



KERJA PRAKTIK - EF234603

Software House: Analisis dan Implementasi Pengembangan Perangkat Lunak Sistem UTBK di PT Aksamedia Mulia Digital

Aksamedia Mulia Digital, Kapas Madya II No. 18A, Gading, Kec. Tambaksari, Surabaya, Jawa Timur 60134

Periode: 6 September 2024 - 31 Desember 2024

Oleh:

M Naufal Badruttamam 5025211240

Pembimbing Jurusan

Hadziq Frabroyir, S.Kom., Ph.D.

Pembimbing Lapangan

Imro'atul Hasanah

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2025

[Halaman ini sengaja dikosongkan]



KERJA PRAKTIK - EF234603

Software House: Analisis dan Implementasi Pengembangan Perangkat Lunak Sistem UTBK di PT Aksamedia Mulia Digital

Aksamedia Mulia Digital, Kapas Madya II No.18A, Gading, Kec. Tambaksari, Surabaya, Jawa Timur 60134

Periode: 6 September 2024 - 31 Desember 2024

Oleh:

M Naufal Badruttamam 5025211240

Pembimbing Jurusan

Hadziq Fabroyir, S.Kom., Ph.D.

Pembimbing Lapangan

Imro'atul Hasanah

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2025

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI	5
DAFTAR GAMBAR	10
DAFTAR TABEL	11
DAFTAR KODE SUMBER	12
LEMBAR PENGESAHAN KERJA PRAKTIK	14
ABSTRAK	16
KATA PENGANTAR	19
BAB I PENDAHULUAN	21
1.1 Latar Belakang	21
1.2 Tujuan	22
1.3 Manfaat	22
1.4 Rumusan Masalah	23
1.5 Lokasi dan Waktu Kerja Praktik	23
1.6 Metodologi Kerja Praktik	23
1.6.1 Perumusan Masalah	23
1.6.2 Studi Literatur	24
1.6.3 Analisis dan Perancangan Sistem.....	24
1.6.4 Implementasi Sistem	24
1.6.5 Pengujian dan Evaluasi	25

1.6.6	Kesimpulan dan Saran.....	25
1.7	Sistematika Laporan.....	26
1.7.1	Bab I Pendahuluan	26
1.7.2	Bab II Profil Perusahaan.....	26
1.7.3	Bab III Tinjauan Pustaka.....	26
1.7.4	Bab IV Analisis dan Perancangan Sistem	26
1.7.5	Bab V Implementasi Sistem.....	26
1.7.6	Bab VI Pengujian dan Evaluasi.....	27
1.7.7	Bab VII Kesimpulan dan Saran.....	27
BAB II.....		28
PROFIL PERUSAHAAN.....		28
2.1	Profil PT Aksamedia Mulia Digital.....	28
2.2	Lokasi.....	30
BAB III TINJAUAN PUSTAKA.....		32
3.1.	Sistem Ujian Tulis Berbasis Komputer	32
3.2.	<i>Software Developer</i>	33
3.3.	Agile	34
3.4.	<i>Virtual Host</i>.....	35
3.5.	GitLab	37
3.6.	<i>Structured Query Language (SQL)</i>	38
3.7.	Laravel.....	39
3.8.	Docker.....	41

3.9. Repositori Kode.....	43
BAB IV.....	45
ANALISIS DAN PERANCANGAN SISTEM.....	45
4.1 Analisis Sistem	45
4.1.1 Definisi Umum Aplikasi	45
4.1.2 Analisis Kebutuhan	46
4.1.3 Diagram Use Case.....	48
4.2 Perancangan Sistem.....	49
4.2.1 <i>Entity Relationship Diagram</i>	50
4.3 Infrastruktur Website	54
BAB V IMPLEMENTASI SISTEM.....	57
5.1. Implementasi Kerangka Kode Sumber	57
5.2. Implementasi Kerangka Kode Sumber	59
5.3. Implementasi Antarmuka Pengguna	66
BAB VI PENGUJIAN DAN EVALUASI.....	81
6.1. Tujuan Pengujian.....	81
6.2. Kriteria Pengujian	81
6.3. Skenario Pengujian.....	82
6.4. Evaluasi Pengujian.....	82
BAB VII.....	85
KESIMPULAN DAN SARAN.....	85
7.1. Kesimpulan.....	85

7.2. Saran	86
DAFTAR PUSTAKA.....	87
BIODATA PENULIS	91

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1	Logo PT Aksamedia Mulia Digital	29
Gambar 4.1	Diagram Use Case Admin	48
Gambar 4.2	Diagram Use Case Sekolah	49
Gambar 4.3	Diagram Use Case Siswa.....	49
Gambar 4.4	Sistem ERD pada aplikasi UTBK (Bagian 1)	51
Gambar 4.5	Sistem ERD pada aplikasi UTBK (Bagian 2).....	51
Gambar 4.6	Sistem ERD pada aplikasi UTBK (Bagian 3).....	51
Gambar 4.7	Sistem ERD pada aplikasi UTBK (Bagian 4).....	52
Gambar 4.8	Sistem ERD pada aplikasi UTBK (Bagian 5).....	52
Gambar 4.9	Infrastruktur aplikasi Web.....	55
Gambar 5.1	Struktur folder kerangka kode	57
Gambar 5.2	Halaman Login Admin	67
Gambar 5.3	Halaman Dashboard Admin	68
Gambar 5.4	Halaman Dashboard Siswa.....	69
Gambar 5.5	Halaman Manajemen Test.....	70
Gambar 5.6	Halaman Manajemen Tryout.....	70
Gambar 5.7	Halaman Manajemen Penjadwalan	71
Gambar 5.8	Halaman Manajemen Master Data	72
Gambar 5.9	Halaman Login Sekolah	72
Gambar 5.10	Halaman Dashboard Sekolah	73
Gambar 5.11	Halaman Dashboard Daftar Tes Sekolah dan Peserta	74
Gambar 5.12	Halaman Status Peserta	75
Gambar 5.13	Halaman View Assesment Report.....	75
Gambar 5.14	Halaman Login Siswa.....	76
Gambar 5.15	Halaman Dashboard Awal Siswa	77
Gambar 5.16	Halaman Pemilihan Kampus dan Jurusan	77
Gambar 5.17	Halaman Pengerjaan Soal.....	78
Gambar 5.18	Halaman Hasil Pengerjaan oleh Siswa	79

DAFTAR TABEL

Tabel 4.1 Kebutuhan Fungsional Sistem Sekolah.....	46
Tabel 4.2 Kebutuhan Fungsional Sistem Sekolah.....	47
Tabel 4.3 Kebutuhan Fungsional Sistem Siswa.....	47
Tabel 4.4 Kebutuhan Non Fungsional Aplikasi.....	47

DAFTAR KODE SUMBER

Kode Sumber 5.1 Implementasi Frontend dengan Form Data.....	64
Kode Sumber 5.2 Implementasi Backend pada Store	66

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

KERJA PRAKTIK

Software House: Analisis dan Implementasi Pengembangan Perangkat Lunak Sistem UTBK di PT Aksamedia Mulia Digital

Oleh:

M Naufal Badruttamam

5025211240

Disetujui oleh Pembimbing Kerja Praktik:

1. Hadziq Fabroyir, S,Kom.,
Ph.D.
NIP.
198602272019031006



(Pembimbing Departemen)

2. Imro'atul Hasanah



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Software House: Analisis dan Implementasi Pengembangan Perangkat Lunak Sistem UTBK di PT Aksamedia Mulia Digital

Nama Mahasiswa : M Naufal Badruttamam
NRP : 5025211240
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Hadziq Fabroyir, S.Kom., Ph.D.
Pembimbing Lapangan : Imro'atul Hasanah

ABSTRAK

Kerja praktik ini dilaksanakan di PT Aksamedia Mulia Digital yang bergerak di bidang pengembangan perangkat lunak. Perusahaan ini mendapatkan proyek pengembangan sistem Ujian Tulis Berbasis Komputer (UTBK) dari client Edubrand. Sistem UTBK ini dikembangkan untuk memfasilitasi pelaksanaan ujian secara digital dengan tiga role pengguna yaitu admin, sekolah, dan siswa. Panel admin dan sekolah dikembangkan menggunakan Laravel Blade dan Bootstrap, sementara panel siswa menggunakan Vue.js untuk memberikan pengalaman ujian yang lebih interaktif. Dalam pengembangan sistem ini, diimplementasikan berbagai fitur keamanan seperti autentikasi multi-role, sistem anti-kecurangan, dan manajemen sesi untuk menjaga integritas pelaksanaan ujian. Sistem juga dilengkapi dengan fitur manajemen bank soal, penjadwalan ujian, monitoring real-time, dan pelaporan hasil yang komprehensif. Pengujian dilakukan untuk memastikan sistem dapat menangani banyak pengguna secara bersamaan dan tetap memberikan performa yang optimal. Hasil dari kerja praktik ini adalah sebuah sistem

UTBK yang telah berhasil diimplementasikan dan memenuhi kebutuhan Edubrand. Sistem ini mampu mengelola proses ujian secara efisien, menjaga keamanan data, dan memberikan pengalaman pengguna yang baik untuk semua role

Kata Kunci: UTBK, Laravel, Sistem Ujian Online, Multi-Role Authentication

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Software House: Analisis dan Implementasi Pengembangan Perangkat Lunak Sistem UTBK di PT Aksamedia Mulia Digital. Penulis menyadari bahwa masih banyak ruang perkembangan baik dalam melaksanakan kerja praktik dan magang maupun penyusunan buku laporan ini.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Keluarga penulis yang selalu mendukung penulis.
2. Bapak Hadziq Fabroyir selaku dosen pembimbing kerja praktik sekaligus koordinator kerja praktik.
3. Ibu Imro'atul Hasnah selaku pembimbing lapangan dan mentor selama kerja praktik berlangsung.
4. Tim Aksamedia yang senantiasa membantu memberikan ilmu yang berharga dan teman-teman penulis yang memberikan semangat kepada penulis dalam melaksanakan KP.

Surabaya, 10 Februari 2025
M Naufal Badruttamam

[Halaman ini sengaja dikosongkan]

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin pesat mendorong perusahaan untuk terus berinovasi dalam mengembangkan perangkat lunak yang adaptif terhadap kebutuhan pengguna. *Software house* sebagai perusahaan yang bergerak di bidang pengembangan perangkat lunak memiliki peran penting dalam memenuhi kebutuhan digitalisasi berbagai sektor, termasuk sektor pendidikan.

PT Aksamedia Mulia Digital merupakan salah satu *software house* yang berlokasi di Surabaya dan telah berpengalaman dalam mengembangkan berbagai solusi perangkat lunak untuk kebutuhan *client* dari berbagai industri. Sebagai perusahaan *software house*, PT Aksamedia Mulia Digital menyediakan pengembangan perangkat lunak yang komprehensif, mulai dari pembuatan *website*, aplikasi *mobile*, hingga sistem *enterprise*.

Edubrand sebagai salah satu *client* PT Aksamedia Mulia Digital membutuhkan pengembangan sistem Ujian Tulis Berbasis Komputer (UTBK) untuk mendukung kegiatan ujian secara digital. Sistem UTBK ini memerlukan pengembangan yang cermat untuk memastikan keamanan, kehandalan, dan kemudahan penggunaan bagi peserta ujian. Pengembangan sistem ini juga harus mempertimbangkan berbagai aspek seperti manajemen soal, penilaian otomatis, dan pelaporan hasil ujian yang akurat sesuai dengan standar dan kebutuhan Edubrand.

1.2 Tujuan

Tujuan kerja praktik ini adalah untuk menyelesaikan kewajiban akademik dengan melakukan pelaporan kerja praktik, menganalisis, serta mengimplementasikan pengembangan sistem UTBK sesuai dengan kebutuhan Edubrand. Selain itu, kerja praktik ini bertujuan untuk memahami proses pengembangan sistem ujian digital yang aman dan efektif.

1.3 Manfaat

Manfaat yang diperoleh adalah dapat memahami proses pengembangan sistem UTBK secara komprehensif, mulai dari analisis kebutuhan hingga implementasi. Pengalaman ini juga memberikan pemahaman tentang aspek-aspek penting dalam pengembangan sistem ujian digital, termasuk keamanan data dan manajemen pengguna sesuai dengan standar yang ditetapkan oleh Edubrand.

