

## BAB III

### PELAKSANAAN KERJA

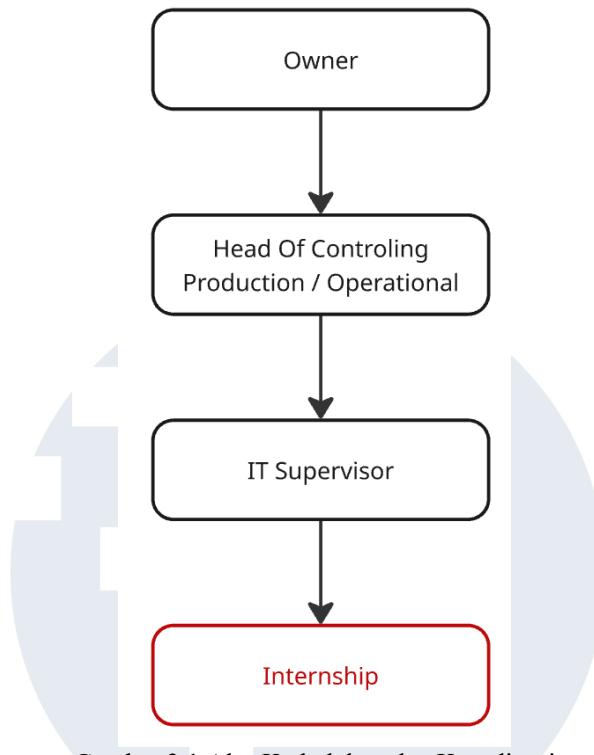
#### 3.1 Kedudukan dan Koordinasi

Pada pelaksanaan magang, penulis diberikan posisi dibawah supervisor dari *controlling of production / operation*, maka dari itu penulis bisa dengan langsung berkomunikasi dan diberikan arahan serta pembelajaran secara terstruktur dan terperinci. Sebelumnya dalam kertas penulis diterima di PT Doremi Digital Solusindo, tetapi Ketika pelaksanaan kerja magang dilaksanakan, penulis dipindahkan ke induk perusahaan Markaz Group, dikarenakan kebutuhan pada saat itu lebih dipentingkan ke Markaz Group di bandingkan dengan PT Doremi Digital Solusindo.

##### 3.1.1 Kedudukan

Penulis dalam struktur organisasi bersifat fleksibel, tidak terbatas pada satu departemen tertentu. Meskipun sebagian besar pekerjaan berkaitan dengan pengembangan sistem berbasis web dan dukungan teknis IT, penulis juga terlibat dalam kegiatan lintas-divisi apabila dibutuhkan oleh perusahaan. Pola koordinasi dilakukan secara berjenjang, dimulai dari Supervisor kemudian kepada Head of Production untuk evaluasi lanjutan.

Untuk mendukung kelancaran pekerjaan, perusahaan menerapkan alur komunikasi dan koordinasi yang tertata, mulai dari briefing rutin, pembagian tugas harian, diskusi teknis, hingga pelaporan progres mingguan. Secara umum, alur kedudukan dan koordinasi penulis dapat digambarkan melalui bagan berikut:



Gambar 3.1 Alur Kedudukan dan Koordinasi

Bagan tersebut menggambarkan posisi penulis dalam struktur koordinasi pekerjaan, di mana setiap alur komunikasi dan pelaporan dilakukan secara berjenjang sesuai struktur organisasi yang berlaku.

