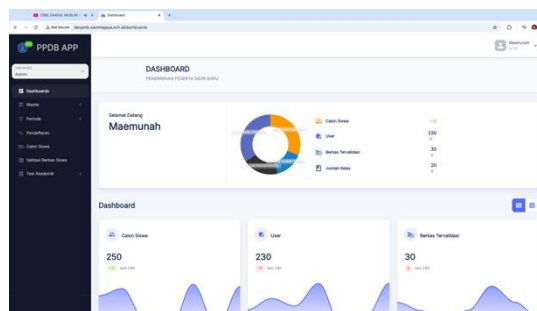
Halaman pertama untuk masuk ke dalam aplikasi. Dengan adanya fungsi perangkat lunak ini dapat membatasi hak akses bagi user yang dapat melihat dan berinteraksi dengan PPDB SMK. Tampilan halaman login sebagai berikut:



Gambar 5. Tampilan Login Siswa

3.2.2 Dashboard Admin

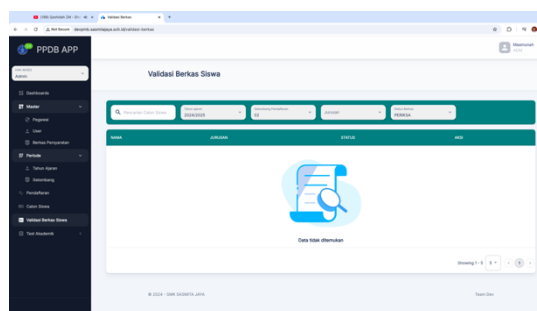
Halaman untuk mengetahui graphic data calon siswa, dan pendaftaran untuk admin.



Gambar 6. Dashboard Admin

3.2.3 Menu Admin

Halaman untuk mengelola semua data calon siswa yang ada di aplikasi, kurang lebih dengan design seperti ini hanya menyajikan table dan filter untuk mencari data siswa, users, dll.



Gambar 7. Menu Admin

3.3 Unit Testing

Unit testing merupakan tahap dasar dalam pengujian. Unit testing berfokus pada pengujian building block yang lebih kecil daripada program atau sistem yang diuji. Pengujian ini mengeksekusi setiap fitur atau fungsionalitas untuk memastikan masing-masing fitur atau fungsionalitas tersebut berfungsi sesuai dengan yang diharapkan [11]. Pada definisi lainnya, unit testing merupakan pengujian yang mencakup pengujian sepotong atau sebagian kode [12]. Dengan demikian, pengujian ini dapat digunakan untuk menguji kelas atau modul tertentu

Tabel 2. Unit Testing

Fitur	Id Case	Description	Pre Condition	Test Steps	Expected Result	Actual Result	Status
Login	TC01	Login admin dengan data yang valid	Berada di menu Login Admin	1. Masukan Username 2. Masukan Password 3. Centang reCAPTCHA 4. Klik Login	Masuk kehalaman dashboard admin	Sesuai	Pass
Login	TC01	Login siswa dengan data yang valid	Berada di menu Login Siswa	1. Masukan Username 2. Masukan Password 3. Centang reCAPTCHA 4. Klik Login	Masuk kehalaman dashboard siswa	Sesuai	Pass
Register	TC01	Register siswa dengan data yang valid	Berada di menu register Siswa	1. Masukan Nisn 2. Masukan Nama Lengkap 3. Masukan Email 4. Masukan Password 5. Masukan Konfirmasi Password 3. Centang reCAPTCHA 4. Klik Daftar	Kembali kehalaman Login dengan alert berhasil mendaftar	Sesuai	Pass
Pegawai	TC01	Tambah Pegawai dengan data valid	Berada di menu master pegawai	1. Klik Tambah Pegawai 2. Masukan User Id Pegawai 3. Masukan Role 4. Klik Daftar	Menutup Dialog Tambah pegawai dengan alert berhasil	Sesuai	Pass
Pegawai	TC01	Pencarian Pegawai dengan data yang valid	Berada di menu master pegawai	1. Masukan Nama dikolom Pencarian	Nama Akan Muncul jika sesuai, dan jika tidak ada muncul ilustrasi	Sesuai	Pass
User	TC01	Pencarian User dengan data yang valid	Berada di menu master user	1. Masukan nama dikolom pencarian	Nama Akan Muncul jika sesuai, dan jika	Sesuai	Pass
Berkas	TC01	Tambah berkas persyaratan dengan data yang valid	Berada di menu master berkas	1. Klik Tambah berkas 2. Masukan nama berkas 3. Pilih Status Aktif 4. Pilih Status Berkas 5. Klik Simpan	Menutup Dialog Tambah berkas dengan alert berhasil	Sesuai	Pass
Berkas	TC01	Update berkas persyaratan dengan data yang valid	Berada di menu master berkas	1. Klik dropdown berkas 2. Klik icon pensil 3. Masukan nama berkas 3. Pilih Status Aktif 4. Pilih Status Berkas 5. Klik Simpan	Menutup Dialog Tambah berkas dengan alert berhasil	Sesuai	Pass
Berkas	TC01	Delete berkas persyaratan	Berada di menu master berkas	1. Klik dropdown berkas 2. Klik icon sampah 5. Klik Simpan	Menutup Dialog Tambah berkas dengan alert berhasil	Sesuai	Pass
Tahun Ajaran	TC01	Tambah tahun ajaran	Berada di menu periode tahun ajaran	1. Klik tambah tahun ajaran 2. Masukan Tahun Sekaran	Menutup Dialog Tambah tahun ajaran dengan alert berhasil	Sesuai	Pass