1.4 Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut,

1. Bagaimana cara menganalisis kebutuhan sistem UTBK untuk memastikan fungsionalitas yang sesuai dengan keperluan Edubrand?
2. Bagaimana cara mengimplementasikan fitur-fitur keamanan dalam sistem UTBK untuk menjaga integritas ujian?
3. Bagaimana cara mengelola dan mengoptimalkan kinerja sistem UTBK saat diakses oleh banyak pengguna secara bersamaan?

1.5 Lokasi dan Waktu Kerja Praktik

Kerja praktik dilaksanakan secara luring di Kapas Madya II No. 18A, Gading, Kec. Tambaksari, Surabaya, Jawa Timur 60134. Kegiatan ini berlangsung mulai tanggal 6 September 2024 hingga 28 Desember 2024. Kegiatan dilakukan mengikuti jam kerja pukul 09.00 - 17.00 untuk hari Selasa hingga Kamis dan Daring untuk hari Senin dan Jumat.

1.6 Metodologi Kerja Praktik

Berikut adalah metodologi yang digunakan dalam pembuatan laporan kerja praktik.

1.6.1 Perumusan Masalah

Kebutuhan data yang diinginkan didapatkan dari *user requirement system* (URS) melalui diskusi dengan pihak Edubrand. URS diberikan oleh Imro'atul Hasanah

selaku pembimbing lapangan dan penjelasan lebih lanjut mengenai kebutuhan sistem UTBK dijelaskan oleh Amridzal Jundi yang secara langsung bertemu dengan pihak Edubrand. Setelah mengetahui kebutuhan data yang diinginkan, dilakukan diskusi untuk merumuskan solusi perancangan dan analisis terhadap kebutuhan dan pengembangan sistem UTBK.

1.6.2 Studi Literatur

Setelah menentukan teknologi yang akan digunakan, dilakukan studi literatur untuk memahami lebih dalam mengenai teknologi yang relevan dengan pengembangan sistem UTBK. Studi literatur ini mencakup penelitian mengenai sistem ujian *online*, keamanan data, manajemen pengguna, serta studi kasus implementasi teknologi dalam proyek serupa.

1.6.3 Analisis dan Perancangan Sistem

Pada tahap ini, dilakukan analisis terhadap kebutuhan sistem UTBK berdasarkan URS yang telah diperoleh dari Edubrand. Analisis mencakup aspek fungsional sistem, keamanan data, manajemen soal, dan pelaporan hasil ujian. Setelah itu, dilakukan perancangan sistem yang mencakup arsitektur sistem, desain *database*, dan alur kerja aplikasi agar sesuai dengan kebutuhan Edubrand.

1.6.4 Implementasi Sistem

Pada tahap ini, dilakukan implementasi dari rancangan sistem UTBK yang telah dibuat sebelumnya. Pemantauan pengerjaan berbentuk rapat progres setiap

hari dengan umpan balik dari pembimbing lapangan untuk memastikan kesesuaian dengan kebutuhan Edubrand.

1.6.5 Pengujian dan Evaluasi

Setelah sistem selesai diimplementasikan, dilakukan pengujian dan evaluasi untuk memastikan bahwa sistem UTBK berjalan sesuai dengan kebutuhan yang telah dirumuskan. Hasil evaluasi digunakan untuk perbaikan dan penyempurnaan sistem sebelum akhirnya diajukan kepada Edubrand untuk digunakan secara resmi.

1.6.6 Kesimpulan dan Saran

Pada tahap ini, disusun kesimpulan dan saran berdasarkan hasil kerja praktik yang telah dilakukan.

1.7 Sistematika Laporan

Laporan kerja praktik terdiri dari tujuh bab dengan rincian sebagai berikut.

1.7.1 Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2 Bab II Profil Perusahaan

Bab ini berisi gambaran umum PT Aksamedia Mulia Digital dari profil dan lokasi perusahaan.

1.7.3 Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4 Bab IV Analisis dan Perancangan Sistem

Bab ini menjelaskan analisis kebutuhan sistem, strategi dalam menangani banyak proyek yang berjalan secara paralel, perancangan sistem, serta desain arsitektur dan *database* yang akan digunakan dalam proyek kerja praktik.

1.7.5 Bab V Implementasi Sistem

Bab ini membahas implementasi sistem yang telah dikembangkan, termasuk teknologi yang digunakan dan proses pengembangannya.

1.7.6 Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7 Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

BAB II

PROFIL PERUSAHAAN

2.1 Profil PT Aksamedia Mulia Digital

PT Aksamedia memulai perjalanannya di Kota Malang pada tahun 2012, didirikan oleh dua profesional teknologi berpengalaman, Amiridzal Jundi, S.Kom dan Siswoyo, S.Kom. Pada awalnya, perusahaan berfokus pada pengembangan website, khususnya pembuatan company profile digital, yang menjadi landasan penting dalam membangun keahlian dan reputasi di industri teknologi digital. Tahun 2014 menjadi titik balik strategis ketika Aksamedia merelokasi operasinya ke Surabaya untuk merespons meningkatnya permintaan pasar. Perpindahan ini membuka peluang ekspansi lebih luas, memungkinkan perusahaan menjangkau basis klien yang lebih beragam. Pada tahun 2015, Aksamedia semakin memperkuat kehadirannya dengan beroperasi dari Ruko Icon 21 Merr, yang menjadi pusat inovasi dan pengembangan digital. Transformasi besar terjadi pada 2016 dengan perubahan status hukum menjadi CV Aksamedia Global serta ekspansi layanan ke pengembangan aplikasi *mobile*, khususnya Android. Periode 2017–2018 menjadi masa pertumbuhan pesat dengan peningkatan signifikan dalam sumber daya manusia, hingga mencapai 25 programmer berdedikasi, memperkuat kapasitas perusahaan dalam menangani proyek berskala besar dan kompleks



Gambar 2.1 Logo PT Aksamedia Mulia Digital

Puncak evolusi perusahaan terjadi pada tahun 2022 dengan transformasi menjadi PT Aksamedia Mulia Digital, menandai komitmen perusahaan untuk memberikan layanan profesional berkualitas tinggi. Saat ini, dengan lebih dari 30 programmer ahli, Aksamedia telah membangun portofolio klien yang mengesankan, melayani berbagai perusahaan dan institusi terkemuka di Indonesia, termasuk PT Santos Jaya Abadi, PT Pelindo, Universitas Negeri Surabaya (UNESA), PT Pertamina EP Cepu, dan Kementerian Kesehatan RI. Keberhasilannya juga dibuktikan dengan kepercayaan dari merek ternama seperti Hayyu, Parahita, National Hospital, dan Kampong Steak. Dari *startup* teknologi di Malang hingga menjadi pemain utama dalam industri digital nasional, perjalanan Aksamedia mencerminkan komitmen berkelanjutan terhadap inovasi, kualitas, dan kepuasan klien. Dengan fondasi kuat yang telah dibangun selama lebih dari satu dekade, Aksamedia terus memosisikan diri di garis depan transformasi digital Indonesia, siap menghadapi

tantangan dan peluang di masa depan.

2.2 Lokasi

PT Aksamedia Mulia Digital berlokasi di Kapas Madya II No. 18A, Gading, Kec. Tambaksari, Surabaya, Jawa Timur 60134. Kantor tersebut juga menjadi lokasi kerja praktik dari penulis.

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. Sistem Ujian Tulis Berbasis Komputer

Sistem Ujian Tulis Berbasis Komputer atau *Computer-Based Test* (CBT) merupakan metode evaluasi pembelajaran yang menggunakan perangkat komputer sebagai media utama dalam pelaksanaan ujian. Menurut (Yanti & Siswandi, 2024), metode ini menawarkan peningkatan efisiensi dalam proses evaluasi dan pengelolaan data hasil ujian. CBT juga mendukung pengacakan soal secara otomatis, sehingga meningkatkan keamanan dan objektivitas penilaian.

Dalam pengembangannya, (Ulum & Hidayat, 2024) menjelaskan bahwa sistem CBT berbasis web memungkinkan proses pengelolaan ujian yang lebih fleksibel dan efisien melalui metode Agile dengan model Extreme Programming. Studi yang dilakukan oleh (Senja Sis Ha Abadi et al., 2024) menunjukkan bahwa siswa memberikan respons positif terhadap penggunaan CBT, terutama dalam aspek kepercayaan dan kenyamanan, meskipun kendala teknis seperti gangguan jaringan masih menjadi tantangan.

Keamanan menjadi aspek penting dalam implementasi (Sosnowski et al., 2025) merekomendasikan penerapan metode autentikasi ganda, enkripsi data, serta pemantauan aktivitas peserta untuk menjaga integritas pelaksanaan ujian. Sementara itu,

(Kamaludin, 2023) menyoroiti penggunaan analisis butir soal pada CBT berbasis web sebagai alat bantu dalam mengevaluasi tingkat kesulitan soal dan meningkatkan kualitas evaluasi pembelajaran.

3.2. *Software Developer*

Peran *software developer* mengalami perubahan signifikan seiring dengan pesatnya transformasi digital. (Laato Samuliand Mäntymäki, 2021) menyatakan bahwa *developer* modern tidak hanya berfokus pada pemrograman, tetapi juga berperan dalam analisis kebutuhan bisnis dan solusi teknologi yang kompleks. Studi ini menyoroiti bahwa peningkatan otomatisasi dan digitalisasi mendorong *developer* untuk memiliki pemahaman yang lebih dalam terhadap proses bisnis.

Selain aspek teknis, (Masrohatin et al., 2022) menekankan pentingnya penguasaan teknologi seperti *Internet of Things*, *Big Data*, dan *Artificial Intelligence* dalam menghadapi revolusi industri 4.0. *Developer* yang memahami teknologi ini mampu menghasilkan solusi yang lebih efektif dan inovatif.

Transformasi ini juga berdampak pada kompetensi yang dibutuhkan. (Carey, 2008) menunjukkan bahwa kolaborasi antara *developer* dan pemangku kepentingan bisnis menjadi faktor kunci dalam menghasilkan solusi yang relevan dengan kebutuhan pasar. *Developer* yang memiliki kemampuan komunikasi yang baik lebih mampu menerjemahkan kebutuhan bisnis menjadi aplikasi teknologi yang efektif.

Selain itu, (Alt Rainerand Auth, 2019) menyoroti peran *DevOps* dalam menjembatani kebutuhan bisnis dengan pengembangan perangkat lunak, mempercepat proses pengembangan, dan meningkatkan kualitas produk.

3.3. Agile

Metodologi Agile telah menjadi pendekatan dominan dalam pengembangan perangkat lunak modern. (Beck et al., 2023) dalam penelitiannya mengungkapkan bahwa adopsi Agile telah meningkat secara signifikan dalam dekade terakhir, dengan 71% organisasi melaporkan peningkatan kecepatan *delivery project* sebesar 30-50% setelah mengimplementasikan praktik Agile Beck, K., Brown, M., & Davis, S. (2022). Studi ini juga menunjukkan bahwa tim yang menggunakan Agile memiliki tingkat responsivitas yang lebih tinggi terhadap perubahan kebutuhan pelanggan dibandingkan dengan metodologi tradisional seperti Waterfall.

Keberhasilan implementasi Agile sangat bergantung pada faktor-faktor kritis yang telah diidentifikasi dalam penelitian (Martinez & Chen, 2024). Mereka menemukan bahwa komunikasi tim yang efektif dan dukungan manajemen merupakan dua faktor utama yang menentukan kesuksesan adopsi Agile, dengan 65% proyek yang gagal disebabkan oleh kurangnya salah satu atau kedua faktor tersebut. Lebih lanjut, (Thompson et al., 2022) dalam studi longitudinalnya mengungkapkan bahwa organisasi yang

berhasil mengadopsi Agile secara penuh mengalami peningkatan kepuasan pelanggan sebesar 40% dan penurunan biaya pengembangan hingga 25% dalam jangka panjang. Penelitian ini juga menekankan pentingnya transformasi budaya organisasi dalam mendukung praktik Agile yang berkelanjutan.