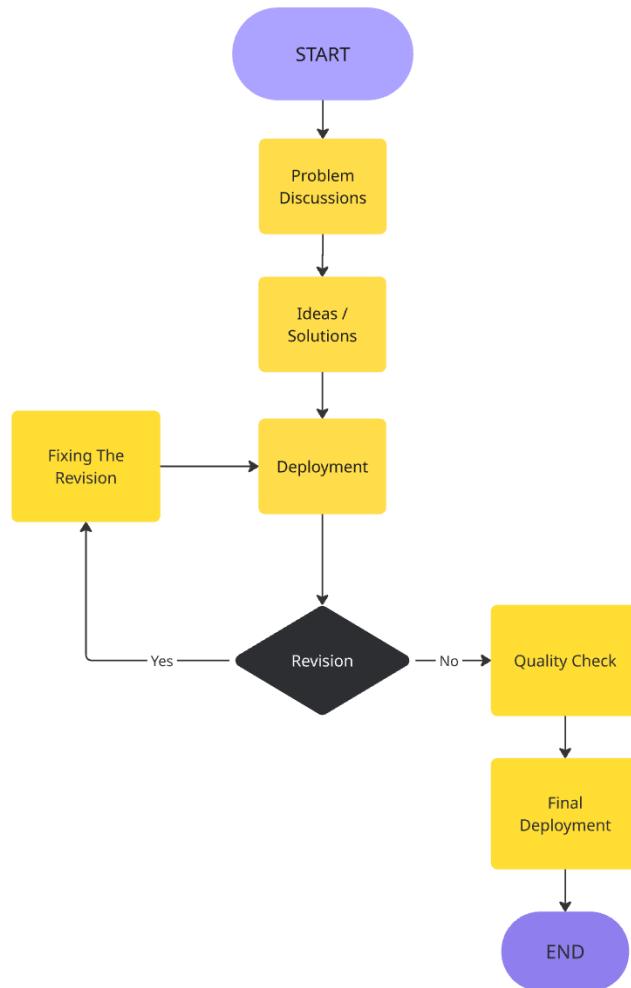
### 3.1.2 Koordinasi

Dalam pelaksanaan kegiatan kerja magang, penulis menempati posisi sebagai Internship di lingkungan kerja Markaz Group. Secara administratif, penulis diterima melalui PT Doremi Digital Solusindo sebagai unit usaha yang membawahi program magang, namun aktivitas kerja harian dan seluruh sistem koordinasi dilakukan langsung di kantor Markaz Group sebagai perusahaan induk.

Selama masa magang, penulis berada di bawah pembinaan dan pengawasan *Head of Production* pada *controlling production/operational*, yang bertanggung jawab dalam memberikan arahan umum terkait proyek, serta memastikan setiap tugas berjalan sesuai kebutuhan operasional perusahaan. Dalam pelaksanaan teknis, penulis berkoordinasi secara rutin dengan

Supervisor, yang berperan sebagai pembimbing langsung dalam penyusunan tugas, penyelesaian masalah, dan pengecekan hasil pekerjaan.

Berikut adalah Alur kerja kegiatan magang penulis :



Gambar 3.2 Flowchart Alur Kerja Magang

### 3.2 Tugas yang Dilakukan

Selama masa pelaksanaan kerja magang, penulis mendapatkan berbagai tugas yang berhubungan langsung dengan proses pengembangan aplikasi *MarkazTest*, yaitu sistem ujian online berbasis web yang digunakan oleh perusahaan untuk mendukung proses rekrutmen internal dan untuk monitoring SDM karyawan perusahaan. Tugas-tugas tersebut meliputi pemahaman kebutuhan sistem, analisis struktur aplikasi, pengembangan fitur baru, perbaikan bug, serta pemeliharaan server yang dilakukan sesuai dengan prosedur perusahaan.

Penulis bertanggung jawab penuh atas pengembangan teknis aplikasi ujian online MarkazTest. Pekerjaan mencakup perancangan *database*, implementasi antarmuka (*frontend*) menggunakan React + Vite, serta pembangunan logika server (*backend*) dengan Node.js dan serta *database* menggunakan Supabase dan Prisma ORM.

Untuk memberikan gambaran yang lebih jelas, seluruh aktivitas kerja penulis dijabarkan dalam bentuk tabel mingguan berikut.