				3. Masukan Tahun Ajaran 4. Pilih Status 5. Klik Simpan			
Tahun Ajaran	TC01	Update tahun ajaran	Berada di menu periode tahun ajaran	1. Klik Aksi 2. Klik Update 3. Masukan Tahun Ajaran 4. Pilih Status 5. Klik Simpan	Menutup Dialog Update tahun ajaran dengan alert berhasil	Sesuai	Pass
Gelombang	TC01	Tambah Gelombang	Berada di menu periode gelombang	1. Klik tambah gelombang 2. Pilih Gelombang 3. Pilih Tahun Ajaran 4. Masukan Tanggal Awal dan akhir pendaftaran 5. Masukan Tanggal awal dan akhir Pembayaran 6. Masukan Tanggal Ujian 7. Klik Simpan	Menutup Dialog Tambah gelombang dengan alert berhasil	Sesuai	Pass
Gelombang	TC01	Update Gelombang	Berada di menu periode gelombang	1. Klik icon pensil 2. Masukan Tanggal Awal dan akhir pendaftaran 3. Masukan Tanggal awal dan akhir Pembayaran 4. Masukan Tanggal Ujian 5. Klik Simpan	Menutup Dialog Update gelombang dengan alert berhasil	Sesuai	Pass
Pendaftaran	TC01	Pendaftaran calon siswa	Berada di menu pendaftaran siswa	1. Pilih Jurusan 2. Klik Next 3. Masukan Biodata Calon Siswa 4. Klik Next 5. Masukan Biodata Orang tua wali 6. Klik Next 7. Upload Berkas 8. Klik Next 9. Klik Submit Pendaftaran	Muncul dialog berhasil mendaftar dan Kembali ke menu pendaftaran	Sesuai	Pass

4. KESIMPULAN

Kesimpulan dari pengembangan dan penerapan Sistem Penerimaan Peserta Didik Baru (PPDB) terintegrasi di SMK Sasmita Jaya menunjukkan keberhasilan signifikan dalam memperlancar proses penerimaan siswa baru. Sistem ini dibangun menggunakan *TypeScript* dengan dukungan arsitektur berbasis web, memberikan kemudahan bagi calon siswa untuk melakukan pendaftaran secara online, verifikasi dokumen, dan penyelesaian pembayaran melalui virtual account. Fitur integrasi pembayaran memungkinkan otomatisasi dalam proses rekonsiliasi keuangan, sehingga mengurangi beban kerja administratif dan meminimalkan potensi kesalahan manusia dalam pencatatan data. Sebelum implementasi sistem, proses pendaftaran manual sering kali melibatkan beberapa tahap yang memerlukan keterlibatan langsung antara calon siswa dan pihak sekolah, yang cenderung lambat dan rentan terhadap kesalahan.