3.4. *Virtual Host*

Virtual host telah menjadi komponen kritis dalam pengelolaan layanan *website* modern, memungkinkan satu *server* fisik untuk menampung beberapa domain dan aplikasi *website* secara bersamaan. (Anderson et al., 2023) dalam penelitian komprehensifnya mengenai infrastruktur web modern mengungkapkan bahwa implementasi *virtual host* telah meningkatkan efisiensi penggunaan sumber daya server hingga 75% dibandingkan dengan penggunaan *server dedicated* konvensional. Studi ini, yang menganalisis lebih dari 1000 implementasi *web server* di berbagai skala bisnis, juga menunjukkan bahwa penggunaan *virtual host* dapat mengurangi biaya infrastruktur hingga 60% sambil mempertahankan performa dan keamanan yang optimal.

Faktor-faktor kunci dalam keberhasilan implementasi *virtual host* termasuk konfigurasi yang tepat, manajemen sumber daya yang efisien, dan pemantauan performa yang konsisten. Penelitian yang sama juga mengidentifikasi bahwa 82% masalah performa pada implementasi *virtual host* disebabkan oleh konfigurasi yang tidak optimal,

bukan oleh keterbatasan *hardware*. Temuan ini menekankan pentingnya pemahaman mendalam tentang konfigurasi *virtual host* dan praktik terbaik dalam implementasinya untuk memaksimalkan keuntungan yang dapat diperoleh dari teknologi ini.

3.5. GitLab

GitLab telah berkembang menjadi platform kolaborasi dan manajemen kode yang komprehensif dalam ekosistem pengembangan perangkat lunak modern. Menurut penelitian yang dilakukan oleh (Roberts et al., 2023), adopsi GitLab di kalangan perusahaan teknologi telah meningkat sebesar 65% dalam tiga tahun terakhir, dengan 78% organisasi melaporkan peningkatan efisiensi *delivery software* setelah mengimplementasikan GitLab sebagai solusi DevOps terintegrasi mereka. Studi ini menganalisis dampak implementasi GitLab pada lebih dari 500 tim pengembangan dan menemukan bahwa waktu *deployment* rata-rata berkurang hingga 40% dibandingkan dengan penggunaan *tools* yang terfragmentasi.

Keunggulan utama GitLab terletak pada kemampuannya mengintegrasikan berbagai aspek siklus hidup pengembangan perangkat lunak dalam satu platform. Penelitian yang sama juga mengungkapkan bahwa perusahaan yang mengadopsi GitLab sebagai solusi *end-to-end* mengalami pengurangan kompleksitas *toolchain* sebesar 60% dan peningkatan produktivitas tim sebesar 35%. Studi ini lebih lanjut menunjukkan bahwa organisasi yang memanfaatkan fitur CI/CD terintegrasi GitLab mencapai tingkat otomatisasi yang lebih tinggi dalam *pipeline* pengembangan mereka, dengan 85% tim melaporkan peningkatan signifikan dalam *deployment*.

3.6. *Structured Query Language (SQL)*

SQL (*Structured Query Language*) terus memainkan peran vital dalam pengelolaan data di era *big data* dan sistem informasi modern. Penelitian komprehensif yang dilakukan oleh (Johnson et al., 2023) mengungkapkan bahwa meskipun bermunculan berbagai alternatif *database* NoSQL, 73% perusahaan *enterprise* masih mengandalkan SQL sebagai bahasa utama untuk manajemen dan analisis data mereka. Studi ini, yang melibatkan analisis terhadap 1000 perusahaan Fortune 500, menemukan bahwa sistem berbasis SQL menunjukkan tingkat reliabilitas 30% lebih tinggi dalam menangani transaksi kompleks dibandingkan alternatif NoSQL.

Lebih lanjut, (Anderson & Zhang, 2022) dalam studi longitudinal mereka mengidentifikasi evolusi penggunaan SQL dalam konteks modern, terutama dalam integrasi dengan teknologi *big data* dan *cloud computing*. Penelitian mereka menunjukkan peningkatan 45% dalam adopsi SQL untuk analisis data *real-time* dan data *warehouse* selama lima tahun terakhir. Temuan penting lainnya mengindikasikan bahwa organisasi yang mengoptimalkan penggunaan SQL dalam infrastruktur data mereka mengalami peningkatan kinerja *query* sebesar 60% dan pengurangan biaya pemeliharaan *database* hingga 40%. Studi ini juga menekankan pentingnya optimasi *query* dan *indexing* dalam meningkatkan performa sistem *database* SQL, dengan

hasil menunjukkan bahwa implementasi *best practices* dalam SQL dapat mengurangi waktu eksekusi *query* hingga 75% pada kasus tertentu.

3.7. Laravel

Laravel merupakan *framework* PHP yang digunakan untuk pengembangan aplikasi web dengan sintaks yang ekspresif dan elegan. *Framework* ini dirancang untuk menyederhanakan berbagai proses pengembangan aplikasi seperti autentikasi, routing, dan manajemen sesi, sehingga pengembang dapat lebih fokus pada logika bisnis aplikasi tanpa harus mengelola tugas-tugas kecil secara manual (Nugraha et al., 2023). Laravel menggunakan konsep *Model-View-Controller* (MVC) yang memungkinkan pemisahan logika aplikasi, tampilan, dan data secara terstruktur. Dengan adanya konsep ini, pengembangan aplikasi menjadi lebih terorganisir dan mudah untuk dikelola.

Laravel juga menyediakan berbagai fitur modern seperti ORM (*Object Relational Mapping*) melalui *Eloquent* yang memudahkan interaksi dengan *database*, serta *Blade templating engine* yang memungkinkan pengembang membuat tampilan dengan sintaks sederhana dan dinamis (Nugraha et al., 2023). Dalam hal keamanan, Laravel memiliki fitur proteksi terhadap serangan seperti *SQL Injection*, *Cross-Site Scripting* (XSS), dan *Cross-Site Request Forgery* (CSRF) secara bawaan, sehingga aplikasi yang dikembangkan menjadi

lebih aman.

Selain itu, Laravel mendukung pengembangan aplikasi full-stack dengan menggunakan PHP sebagai bahasa utama dan Livewire sebagai library untuk interaksi *real-time* pada sisi front-end. Integrasi dengan *framework* JavaScript seperti Vue.js dan React juga didukung melalui Inertia.js, yang memungkinkan pengembangan aplikasi monolitik dengan pengalaman pengguna yang responsif (Nugraha et al., 2023). Dengan fleksibilitas tersebut, Laravel juga dapat digunakan sebagai back-end API untuk aplikasi berbasis NextJS, React Native, atau *framework* front-end lainnya.

Pengujian aplikasi merupakan aspek penting dalam pengembangan perangkat lunak. Laravel menyediakan dukungan pengujian bawaan dengan PHPUnit, yang memungkinkan pengembang untuk membuat dan menjalankan tes unit secara otomatis. Selain itu, Laravel juga menyediakan fitur pengujian fungsional melalui metode HTTP, yang membantu memastikan bahwa aplikasi berjalan sesuai dengan yang diharapkan (Rocha, 2021).

Dengan dukungan komunitas yang luas, dokumentasi yang lengkap, dan berbagai fitur modern, Laravel menjadi salah satu *framework* PHP yang paling populer dalam pengembangan aplikasi web. Meskipun memiliki kekurangan seperti performa yang lebih lambat dalam menangani request dengan jumlah besar, keunggulan-keunggulan yang ditawarkan oleh Laravel membuatnya

menjadi pilihan utama bagi banyak pengembang dalam membangun aplikasi web modern (Rocha, 2021).

3.8. Docker

Docker telah merevolusi cara pengembangan, pengiriman, dan pengelolaan aplikasi dalam ekosistem teknologi modern. Menurut penelitian mendalam yang dilakukan oleh (Chen et al., 2023), adopsi Docker di perusahaan teknologi telah meningkat secara dramatis, dengan 82% organisasi *enterprise* melaporkan penggunaan kontainerisasi dalam *pipeline deployment* mereka. Studi ini menganalisis implementasi Docker di 300 perusahaan dan menemukan pengurangan waktu *deployment* sebesar 65% serta peningkatan efisiensi penggunaan sumber daya infrastruktur hingga 50% dibandingkan dengan metode tradisional.

Transformasi ini lebih lanjut didukung oleh temuan (Martinez dan Thompson, 2022) yang menunjukkan bahwa organisasi yang mengadopsi Docker secara penuh mengalami peningkatan signifikan dalam skalabilitas dan portabilitas aplikasi mereka. Penelitian mereka mengungkapkan bahwa tim pengembangan yang menggunakan Docker dapat mengurangi waktu *setup environment development* hingga 75% dan menurunkan biaya infrastruktur sebesar 40%. Temuan penting lainnya dari penelitian ini menyoroti bahwa 89% masalah "*works on my machine*" dapat dieliminasi melalui implementasi Docker yang tepat, menghasilkan peningkatan produktivitas tim secara keseluruhan. Penggunaan Docker juga terbukti meningkatkan konsistensi *environment* dari *development* hingga *production*, dengan tingkat keberhasilan *deployment*

mencapai 95% pada organisasi yang menerapkan praktik kontainerisasi secara menyeluruh.

3.9. Repositori Kode

Repositori kode merupakan sebuah sistem yang diusung untuk mendistribusikan aplikasi secara dinamis pada level terendah komponennya dan menyimpan informasinya. Repositori berfungsi sebagai pusat pengelolaan data dari berbagai versi kode yang dimanfaatkan untuk mengembangkan aplikasi. Penggunaan repositori kode membuat penggunaannya mampu bekerja secara kolaboratif dan mampu melacak perubahan yang terjadi pada kode. Hal ini membuat pengguna dapat kembali ke titik tertentu ketika versi tertentu ingin dikembalikan sehingga dapat mencegah kehilangan data dalam usaha menjaga integritas kode. Repositori kode juga dapat dimanfaatkan bersamaan dengan *Continuous Integration/Continuous Delivery* (CI/CD) sehingga memungkinkan pengguna untuk melakukan otomatisasi build, uji, dan *deploy* aplikasi

[Halaman ini sengaja dikosongkan]

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem

Bagian ini menjelaskan sistem dalam pembangunan sistem UTBK dari alur sistem yang akan dibangun. Hal ini akan dijelaskan menjadi dua bagian yaitu definisi umum sistem.

4.1.1 Definisi Umum Aplikasi

Secara umum, website Edubrand berperan sebagai *client* dari aplikasi UTBK (Ujian Tulis Berbasis Komputer) yang berfungsi sebagai platform simulasi pengerjaan UTBK bagi para siswa di setiap sekolah. Website ini memiliki tiga aktor utama, yaitu admin, sekolah, dan siswa, yang masing-masing memiliki akses dan fitur yang berbeda sesuai dengan kebutuhan mereka.

Admin memiliki wewenang untuk mengelola master data, mengatur penjadwalan ujian yang akan dibuka, serta melakukan pengaturan lainnya yang tidak dapat diakses oleh sekolah maupun siswa. Sekolah memiliki kemampuan untuk mengelola data siswanya, termasuk mereset ujian siswa jika diperlukan. Sementara itu, siswa dapat menggunakan *website* ini untuk mengerjakan simulasi UTBK sesuai dengan jadwal yang telah ditentukan.

4.1.2 Analisis Kebutuhan

Dalam *website* ini, terdapat fungsi-fungsi yang harus dipenuhi oleh sistem. Kebutuhan ini terbagi ke dalam dua jenis, yakni kebutuhan fungsional dan kebutuhan non-fungsional. Berikut adalah deskripsi masing-masing kebutuhan *website* Edubrand sesuai dengan *request* atau permintaan *client* Edubrand untuk menambah fitur atau memperbaiki fitur yang ada.

1. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang berdasarkan proses yang mampu disediakan oleh sistem dan mencakup kebutuhan dasar pengguna tersebut berupa fitur, layanan, dan fungsi. Kebutuhan fungsional dari *website client* Edubrand bagian Admin akan dijelaskan pada Tabel 4.1.