Table 3. 1 Alur Kegiatan Magang

Minggu	Fase	Keterangan Aktivitas
Minggu ke-1	Fase 1 – Inisialisasi dan Analisis Awal	<ul style="list-style-type: none"> <li>- Pengenalan lingkungan kerja di Markaz Group dan PT Doremi Digital Solusindo</li> <li>- Pemahaman profil perusahaan dan sistem kerja</li> <li>- Pengenalan sistem internal perusahaan (OtomaX)</li> <li>- Diskusi awal konsep aplikasi ujian online berbasis web</li> </ul>
Minggu ke-2		<ul style="list-style-type: none"> <li>- Analisis referensi aplikasi tes online (tes mengetik dan tes hafalan)</li> <li>- Pengamatan alur kerja sistem existing</li> <li>- Perancangan awal UI/UX aplikasi ujian online</li> </ul>
Minggu ke-3		<ul style="list-style-type: none"> <li>- Implementasi awal backend sistem tes mengetik</li> <li>- Implementasi frontend dasar aplikasi</li> <li>- Pembuatan struktur proyek</li> <li>- Pengembangan admin dashboard awal</li> <li>- Penerapan sistem penilaian otomatis (lulus/gagal)</li> </ul>

Minggu ke-4		<ul style="list-style-type: none"> <li>- Pengembangan fitur lanjutan tes mengetik</li> <li>- Implementasi highlight jawaban benar dan salah</li> <li>- Pembuatan progress bar pengajaran</li> <li>- Implementasi sistem penilaian berbasis baris (row-based scoring)</li> </ul>
Minggu ke-5	Fase 2 – Manajemen Data dan Integrasi Backend	<ul style="list-style-type: none"> <li>- Pengembangan backend upload materi tes</li> <li>- Parsing file Excel sebagai sumber soal</li> <li>- Pembuatan form input peserta</li> <li>- Implementasi dropdown group dan server</li> <li>- Perbaikan tampilan tabel input frontend</li> </ul>
Minggu ke-6		<ul style="list-style-type: none"> <li>- Migrasi backend ke Prisma ORM</li> <li>- Integrasi database dengan sistem</li> <li>- Penyesuaian API frontend dan backend</li> <li>- Perbaikan struktur data hasil tes</li> <li>- Debugging koneksi sistem</li> </ul>
Minggu ke-7		<ul style="list-style-type: none"> <li>- Pengembangan fitur kalender admin dan user</li> <li>- Pengelompokan materi (group dan sub-group)</li> <li>- Perbaikan tampilan UI admin panel</li> <li>- Sinkronisasi data hasil tes</li> </ul>
Minggu ke-8	Fase 3 – Keamanan Sistem dan Deployment	<ul style="list-style-type: none"> <li>- Implementasi sistem login dan signup</li> <li>- Penerapan autentikasi JWT</li> <li>- Pengaturan role-based access control</li> </ul>

		<ul style="list-style-type: none"> <li>- Perbaikan routing aplikasi</li> <li>- Penyempurnaan alur tes mengetik</li> </ul>
Minggu ke-9		<ul style="list-style-type: none"> <li>- Deployment frontend ke Vercel/Netlify</li> <li>- Deployment backend ke Render</li> <li>- Konfigurasi dan perbaikan CORS</li> <li>- Pengujian koneksi frontend dan backend</li> <li>- Debugging error runtime</li> </ul>
Minggu ke-10		<ul style="list-style-type: none"> <li>- Pengembangan fitur manajemen pengguna</li> <li>- Pembuatan laporan admin</li> <li>- Implementasi paginasi data</li> <li>- Peningkatan keamanan dan stabilitas sistem</li> </ul>
Minggu ke-11	Fase 4 – Fitur Lanjutan	<ul style="list-style-type: none"> <li>- Perancangan logika Tes Hafalan</li> <li>- Implementasi sistem multi-sesi tes</li> <li>- Pembaruan struktur database</li> <li>- Penyesuaian endpoint backend</li> </ul>
Minggu ke-12		<ul style="list-style-type: none"> <li>- Pengembangan fitur pengaturan tes (Admin Settings)</li> <li>- Konfigurasi timer ujian</li> <li>- Pengaturan sistem skoring</li> <li>- Implementasi fitur koreksi manual hasil Tes Hafalan</li> </ul>
Minggu ke-13		<ul style="list-style-type: none"> <li>- Restrukturisasi database untuk Test Builder dinamis</li> <li>- Pemisahan tipe tes (Mengetik, Hafalan, Tata Letak)</li> <li>- Sinkronisasi frontend admin</li> </ul>
Minggu ke-14	Fase 5 – Finalisasi dan Serah Terima	<ul style="list-style-type: none"> <li>- Implementasi Tes Tata Letak</li> <li>- Pengembangan algoritma pengacakan soal</li> </ul>