Penggunaan arsitektur berbasis web pada sistem ini tidak hanya meningkatkan efisiensi waktu tetapi juga memberikan fleksibilitas akses bagi calon siswa yang tidak harus hadir secara fisik untuk menyelesaikan seluruh proses administrasi. Dari evaluasi yang dilakukan, sistem ini menunjukkan efektivitas yang tinggi dalam mempercepat waktu pemrosesan serta meningkatkan kepuasan pengguna, baik dari sisi administrator yang mengelola data siswa baru maupun calon siswa yang melalui proses pendaftaran. Peningkatan signifikan dalam efisiensi ini menjadi bukti manfaat dari digitalisasi proses dalam dunia pendidikan, terutama untuk kebutuhan administrasi yang sering kali memerlukan ketelitian tinggi dan efisiensi waktu.

Secara keseluruhan, penerapan sistem digital PPDB ini membuktikan bahwa teknologi informasi mampu memberikan solusi konkret dalam menangani tantangan administratif. Selain itu, dengan adanya peningkatan efisiensi dan akurasi, sistem ini mampu menjadi model pengembangan PPDB digital untuk institusi pendidikan lainnya. Kesuksesan ini membuka peluang untuk iterasi dan pengembangan lebih lanjut di masa mendatang, yang dapat mencakup integrasi fitur-fitur tambahan yang mendukung pengalaman pengguna, seperti pelaporan otomatis dan peningkatan keamanan data. Dengan terus diperbaiki, sistem ini diharapkan dapat mendukung pengelolaan penerimaan siswa baru secara lebih efektif dan menjadi landasan bagi implementasi digitalisasi lebih luas dalam sektor pendidikan.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini, terutama kepada kedua orangtua yang selalu mensupport kami, guru-guru yang membimbing kami, teman-teman yang menyemangati, semoga penelitian ini bermanfaat bagi diri kami, dan bagi khalayak ramai.

REFERENCES

- [1] Susanto, "Digital Transformation in Student Admission Systems: A Case Study," *Journal of Educational Technology*, vol. 15, no. 3, pp. 215-228, 2022.
- [2] R. Iskandar, "Web-Based Systems for Education Management: Accessibility and Efficiency," *Education and Technology Journal*, vol. 9, no. 2, pp. 123-137, 2021.
- [3] T. Mulyana, "Impact of Virtual Account Integration on Administrative Efficiency," *Journal of Financial Technology*, vol. 7, no. 1, pp. 50-65, 2023.
- [4] Haryanto et al., "Benefits of TypeScript in Large-Scale Web Applications: A Case Study in Education Sector," *Journal of Software Engineering*, vol. 14, no. 2, pp. 102-115, 2022.
- [5] M. Putra, "Challenges in Online Enrollment Systems and Solutions," *International Journal of Education and Development*, vol. 10, no. 4, pp. 389-403, 2023.
- [6] B. Haryanto et al., "Benefits of TypeScript in Large-Scale Web Applications: A Case Study in Education Sector," *Journal of Software Engineering*, vol. 14, no. 2, pp. 102-115, 2022.
- [7] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd ed. Addison-Wesley, 2003.
- [8] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Addison-Wesley Professional, 2018.
- [9] T. Connolly and C. Begg, *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6th ed. Pearson Education, 2015.
- [10] A. Dennis, B. H. Wixom, and D. Tegarden, *Systems Analysis and Design with UML*, 5th ed. John Wiley & Sons, 2015.
- [11] W. Lewis, *Software Testing and Continuous Quality Improvement*, Third Edition., Boston: Auerbach Publication, 2009.
- [12] R. Black, *Managing the Testing Process: Practical Tolls and Techniques for Managing Hardware and Software Testing*, 2nd Edition., Hoboken: Wiley Publishing Inc, 2002.
- [13] M. H. Handoko and A. B. Putra, "Sistem Informasi PPDB Berbasis Web di SMK XYZ," *Jurnal Teknologi Pendidikan*, vol. 18, no. 2, pp. 120-130, 2023.
- [14] N. Rahmawati and T. Kurniawan, "Integrasi Virtual Account untuk Sistem Pembayaran Pendidikan," *Journal of Information Systems*, vol. 15, no. 1, pp. 45-55, 2022.