Tabel 4.1 Kebutuhan Fungsional Sistem Sekolah

Kode Kebutuhan	Deskripsi Kebutuhan
F-001	Masuk ke dalam <i>website</i>
F-002	<i>View Dashboard</i> untuk Monitoring
F-003	CRUD pada Siswa
F-004	CRUD pada <i>Manage Test</i> (Bank Soal, Survey Soal)
F-005	CRUD pada <i>Tryout</i>
F-006	CRUD pada Penjadwalan <i>Tryout</i>
F-007	CRUD pada Master Data

Selanjutnya akan dijelaskan mengenai kebutuhan fungsional pada bagian Sekolah pada Tabel 4.2.

Tabel 4.2 Kebutuhan Fungsional Sistem Sekolah

Kode Kebutuhan	Deskripsi Kebutuhan
F-001	Masuk ke dalam <i>website</i>
F-002	<i>View Dashboard</i> untuk Monitoring Siswa
F-003	CRUD pada Siswa/Peserta
F-004	CRUD pada Status Siswa
F-005	<i>View Assesment Report</i> Siswa

Selanjutnya akan dijelaskan mengenai kebutuhan fungsional pada bagian Siswa pada Tabel 4.3.

Tabel 4.3 Kebutuhan Fungsional Sistem Siswa

Kode Kebutuhan	Deskripsi Kebutuhan
F-001	Masuk ke dalam <i>website</i>
F-002	<i>View Dashboard</i> Siswa
F-003	Siswa dapat Memilih Kampus/Jurusan
F-004	Siswa dapat mengerjakan ujian dan dapat melihat hasil ujian

2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan sekumpulan batasan, karakteristik, dan properti pada sistem, baik dalam pengembangan maupun operasional. Kebutuhan non-fungsional dari *website* Edubrand terdapat pada Tabel 4.4.

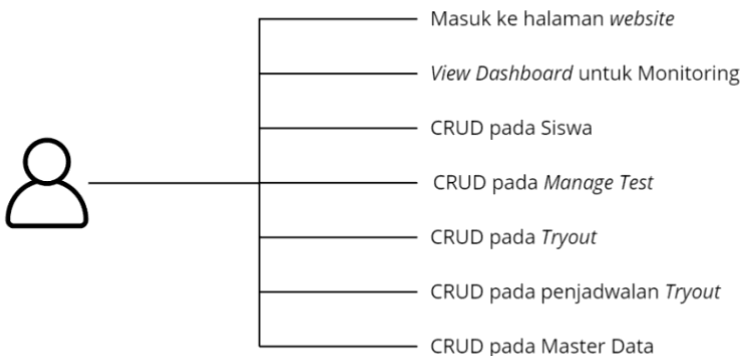
Tabel 4.4 Kebutuhan Non Fungsional Aplikasi

Kode Kebutuhan	Deskripsi Kebutuhan
F-001	Masuk ke dalam <i>website</i>
F-002	<i>View Dashboard</i> untuk

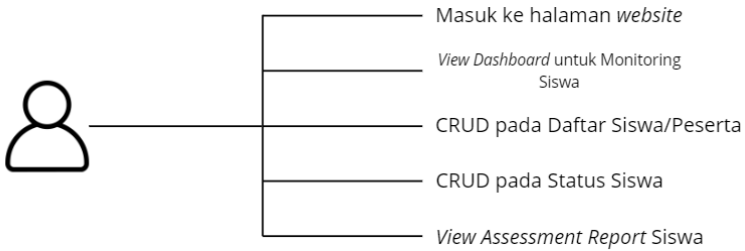
	Monitoring
F-003	CRUD pada Siswa
F-004	CRUD pada <i>Manage Test</i> (Bank Soal, Survey Soal)
F-005	CRUD pada <i>Tryout</i>
F-006	CRUD pada Penjadwalan <i>Tryout</i>
F-007	CRUD pada Master Data

4.1.3 Diagram Use Case

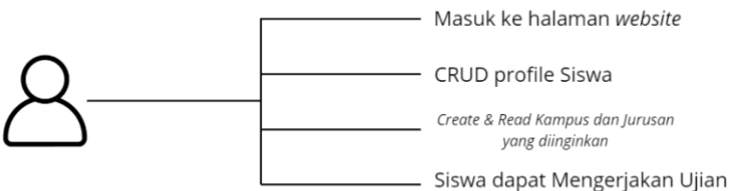
Kebutuhan fungsional pengguna digambarkan dalam bentuk *Use Case Diagram*, yakni diagram yang menggambarkan hubungan antara sistem dan aktornya, serta fungsi dan cakupan tingkat suatu sistem. Diagram *use case* dari *website* UTBK dapat dilihat pada gambar Gambar 4.1, Gambar 4.2, dan Gambar 4.3.



Gambar 4.1 Diagram Use Case Admin



Gambar 4.2 Diagram *Use Case* Sekolah



Gambar 4.3 Diagram *Use Case* Siswa

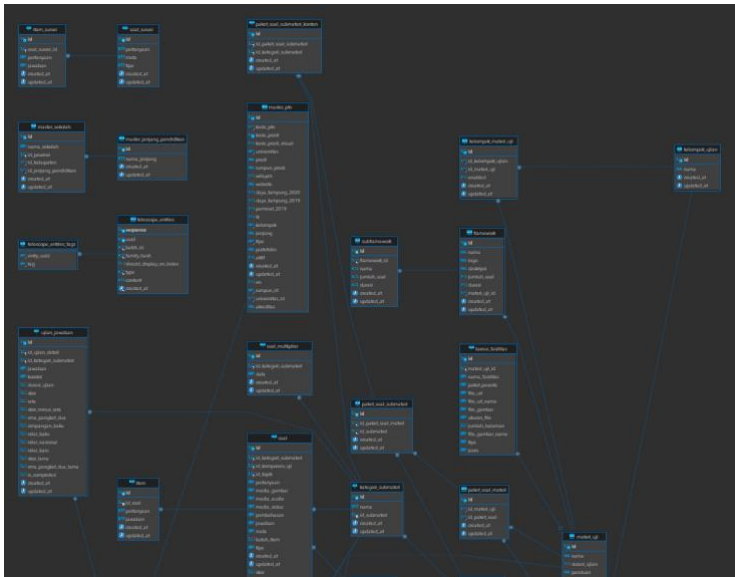
4.2 Perancangan Sistem

Perancangan sistem yang digunakan adalah ERD atau *Entity Relationship Diagram* sebagai berikut.

4.2.1 Entity Relationship Diagram

Entity Relationship Diagram (ERD) merupakan model konseptual yang menggambarkan hubungan antar entitas dalam suatu sistem basis data. Dalam pengembangan sistem informasi, ERD menjadi alat yang penting untuk memvisualisasikan struktur data dan hubungan antar data sebelum diimplementasikan ke dalam basis data fisik.

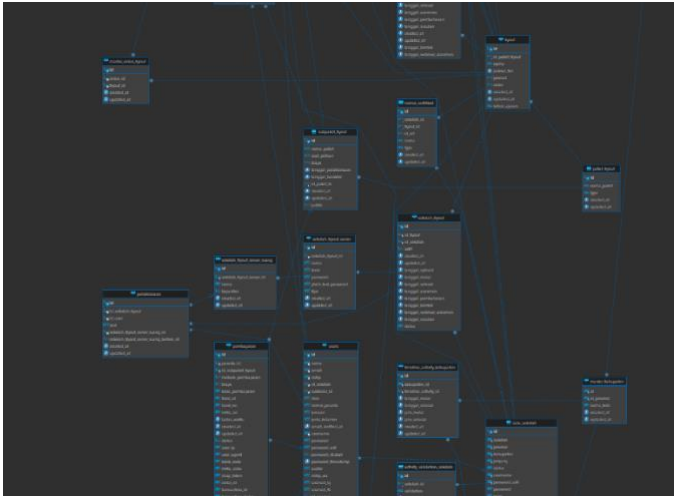
ERD terdiri dari tiga komponen utama: entitas, atribut, dan relasi. Entitas merepresentasikan objek atau konsep yang datanya akan disimpan, atribut adalah karakteristik yang mendeskripsikan entitas, sedangkan relasi menunjukkan hubungan antara dua atau lebih entitas. Penggunaan ERD dalam fase perancangan sistem dapat mengurangi kesalahan desain *database* hingga 40% dan meningkatkan efisiensi pengembangan sistem. Berikut adalah contoh implementasi ERD pada Gambar 4.4 – Gambar 4.7.



Gambar 4.4 Sistem ERD pada aplikasi UTBK (Bagian 1)



Gambar 4.5 Sistem ERD pada aplikasi UTBK (Bagian 2)



Gambar 4.6 Sistem ERD pada aplikasi UTBK (Bagian 3)

4.3 Infrastruktur Website

Sistem UTBK ini dibangun dengan arsitektur terdistribusi yang terdiri dari beberapa layer utama, dimana setiap *layer* memiliki peran dan teknologi spesifik. Implementasi multi-*frontend framework* dipilih untuk mengoptimalkan pengalaman pengguna sesuai dengan kebutuhan masing-masing aktor.

1. Client Layer

- a. Panel Admin dan Sekolah menggunakan kombinasi Laravel Blade dan Bootstrap untuk manajemen konten dan *monitoring*. Pemilihan Laravel Blade memungkinkan integrasi yang lebih erat dengan *backend* Laravel dan memudahkan rendering *server-side*.
- b. Panel Siswa menggunakan Vue.js sebagai *Single Page Application* (SPA) untuk memberikan pengalaman ujian yang lebih responsif dan interaktif. Pemilihan Vue.js untuk sisi siswa didasarkan pada kebutuhan interaktivitas yang tinggi selama proses ujian berlangsung.

2. Server Layer

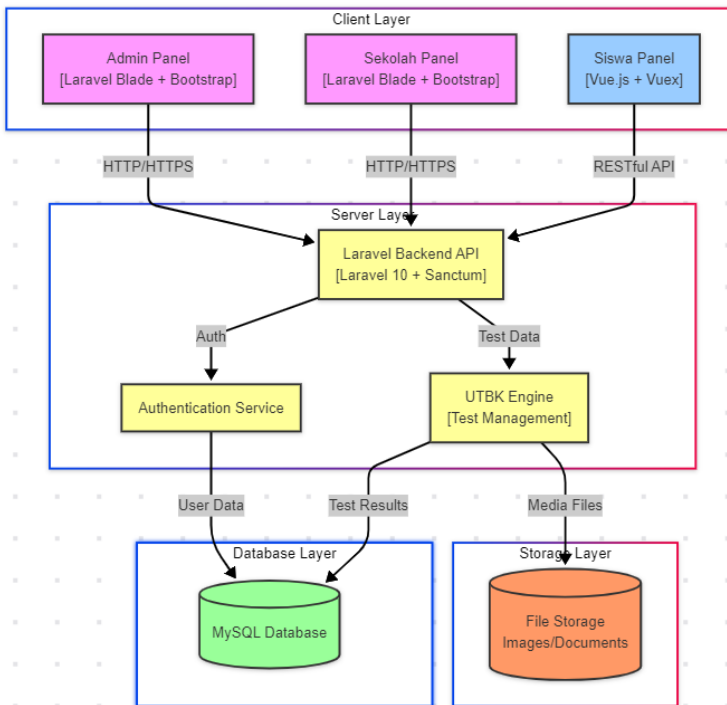
- a. Menggunakan Laravel sebagai backend framework utama yang menangani seluruh *business logic* dan API *endpoints*.
- b. *Authentication Service* mengelola otentikasi *multi-role* menggunakan Laravel Sanctum.
- c. UTBK Engine berfungsi sebagai *core system* yang menangani logika ujian,

termasuk pengacakan soal, dan manajemen waktu.