		<ul style="list-style-type: none"> <li>- Implementasi penilaian fleksibel berbasis kata kunci</li> <li>- Pengujian end-to-end seluruh tipe tes</li> </ul>
Minggu ke- 15 - 16		<ul style="list-style-type: none"> <li>- Perbaikan bug sistem</li> <li>- Sinkronisasi zona waktu dan timer</li> <li>- Pengujian sistem secara menyeluruh</li> <li>- Persiapan serah terima aplikasi</li> </ul>
Minggu ke- 17		<ul style="list-style-type: none"> <li>- Penyusunan laporan kerja magang</li> <li>- Penyelesaian administrasi magang</li> </ul>

### 3.3 Uraian Pelaksanaan Kerja

Pelaksanaan kerja magang pada PT Doremi Digital Solusindo yang ditempatkan di Markaz Group difokuskan pada pengembangan aplikasi ujian online berbasis web yang diberi nama MarkazTest. Aplikasi ini dirancang untuk mendukung berbagai jenis evaluasi, seperti Tes Mengetik, Tes Hafalan, dan Tes Tata Letak, yang digunakan sebagai alat seleksi dan pengukuran kemampuan karyawan. Berdasarkan jadwal kegiatan yang telah direncanakan, proses pengembangan sistem dibagi ke dalam beberapa fase utama yang saling berkesinambungan, dimulai dari tahap inisialisasi hingga tahap finalisasi dan serah terima sistem.

#### 3.3.1 Proses Pelaksanaan

##### A. Fase Inisialisasi dan Analisis Awal (Minggu 1–4)

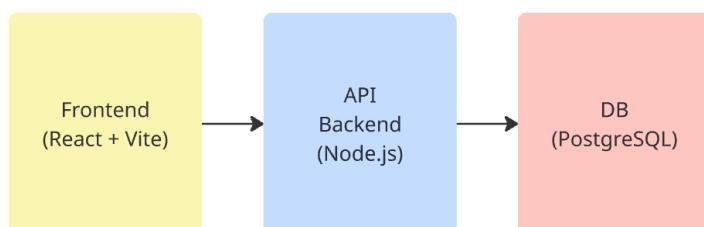
Pada fase awal pelaksanaan kerja magang, penulis difokuskan pada proses inisialisasi sistem serta pemahaman menyeluruh terhadap lingkungan kerja, kebutuhan perusahaan, dan konsep dasar aplikasi ujian online yang akan dikembangkan. Fase ini menjadi tahap yang sangat krusial karena berperan sebagai fondasi utama dalam keseluruhan proses pengembangan sistem MarkazTest, khususnya untuk fitur Tes Mengetik sebagai modul inti aplikasi. Selain itu, penulis juga mendapatkan pengenalan terhadap beberapa sistem internal perusahaan, salah satunya adalah OtomaX, yaitu sistem yang digunakan perusahaan dalam pengelolaan layanan

pulsa dan transaksi digital [3]. Pada tahap ini, penulis diberikan pemahaman mengenai alur bisnis sistem Otomax, cara kerja server, proses transaksi, serta peran sistem tersebut dalam mendukung operasional perusahaan.

Memasuki minggu kedua, fokus kegiatan beralih ke analisis kebutuhan aplikasi ujian online yang akan dikembangkan. Penulis melakukan diskusi awal dengan supervisor terkait tujuan pembuatan aplikasi, target pengguna, serta permasalahan yang ingin diselesaikan oleh sistem. Dari hasil diskusi tersebut, disepakati bahwa aplikasi harus mampu mendukung beberapa jenis tes, dengan Tes Mengetik sebagai modul utama yang membutuhkan akurasi penilaian dan pengalaman pengguna yang baik.