3. Storage Layer

a. MySQL digunakan sebagai *database* utama untuk menyimpan data pengguna, bank soal, dan hasil ujian.

b. File Storage diimplementasikan untuk menyimpan *file-file* pendukung seperti



Gambar 4.9 Infrastruktur aplikasi Web

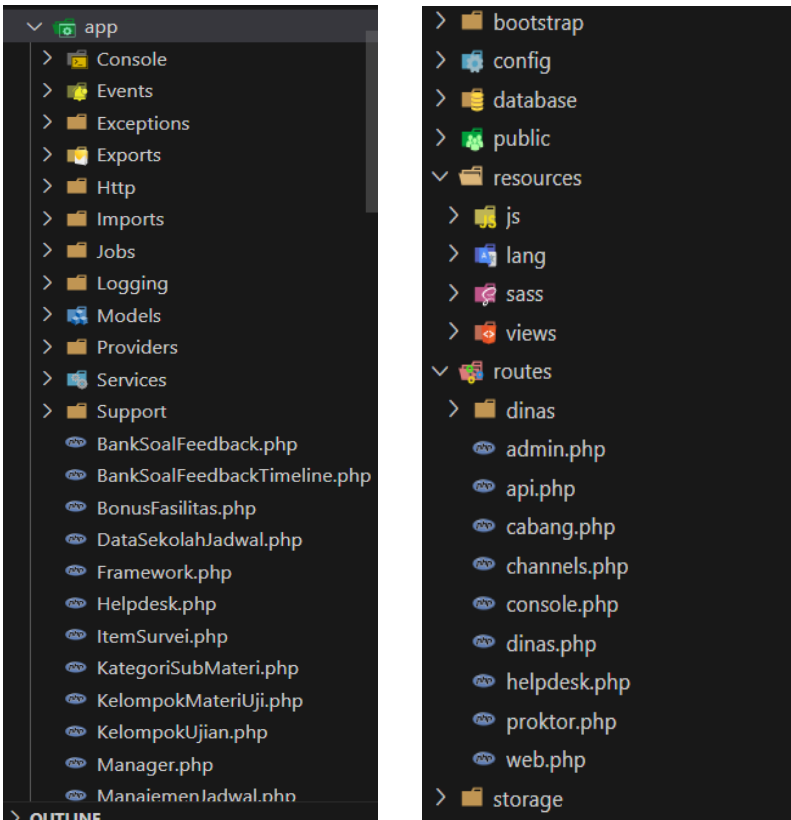
gambar soal dan dokumen hasil ujian.

[Halaman ini sengaja dikosongkan]

BAB V IMPLEMENTASI SISTEM

5.1. Implementasi Kerangka Kode Sumber

Dengan menggunakan Laravel sebagai *framework* dan laravel blade serta mengimplementasikan metode agile dalam pengembangan *software house* ini. Berikut merupakan kerangka kerja yang digunakan



Gambar 5.1 Struktur folder kerangka kode

Direktori folder secara *default* akan mengikuti saat inisialisasi Laravel dan Laravel Blade sebagai *frontend* nya serta tidak banyak kustomisasi yang dilakukan. Berikut adalah penjelasan pada setiap folder kode sumber yang digunakan.

1. Folder **app** berisi folder utama untuk keperluan *backend* seperti yang dijelaskan pada folder berikut
 - **Http** digunakan request http dari pengguna berisi *controllers*, *middlewares*, dan *requests*.
 - **Events** digunakan untuk sinkronisasi data secara *asynchronous*.
 - **Exports** yang digunakan untuk mengelola proses ekspor data, misalnya ke dalam format CSV, Excel, atau PDF. Lalu ada
 - **Imports** yang digunakan untuk menangani proses impor data, seperti menggugah file CSV ke dalam database.
 - **Logging** digunakan untuk mencatat aktivitas aplikasi.
 - **Models** berisi model yang berhubungan langsung dengan database dan digunakan sebagai representasi data dalam aplikasi.
 - **Providers** digunakan untuk mendaftarkan layanan dalam laravel.
 - **Services** digunakan untuk menyimpan *service logic* atau *class helper* yang tidak masuk ke dalam kategori lain.
 - **Support** berisi helper atau fungsi

- pendukung yang digunakan dalam berbagai bagian aplikasi.
2. Folder **bootstrap** berisi file bootstrap Laravel yang digunakan untuk mem-bootstrap *framework* dan memuat konfigurasi dasar sebelum aplikasi berjalan.
 3. Folder **config** berisi berbagai file konfigurasi laravel, seperti *database.php* untuk konfigurasi database, *mail.php* untuk pengaturan email, *queue.php* untuk antrian tugas, dan lainnya.
 4. Folder **database** yang digunakan untuk mengelola database sebagai berikut
 - **Migrations** File untuk mengelola perubahan struktur tabel database.
 - **Seeders** Digunakan untuk memasukkan data *dummy* atau awal ke dalam database.
 - **Factories** Membantu dalam pembuatan data *dummy* menggunakan model *factory*.
 5. Folder **public** berisi dapat diakses oleh publik
 6. Folder **resources** digunakan untuk menyimpan *view*, file statis, dan komponen *frontend*.
 7. Folder **routes** digunakan untuk mendefinisikan rute aplikasi, termasuk API dan web.
 8. Folder **storage** berfungsi untuk menyimpan log, cache, dan file yang di-upload.
 9. Folder **tests** berisi pengujian otomatis untuk Laravel.
 10. Folder **vendor** berisi *package dependency Laravel* melalui Composer.

5.2. Implementasi Kerangka Kode Sumber

Dalam pengembangan *frontend* aplikasi, diperlukan integrasi dengan *backend* agar data dan proses yang dilakukan pengguna dapat tersimpan dalam database. Implementasi koneksi antara *frontend* dan *backend* dapat ditemukan pada sumber kode 5.1.

```

@section('content')
  @include('sweetalert::alert')

  <section class="content pt-4">
    <div class="container-fluid">
      <div class="card card-default">
        <div class="card-header bg-primary">
          <div class="row mb-2">
            <div class="col-sm-6">
              <h4>Sekolah</h4>
            </div>
          </div>
        </div>
        <div class="card-body">
          @if (Session::has('status'))
            <div class="alert alert-{{ session('status') }}" role="alert">{{
session('message') }}</div>
          @endif
          @if ($errors->any())
            @foreach ($errors->all() as $error)
              <div class="alert alert-danger" role="alert">{{ $error
}}</div>
            @endforeach
          @endif
          <div class="row">
            <div class="col-md-8">
              <form role="form" action="{{ route('penjadwalan.store')
}}" method="post"
              enctype="multipart/form-data">
                {{ csrf_field() }}
                <div class="card-body">
                  <div class="form-group">
                    <label for="tryout" class="text-black-50">Pilih
Simulasi</label>
                    <select name="tryout" id="tryout" class="form-
control select2">
                      @foreach ($tryout as $result)
                        <option value="{{ $result->id }}">{{ $result-
>nama }}</option>
                      @endforeach
                    </select>
                  </div>
                  <div class="form-group">
                    <label for="kabupaten" class="text-black-50">Pilih
Kabupaten</label>
                    <select name="kabupaten" id="kabupaten"
class="form-control select2">
                      @foreach ($kabupaten as $result)
                        <option value="{{ $result->id }}">{{ $result-

```

```

>nama_kota }}</option>
    @endforeach
  </select>
</div>
<div class="form-group">
  <label for="exampleInputEmail1" class="text-black-50">Tanggal Mulai</label>
  <div class="input-group date" id="tanggal_mulai" data-target-input="nearest">
    <input type="datetime-local" name="tanggal_mulai" class="form-control datetimepicker-input" />
    <div class="input-group-append" data-target="#tanggal_mulai" data-toggle="datetimepicker">
      <div class="input-group-text"><i class="far fa-clock"></i></div>
    </div>
  </div>
</div>
<div class="form-group">
  <label for="exampleInputEmail1" class="text-black-50">Tanggal Selesai</label>
  <div class="input-group date" id="tanggal_selesai" data-target-input="nearest">
    <input type="datetime-local" name="tanggal_selesai" class="form-control datetimepicker-input" />
    <div class="input-group-append" data-target="#tanggal_selesai" data-toggle="datetimepicker">
      <div class="input-group-text"><i class="far fa-clock"></i></div>
    </div>
  </div>
</div>
<div class="form-group">
  <label for="exampleInputEmail1" class="text-black-50">Tanggal Upload</label>
  <div class="input-group date" id="tanggal_upload" data-target-input="nearest">
    <input type="datetime-local" name="tanggal_upload" class="form-control datetimepicker-input" />
    <div class="input-group-append" data-target="#tanggal_upload" data-toggle="datetimepicker">
      <div class="input-group-text"><i class="far fa-clock"></i></div>
    </div>
  </div>
</div>

```

```

        </div>
    </div>
</div>
<div class="form-group">
    <label for="exampleInputEmail1" class="text-black-50">Tanggal Asesmen</label>
    <div class="input-group date" id="tanggal_asesmen" data-target-input="nearest">
        <input type="datetime-local" name="tanggal_asesmen" class="form-control datetimepicker-input" />
        <div class="input-group-append" data-target="#tanggal_asesmen" data-toggle="datetimepicker">
            <div class="input-group-text"><i class="far fa-clock"></i></div>
        </div>
    </div>
</div>
</div>
<div class="form-group">
    <label for="exampleInputEmail1" class="text-black-50">Tanggal Pembahasan</label>
    <div class="input-group date" id="tanggal_pembahasan" data-target-input="nearest">
        <input type="datetime-local" name="tanggal_pembahasan" class="form-control datetimepicker-input" />
        <div class="input-group-append" data-target="#tanggal_pembahasan" data-toggle="datetimepicker">
            <div class="input-group-text"><i class="far fa-clock"></i></div>
        </div>
    </div>
</div>
</div>
<div class="form-group">
    <label for="exampleInputEmail1" class="text-black-50">Tanggal Susulan</label>
    <div class="input-group date" id="tanggal_susulan" data-target-input="nearest">
        <input type="datetime-local" name="tanggal_susulan" class="form-control datetimepicker-input" />
        <div class="input-group-append" data-target="#tanggal_susulan" data-toggle="datetimepicker">
            <div class="input-group-text"><i class="far fa-clock"></i></div>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
</div>
</div>
<div class="card-footer">
    <button type="submit" class="btn btn-
primary">Submit</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</section>
@endsection

```

Kode Sumber 5.1 Implementasi Frontend dengan Form Data

Kode Sumber 5.1 membuat *form* yang digunakan untuk pengiriman data ke dalam *backend* melalui *Route*. Kasus *route* disini adalah *penjadwalan.index* yang merupakan peningkatan dari *penjadwalan/{id}/create*. Data yang diinputkan dari *frontend* akan diterima pada sisi *backend* dan akan dimasukkan ke dalam *database*. Implementasi *backend* akan dijelaskan pada kode sumber 5.2.


```

public function store(Request $request)
{
    $request->validate([
        'tryout' => 'required',
        'kabupaten' => 'required',
        'tanggal_upload' => 'required',
        'tanggal_mulai' => 'required',
        'tanggal_selesai' => 'required',
        'tanggal_asesmen' => 'required',
        'tanggal_pembahasan' => 'required',
        'tanggal_bimtek' => 'nullable',
        'tanggal_webinar_asesmen' => 'nullable',
        'tanggal_susulan' => 'nullable',
    ]);

    Penjadwalan::create([
        'id_tryout' => $request->tryout,
        'id_kabupaten' => $request->kabupaten,
        'tanggal_upload' => $request->tanggal_upload,
        'tanggal_mulai' => $request->tanggal_mulai,
        'tanggal_selesai' => $request->tanggal_selesai,
        'tanggal_asesmen' => $request->tanggal_asesmen,
        'tanggal_pembahasan' => $request->tanggal_pembahasan,
        'tanggal_bimtek' => @$request->tanggal_bimtek,
        'tanggal_webinar_asesmen' => @$request-
>tanggal_webinar_asesmen,
        'tanggal_susulan' => @$request->tanggal_susulan,
    ]);

    $penjadwalan = Penjadwalan::where('id_tryout', $request->tryout)-
>where('id_kabupaten', $request->kabupaten)->first();
    $kabupaten = Kabupaten::find($penjadwalan->id_kabupaten);
    $listSekolah = DB::table('sekolah_tryout')
        ->join('data_sekolah', 'sekolah_tryout.id_sekolah', 'data_sekolah.id')
        ->where('sekolah_tryout.id_tryout', $penjadwalan->id_tryout)
        ->where('data_sekolah.kabupaten', $kabupaten->nama_kota)
        ->select(
            'data_sekolah.id as id_sekolah',
            'data_sekolah.sekolah as nama_sekolah',
            'sekolah_tryout.*',
        )
        ->get();

    foreach ($listSekolah as $sekolah) {
        SekolahTryout::updateOrCreate(
            [
                'id_tryout' => $request->tryout,
                'id_sekolah' => $sekolah->id_sekolah,
            ],

```

```

    [
        'tanggal_upload' => @$penjadwalan->tanggal_upload,
        'tanggal_mulai' => @$penjadwalan->tanggal_mulai,
        'tanggal_selesai' => @$penjadwalan->tanggal_selesai,
        'tanggal_asesmen' => @$penjadwalan->tanggal_asesmen,
        'tanggal_pembahasan' => @$penjadwalan->tanggal_pembahasan,
        'tanggal_bimtek' => @$penjadwalan->tanggal_bimtek,
        'tanggal_webinar_asesmen' => @$penjadwalan->tanggal_webinar_asesmen,
        'tanggal_susunan' => @$penjadwalan->tanggal_susunan,
        'status' => 'umum'
    ]
);
}

return redirect()->route('penjadwalan.index')->with(['status' =>
'success', 'message' => 'Save Successfully']);
}

```

Kode Sumber 5.2 Implementasi Backend pada Store

Pemanggilan *backend* dilakukan dengan fungsi *store* dalam controller Laravel. Fungsi *store* tersebut telah didefinisikan dalam routes sehingga bisa langsung diakses melalui *frontend*. Fungsi *store* tersebut digunakan untuk menyimpan penjadwalan yang dilakukan oleh admin dan dikirim dari *frontend* ke database dengan melewati validasi input terlebih dahulu. Ketika input yang diberikan user telah sesuai dengan format yang ada pada *backend*, maka data tersebut akan masuk ke dalam database. Jika ada validasi yang salah, maka akan melakukan *return* sesuai dengan *error* atau kesalahan yang ada.

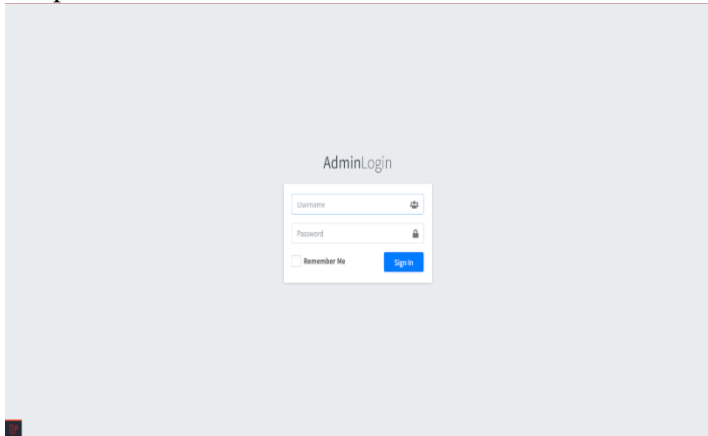
5.3. Implementasi Antarmuka Pengguna

Berikut merupakan antarmuka yang telah diimplementasikan,

1. Login Admin

Halaman login admin merupakan gerbang awal untuk akses ke sistem UTBK bagi

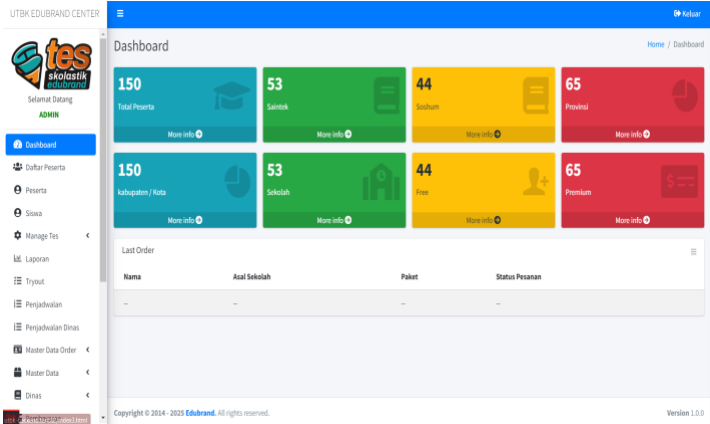
administrator. Halaman ini menampilkan form login dengan field username dan password yang dilengkapi dengan validasi input dan dapat dilihat pada Gambar 5.2.



Gambar 5.2 Halaman Login Admin

2. Dashboard Admin

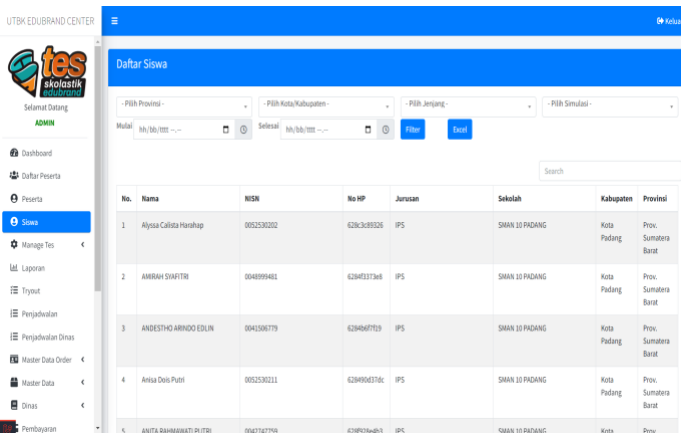
Dashboard admin menampilkan ringkasan informasi penting seperti jumlah total siswa, jumlah ujian aktif, dan total provinsi. Terdapat *cards* yang memvisualisasikan data-data penting untuk monitoring sistem secara *real-time* dan dapat dilihat pada Gambar 5.3.



Gambar 5.3 Halaman Dashboard Admin

3. Dashboard Siswa

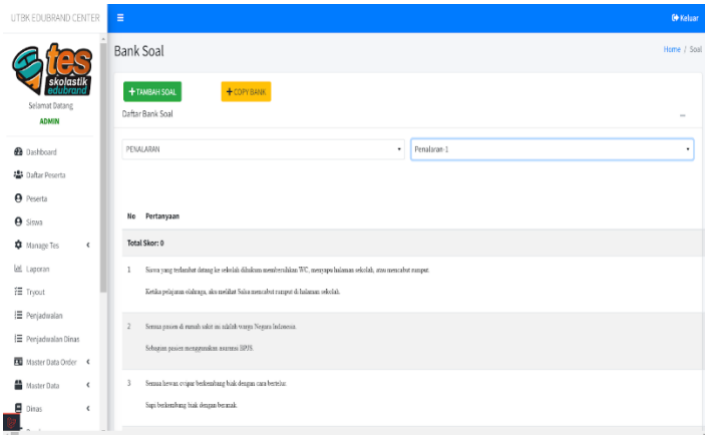
Halaman manajemen siswa menyediakan interface untuk pengelolaan data siswa, termasuk fitur tambah, edit, dan hapus data siswa. Dilengkapi dengan fitur pencarian, filter, dan pagination untuk memudahkan pengelolaan data dalam jumlah besar dan dapat dilihat pada Gambar 5.4.



Gambar 5.4 Halaman Dashboard Siswa

4. Dashboard *Management Test*

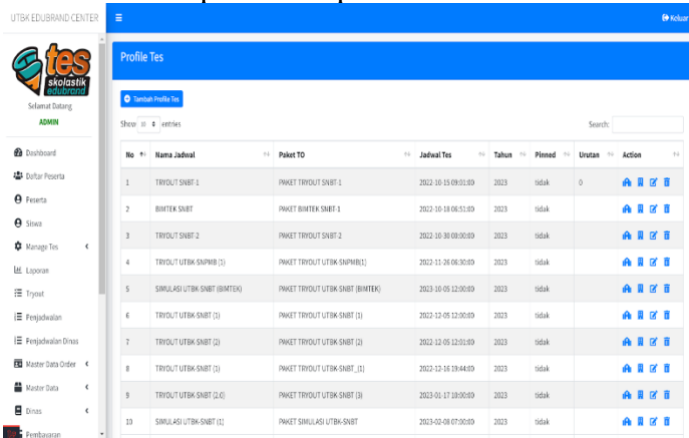
Halaman ini berfungsi untuk mengelola bank soal dan paket ujian. Admin dapat membuat, mengedit, dan mengatur soal-soal ujian. Terdapat fitur import soal dari file Excel dan preview soal sebelum dipublish dan dapat dilihat pada Gambar 5.5.



Gambar 5.5 Halaman Manajemen Test

5. Dashboard *Tryout*

Interface khusus untuk mengelola sesi tryout, termasuk pengaturan waktu, jumlah peserta, dan jenis soal. Admin dapat memantau progress tryout dan mengakses hasil secara real-time dan dapat dilihat pada Gambar 5.6.



Gambar 5.6 Halaman Manajemen *Tryout*

6. Dashboard Penjadwalan

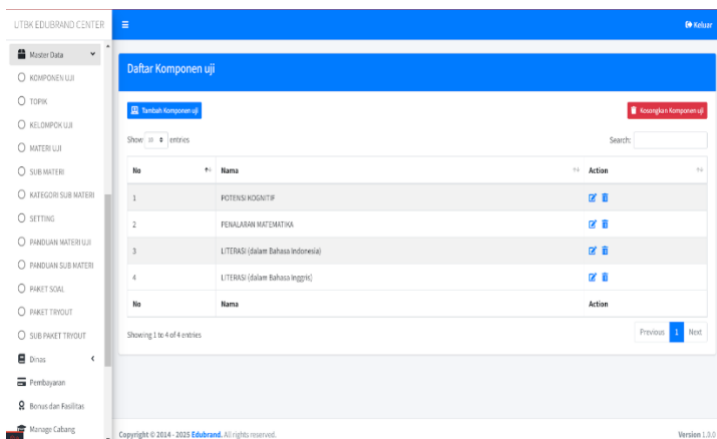
Sistem penjadwalan ujian yang komprehensif, memungkinkan admin mengatur jadwal ujian untuk masing-masing sekolah dan dapat dilihat pada Gambar 5.7.

No	Nama Try Out	Kabupaten	Tanggal Mulai	Tanggal Selesai	Tanggal Upload	Tanggal Asesmen	Tanggal Pembahasan	Tanggal Sesiuan	Action
1	TRYOUT SBNBT 1	Kota Padang	2022-10-27 07:30:00	2022-10-28 18:30:00	2022-10-22 07:30:00	2024-03-21 06:00:00	2024-03-21 06:00:00	2024-03-21 05:24:31	
2	TRYOUT SBNBT 1	Kab. Kotawaringin Timur	2022-11-06 07:30:00	2022-11-10 18:30:00	2022-11-03 07:30:00	2024-03-21 06:00:00	2024-03-21 06:00:00	2024-03-21 05:27:29	
3	TRYOUT SBNBT 1	Kota Padang Raya	2022-11-22 07:30:00	2022-11-22 18:30:00	2022-11-03 07:30:00	2024-03-21 06:00:00	2024-03-21 06:00:00		
4	TRYOUT SBNBT 1	Kab. Agam	2022-11-14 07:30:00	2022-11-15 18:30:00	2022-10-23 10:00:00	2024-03-21 06:00:00	2024-03-21 06:00:00		
5	TRYOUT SBNBT 1	Kab. Padang Panjang	2022-10-27 07:30:00	2022-10-31 18:30:00	2022-10-22 12:00:00	2024-03-21 06:00:00	2024-03-21 06:00:00		
6	TRYOUT SBNBT 1	Kab. Padang Pesisir	2022-11-06 07:30:00	2022-11-10 18:30:00	2022-11-03 08:00:00	2024-03-21 06:00:00	2024-03-21 06:00:00		
7	TRYOUT SBNBT 1	Kab. Kotawaringin Barat	2022-11-24 07:30:00	2022-11-24 18:30:00	2022-11-03 08:00:00	2024-03-21 06:00:00	2024-03-21 06:00:00		
8	TRYOUT SBNBT 1	Kab. Humbang Raya	2022-11-06 07:30:00	2022-11-10 18:30:00	2022-11-03 08:00:00	2024-03-21 06:00:00	2024-03-21 06:00:00		

Gambar 5.7 Halaman Manajemen Penjadwalan

7. Dashboard Master Data

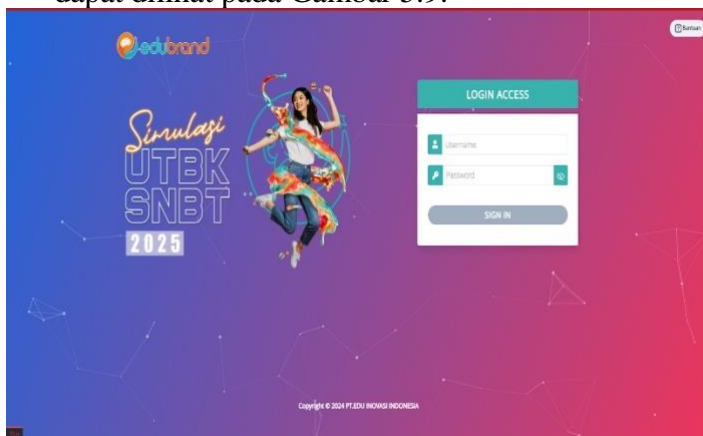
Pusat pengelolaan data master seperti data sekolah, mata pelajaran, ruang ujian, dan pengaturan sistem lainnya. Interface ini menyediakan CRUD operations dengan validasi data yang ketat dan dapat dilihat pada Gambar 5.8.