Penulis juga melakukan studi referensi terhadap beberapa aplikasi tes online yang telah ada, baik tes mengetik maupun tes hafalan, untuk memahami pola alur pengerjaan tes, tampilan antarmuka, serta metode penilaian yang digunakan. Hasil observasi ini kemudian dijadikan dasar dalam perancangan awal aplikasi dan logika untuk kemudahan penggunaan oleh para karyawan.

Pada tahap ini, penulis mulai menyusun gambaran arsitektur sistem yang akan digunakan. Aplikasi dirancang menggunakan arsitektur Client–Server yang terpisah, di mana frontend bertugas menangani antarmuka dan interaksi pengguna, sedangkan backend menangani logika bisnis, pengolahan data, dan penyimpanan hasil tes. Alur arsitektur bisa dilihat pada Gambar 3.3

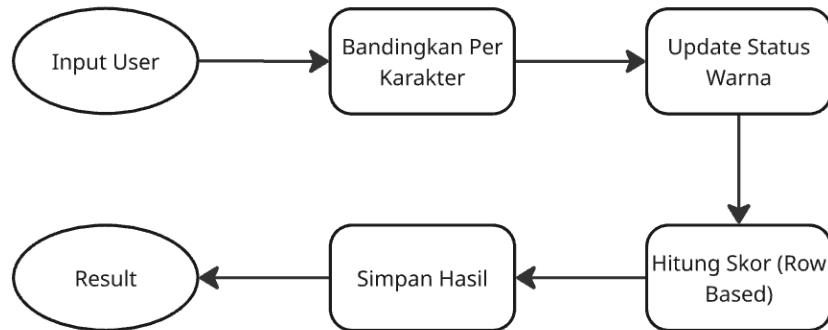


Gambar 3.3 Arsitektur Environmet MarkazTest

Setelah analisis kebutuhan selesai, penulis mulai melakukan setup environment pengembangan. Frontend dikembangkan menggunakan React dengan Vite untuk mendukung performa yang cepat dan modularitas komponen yang memadai, sedangkan backend dibangun menggunakan Node.js. Penulis melakukan instalasi dependency yang diperlukan, pengaturan struktur folder proyek, serta konfigurasi environment lokal agar sesuai dengan kebutuhan server perusahaan. Struktur proyek dirancang dengan pemisahan yang jelas antara frontend dan backend, termasuk pemisahan komponen UI, halaman, service API, serta middleware backend.

Pada minggu keempat, penulis mulai terlibat langsung dalam implementasi fitur inti aplikasi, yaitu Tes Mengetik. Pengembangan difokuskan pada pembuatan logika penilaian otomatis yang mampu menilai jawaban pengguna secara lebih akurat. Sistem penilaian tidak lagi dilakukan per kolom, melainkan menggunakan pendekatan Row-based Scoring, di mana setiap baris jawaban dibandingkan dengan baris referensi.

Untuk meningkatkan pengalaman pengguna, penulis mengembangkan algoritma highlight jawaban, yang memberikan umpan balik visual secara langsung. Karakter yang benar ditandai dengan warna hijau, karakter yang salah dengan warna merah, serta kelebihan input ditandai dengan warna kuning. Selain itu, ditambahkan progress bar yang menunjukkan progres penggerjaan tes berdasarkan jumlah baris yang telah diselesaikan dengan benar. Adapun Struktur Logika Tes Mengetik dapat dilihat pada gambar 3.4