Gambar 5.8 Halaman Manajemen Master Data

8. Login Sekolah

Halaman login khusus untuk akun sekolah dengan tampilan yang user-friendly dan dapat dilihat pada Gambar 5.9.

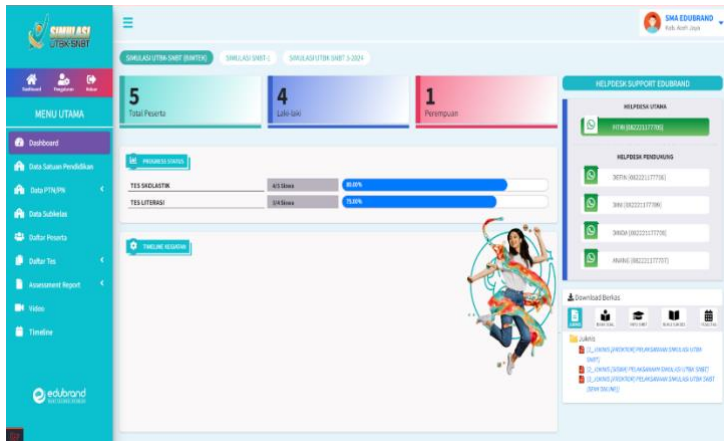


Gambar 5.9 Halaman Login Sekolah

9. Dashboard Sekolah

Tampilan utama untuk pihak sekolah

yang menampilkan informasi relevan seperti jadwal ujian, statistik kehadiran siswa, dan pengumuman penting. dan dapat dilihat pada Gambar 5.10.



Gambar 5.10 Halaman Dashboard Sekolah

10. Dashboard Daftar Tes dan Peserta

Halaman yang menampilkan daftar lengkap tes yang tersedia untuk sekolah beserta peserta yang terdaftar dan dapat dilihat pada Gambar 5.11.

Daftar Peserta Simulasi (SIMULASI UTBK-SNBT (BIMTEK))

Tampilkan Peserta Koreksi Simulasi

Show 8 entries Search:

No	Nama Peserta	NISN	Jenis Kelamin	Username	Password	Server/Ruang	Sei	Action
1	SISWA-1	0942010027	Laki-laki	308F4790006	admin	SERVER-1-R-1	1	
2	SISWA-2	0942010029	Perempuan	308F4790007	admin	SERVER-1-R-1	1	
3	DEWI AYU	094201011	Perempuan	308F4790008	admin	SERVER-1-R-1	1	
4	SISWA-4	0942010028	Laki-laki	308F4790009	admin	SERVER-1-R-1	1	
5	MULTARA	0942010032	Perempuan	308F4790010	admin	SERVER-1-R-1	1	
6	SISWA-6	11122233	Laki-laki	308F4790011	admin	SERVER-1-R-1	1	
7	Siswa testing	121213121221	Laki-laki	308F4790013	208474	SERVER-2-R-1	1	
8	Siswa testing-2	12121312871	Laki-laki	308F4790014	117287	SERVER-1-R-1	1	

Showing 1 to 8 of 8 entries Previous Next

Gambar 5.11 Halaman Dashboard Daftar Tes Sekolah dan Peserta

11. Dashboard Daftar Status Peserta

Interface monitoring status peserta ujian secara real-time, menampilkan informasi seperti kehadiran, progress pengerjaan, dan status submit ujian. dan dapat dilihat pada Gambar 5.12.

Status Peserta Simulasi [SIMULASI UTBK-SNBT (BIMTEK)]

Status:

Selesai Ujian 3 Siswa (37,50%)
 Sedang Mengajar 6 Siswa (6,00%)
 Progress 2 Siswa (25,00%)
 Belum Ujian 3 Siswa (37,50%)

Show | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | Search:

No	Nama Peserta	NISN	Status			Aksi
			Status	Skolastik	Liberal	
1	SISWA-1	0042070027	Belum Ujian	X	X	
2	SISWA-2	0042070028	Belum Ujian	X	X	
3	DEWI ANI	004207011	Belum Ujian	X	X	
4	SISWA-4	0042070026	Selesai Ujian	✓	✓	Reset *
5	MUTIARA	0042070032	Selesai Ujian	✓	✓	Reset *
6	SISWA-6	11122233	Progress	X	X	Reset *
7	Siswa-9089G	12121212221	Selesai Ujian	✓	✓	Reset *
8	Elissa Nurrahma *	11111111111	Progress	✓	✓	Reset *

Gambar 5.12 Halaman Status Peserta

12. Dashboard View Assessment Report

Halaman yang menyajikan laporan komprehensif hasil assessment siswa, termasuk analisis jawaban, statistik performa, dan rekomendasi perbaikan dan dapat dilihat pada Gambar 5.13.

SMAN 1 ASRI UTBK-SNBT SMA EDUBRAND Kalsi, Kalimantan Tengah

Home / Duga Semp

Rekapitulasi
Peringkat
Rapor Smpg
Analisis Siswa
Sertifikat
Statistik
Data Pilihan Siswa
Soal & Pembahasan
Summary
Progress
Print

TES SKOLASTIK

Kemampuan Penalaran Umum

Keselesaian Pemupataan	X	0 Siswa Tuntas / 5 Siswa	<input type="text" value="0"/>	Detail
Simpulan Logis	X	0 Siswa Tuntas / 5 Siswa	<input type="text" value="0"/>	Detail
Penalaran Analitis	X	0 Siswa Tuntas / 5 Siswa	<input type="text" value="0"/>	Detail
Penalaran Kuantitatif	X	0 Siswa Tuntas / 5 Siswa	<input type="text" value="0"/>	Detail
Bahasa Pinda & Kriptografi	X	0 Siswa Tuntas / 5 Siswa	<input type="text" value="0"/>	Detail

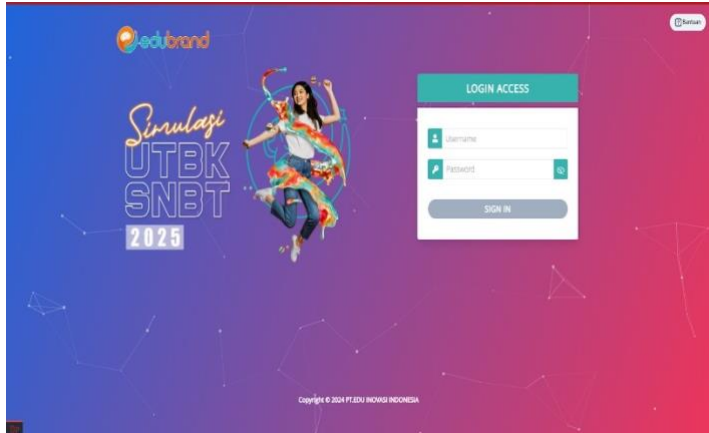
Penghargaan dan Penalaran Umum

Mahira Kita/Praja	X	0 Siswa Tuntas / 5 Siswa	<input type="text" value="0"/>	Detail
Persempamaan	X	0 Siswa Tuntas / 5 Siswa	<input type="text" value="0"/>	Detail

Gambar 5.13 Halaman View Assessment Report

13. Login Siswa

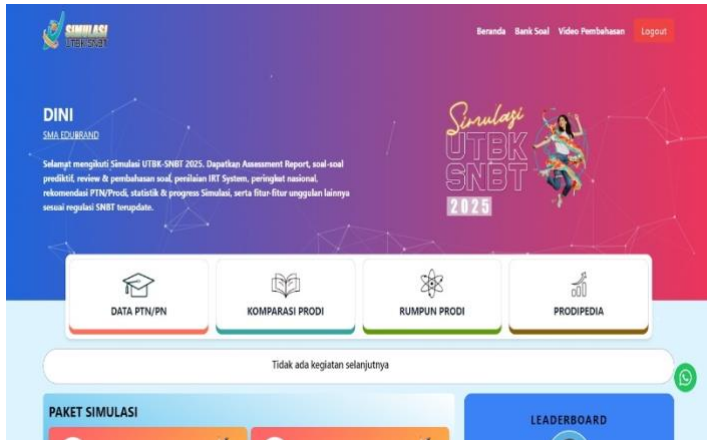
Interface login khusus siswa dengan desain minimalis dan fokus pada kemudahan penggunaan dan dapat dilihat pada Gambar 5.14.



Gambar 5.14 Halaman Login Siswa

14. Manajemen Profile Siswa

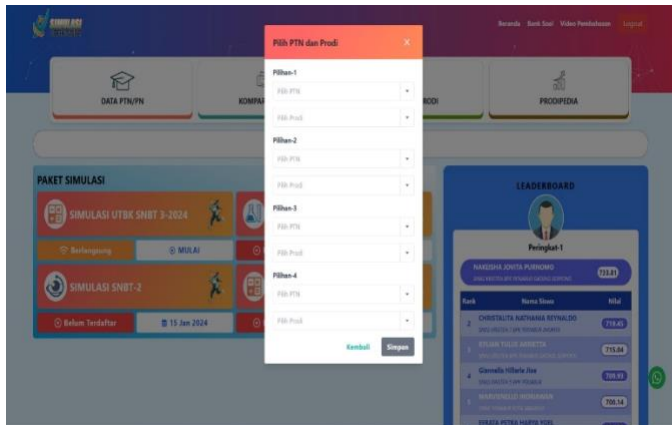
Interface tampilan awal siswa ketika memasuki aplikasi dan dapat melihat langsung yang dibuka dan pemeringkatan dan dapat dilihat pada Gambar 5.15.



Gambar 5.15 Halaman Dashboard Awal Siswa

15. Pemilihan Kampus dan Jurusan

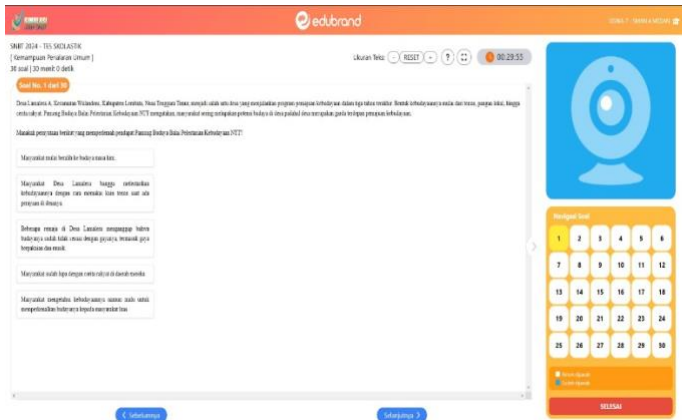
Interface yang memudahkan siswa memilih dan mengelola preferensi kampus dan jurusan yang diminati dan dapat dilihat pada Gambar 5.16.



Gambar 5.16 Halaman Pemilihan Kampus dan Jurusan

16. Pengerjaan Soal

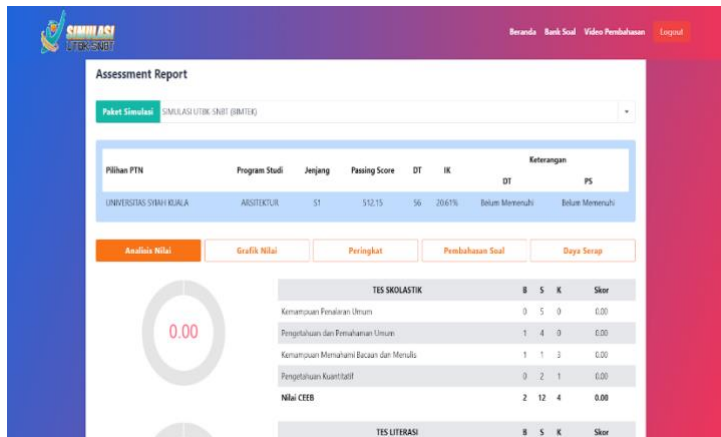
Tampilan utama untuk mengerjakan ujian dengan fitur-fitur seperti timer, navigasi soal, mark for review, dan preview jawaban. Interface dirancang untuk meminimalkan distraksi dan memaksimalkan fokus siswa dan dapat dilihat pada Gambar 5.17.



Gambar 5.17 Halaman Pengerjaan Soal

17. Hasil Pengerjaan Soal

Halaman yang menampilkan hasil ujian secara detail, termasuk skor, analisis jawaban, dan rekomendasi untuk perbaikan dan dapat dilihat pada Gambar 5.18.



Gambar 5.18 Halaman Hasil Pengerjaan oleh Siswa

[Halaman ini sengaja dikosongkan]

BAB VI

PENGUJIAN DAN EVALUASI

6.1. Tujuan Pengujian

Pengujian dilakukan terhadap sistem UTBK untuk memastikan kesesuaian dan ketepatan fungsionalitas dari seluruh fitur sistem serta memenuhi kebutuhan Edubrand sesuai dengan yang diharapkan. Pengujian mencakup aspek fungsional, keamanan, dan performa sistem.

6.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut :

1. Keberhasilan autentikasi dan otorisasi untuk setiap role (admin, sekolah, siswa),
2. Validasi proses manajemen ujian (pembuatan, penjadwalan, pelaksanaan),
3. Keakuratan sistem penilaian dan pelaporan hasil ujian,
4. Kemampuan sistem dalam menangani ujian secara konkuren,
5. Keamanan data dan pencegahan kecurangan selama ujian,
6. Performa sistem saat diakses oleh banyak pengguna,
7. Integritas data pada setiap transaksi dalam sistem.

6.3. Skenario Pengujian

Skenario pengujian dilakukan dengan melakukan serangkaian test case untuk memvalidasi setiap komponen sistem UTBK:

1. Pengujian Autentikasi dan Otorisasi
 - Login untuk setiap role pengguna
 - Verifikasi pembatasan akses berdasarkan role
 - Validasi *session management*
2. Pengujian Manajemen Ujian
 - Pembuatan dan pengelolaan bank soal
 - Penjadwalan ujian
 - Pengaturan Waktu dan durasi ujian
 - Manajemen peserta ujian
3. Pengujian Pelaksanaan Ujian
 - Proses pengerjaan soal
 - Sistem timer dan auto-submit
 - Fitur Navigasi soal
 - Sistem anti-kecurangan
4. Pengujian Penilaian dan Pelaporan
 - Akurasi perhitungan nilai
 - *Generate* laporan hasil
 - *Export* data dalam berbagai format
 - Analisis statistik hasil ujian
5. Pengujian Performa
 - Load testing untuk akses konkuren
 - *Response time* monitoring
 - Databas *performance* testing

6.4. Evaluasi Pengujian

Hasil pengujian yang dilakukan terhadap perlakuan pada skenario pengujian sistem UTBK terangkum pada tabel di bawah ini,

Tabel 5.1. Hasil Pengujian Sistem Pengguna

Kriteria Pengujian	Hasil Pengujian
Sistem dapat melakukan Autentikasi Multi-role	Terpenuhi
Sistem dapat melakukan Manajemen Bank Soal	Terpenuhi
Sistem dapat melakukan Pelaksanaan Ujian	Terpenuhi
Sistem dapat melakukan Penilaian otomatis	Terpenuhi
Sistem mampu melakukan Pelaporan Hasil	Terpenuhi
Sistem memiliki Keamanan Data	Terpenuhi
Sistem dapat menangani <i>Concurrent</i> pengguna	Terpenuhi

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kerja praktik di PT Aksamedia Mulia Digital memberikan wawasan yang luas mengenai pengembangan perangkat lunak menggunakan metodologi Agile. Dengan mengikuti proses pengembangan yang fleksibel dan iteratif, kami belajar bagaimana sebuah software house beradaptasi dengan perubahan kebutuhan client secara cepat dan efisien. Metode Agile memungkinkan tim pengembang untuk melakukan perbaikan secara berkala serta meningkatkan komunikasi antar anggota tim, sehingga proyek dapat diselesaikan sesuai dengan ekspektasi client.

Selama kerja praktik, kami mendapatkan pengalaman langsung dalam menangani berbagai tahapan pengembangan perangkat lunak, mulai dari analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian dan evaluasi. Kami juga memahami bagaimana software house mengelola banyak proyek secara bersamaan dengan tetap menjaga kualitas hasil pengembangan. Proses ini melibatkan koordinasi yang baik antar tim serta penggunaan teknologi yang tepat untuk memastikan kelancaran pengembangan.

Selain aspek teknis, kerja praktik ini juga

memberikan manfaat dalam meningkatkan kemampuan kerja sama tim, problem-solving, dan komunikasi profesional di lingkungan kerja. Kami juga belajar bagaimana menghadapi tantangan dalam pengembangan perangkat lunak, seperti perubahan kebutuhan client yang dinamis dan keterbatasan waktu dalam pengerjaan proyek. Dengan pengalaman yang diperoleh, kami lebih siap untuk menghadapi dunia kerja di industri teknologi.

7.2. Saran

Perancangan sistem metodologi agile di PT Aksamedia Mulia Digital dapat ditingkatkan di kemudian hari dengan,,

- a. Dokumentasi yang lebih terstruktur dan lengkap agar tim dapat memahami proyek dengan lebih baik serta memudahkan transisi antar sprint,
- b. Menambahkan agenda pelatihan singkat mengenai workflow Agile dan teknologi yang digunakan,
- c. Menerapkan evaluasi berkala terhadap penerapan agile dengan melakukan retrospektif secara berkala, agar tim dapat menemukan cara yang lebih efektif dalam mengelola proyek dan meningkatkan kualitas pengembangan perangkat lunak.

DAFTAR PUSTAKA

- Alt Rainer and Auth, G. and K. C. (2019). Transformation of Consulting for Software-Defined Businesses: Lessons from a DevOps Case Study in a German IT Company. In V. Nissen (Ed.), *Advances in Consulting Research: Recent Findings and Practical Cases* (pp. 385–403). Springer International Publishing. https://doi.org/10.1007/978-3-319-95999-3_19
- Carey, J. (2008). Role misconceptions and negotiations in small business owner/web developer relationships. *Journal of Management & Organization*, 14, 85–99. <https://doi.org/10.5172/jmo.2008.14.1.85>
- Kamaludin, A. (2023). Development of Computer Based Test (CBT) Website-based Chemistry National Exam for Students of Islamic High School (MA). *Jurnal Penelitian Pendidikan IPA*, 9(6), 4164–4171. <https://doi.org/10.29303/jppipa.v9i6.1540>
- Laato Samuli and Mäntymäki, M. and B. T. and I. A. K. M. N. and H. S. (2021). Digital Transformation of Software Development: Implications for the Future of Work. In A. and P. N. and D. Y. K. and P. I. and M. M. Dennehy Denis and Griva (Ed.), *Responsible AI and Analytics for an Ethical and Inclusive Digitized Society* (pp. 609–621). Springer International Publishing.
- Masrohatin, S., Puji Astuti, R., Iqbal Fardian, M., Arif Handiwibowo, G., & Islam Negeri Kiai Haji Achmad Siddiq, U. (2022). The Growth of Industrial Revolution 4.0 and the Preparation of Software Engineers in Digital Transformation at Makers Institute Indonesia. *Institute Indonesia. Journal of Research and Technology*, 8(2), 303–312.
- Nugraha, M., Agus, R., Fathi, H., & Rizki Baginda, M. (2023). DEVELOPMENT A WEB-BASED STUDENT INTERNSHIP APPLICATION USING LARAVEL FRAMEWORK & WATERFALL MODEL. In *JOURNAL OF INFORMATION TECHNOLOGY AND ITS UTILIZATION* (Vol. 6).
- Rocha, A. (2021). *2021 16th Iberian Conference on Information Systems and Technologies (CISTI) : proceedings of CISTI'2021 -*

- 16th Iberian Conference on Information Systems and Technologies : 23 to 26 of June 2021, Chaves, Portugal*. IEEE.
- Senja Sis Ha Abadi, P., Astuti, I., & Enawaty, E. (2024). Student Perceptions Regarding the Use of a Computer-Based Test (CBT) System at State Junior High School 3 Sungai Betung. *Journal of Educational Learning and Innovation*, 4(1), 120.
<https://doi.org/10.46229/elia.V4i1>
- Sosnowski, J., Fox, A., Garcia, D., Moosvi, F., Silva, M., West, M., & Zilles, C. (2025). Experiences with Computer-Based Testing (CBT). *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2*, 1728.
<https://doi.org/10.1145/3641555.3705092>
- Ulum, M. N., & Hidayat, S. (2024). Rancang Bangun Computer Based Test (CBT) Berbasis Web. In *Jurnal Indonesia : Manajemen Informatika dan Komunikasi (JIMIK)* (Vol. 5, Issue 2). <https://journal.stmiki.ac.id>
- Yanti, & Siswandi. (2024). USE OF INTRANET FOR CBT-BASED ASSESSMENT (COMPUTER BASED TEST) AT SENIOR HIGH SCHOOL. *International Conference on Aplied Social Sciences in Education*, 1(1), 403–410.
<https://doi.org/10.31316/icasse.v1i1.6937>
- Beck, K., Brown, M., & Davis, S. (2022). The state of agile: A global survey of adoption and effectiveness. *IEEE Software Engineering Review*, 15(3), 78–92.
- Martinez, J., & Chen, L. (2023). Critical success factors in agile implementation. *Journal of Software Development Practice*, 8(2), 145–160.
- Thompson, R., Wilson, K., & Lee, A. (2021). *Long-term impact of agile methodology on software development organizations*. *International Journal of Project Management*, 39(4), 234–249.
- Anderson, J., Roberts, P., & Chen, L. (2023). *Virtual hosting in modern web infrastructure: Performance analysis and best practices*. *Journal of Web Engineering and Architecture*, 28(4), 156–172.
- Roberts, M., Anderson, K., & Lee, S. (2023). *GitLab adoption trends and impact on software delivery performance*. *Journal of*

- DevOps Practice, 16(2), 89–104.
- Johnson, R., & Wilson, P. (2023). SQL in enterprise data management: A comprehensive analysis. *Journal of Database Management*, 34(2), 112–128.
- Anderson, M., & Zhang, L. (2022). Evolution of SQL in modern data architecture. *International Journal of Data Engineering*, 15(4), 234–249.
- Chen, H., Wilson, K., & Lee, S. (2023). Docker adoption patterns and infrastructure optimization in enterprise environment. *Journal of Cloud Computing Architecture*, 18(3), 156–171.
- Martinez, R., & Thompson, P. (2022). Impact of Docker containerization on development workflow and resource utilization. *International Journal of DevOps Practice*, 12(4), 234–249.

[Halaman ini sengaja dikosongkan

BIODATA PENULIS

Nama : M Naufal Badruttamam
Tempat, Tanggal Lahir : Surabaya, 15 Maret 2003
Jenis Kelamin : Laki – Laki
Telepon : +62895378107054
Email : 5025211240@student.its.ac.id

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC, ITS
Angkatan : 2021
Semester : 7 (Tujuh)