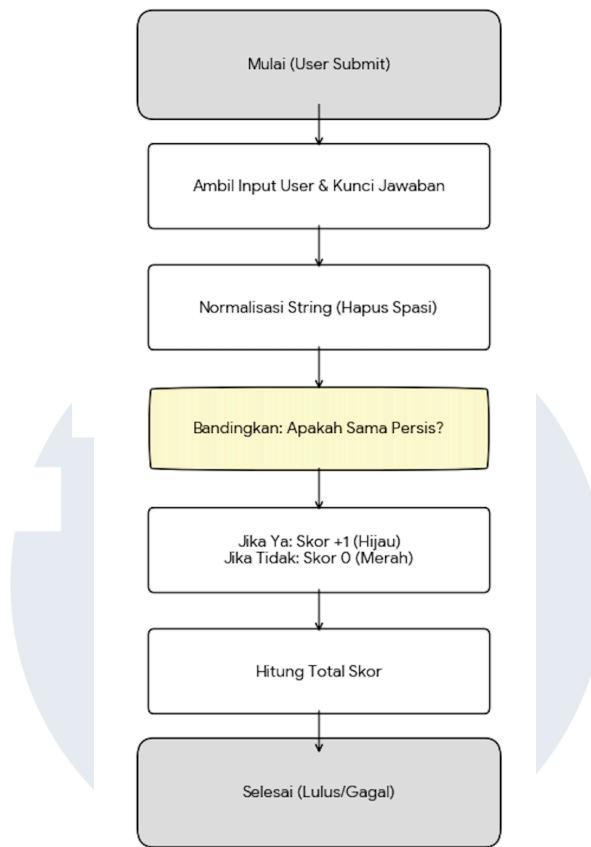


Gambar 3.4 Struktur Logika Tes Mengetik MarkazTest

Untuk memastikan akurasi penilaian, Tes Mengetik pada aplikasi MarkazTest menggunakan metode *row-based scoring*, yaitu sistem penilaian berbasis baris. Pada metode ini, setiap baris input peserta dibandingkan secara langsung dengan baris referensi yang tersimpan di dalam database. Proses komparasi dilakukan menggunakan metode *exact match*, di mana sistem akan memeriksa kesesuaian karakter demi karakter secara menyeluruh.

Jika seluruh karakter pada satu baris input peserta identik dengan baris referensi, maka baris tersebut dinilai benar dan diberikan skor 1. Sebaliknya, apabila terdapat perbedaan satu karakter saja, baik berupa kesalahan ketik, kekurangan huruf, maupun kelebihan karakter, maka baris tersebut dinilai salah dan diberikan skor 0. Pendekatan ini diterapkan untuk menjaga ketelitian tinggi, mengingat data yang diuji bersifat sensitif, seperti kode produk, nomor, dan identitas transaksi digital.

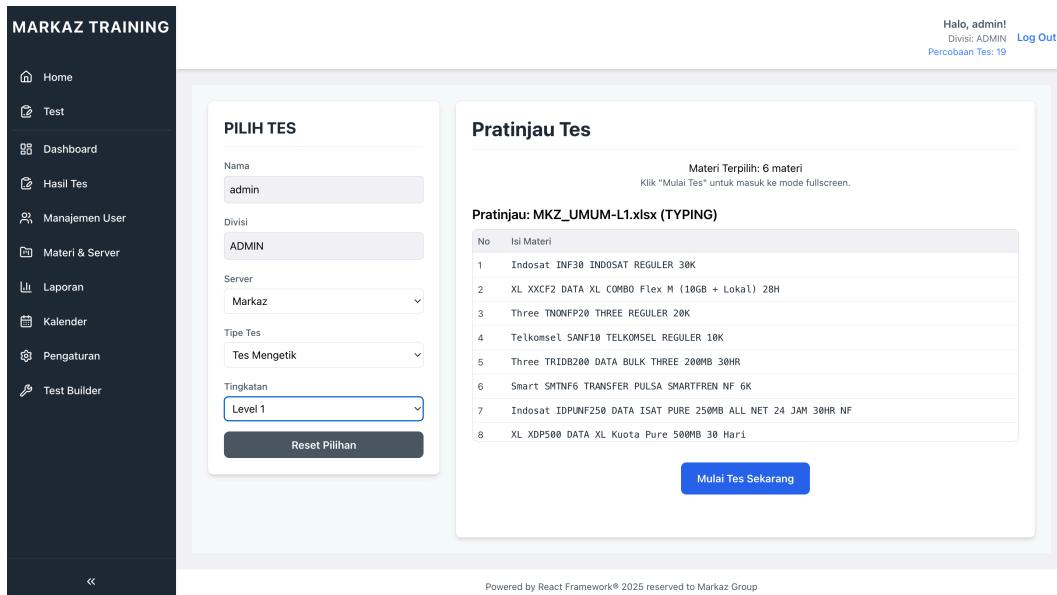
Skor akhir peserta dihitung berdasarkan jumlah total baris yang dijawab dengan benar. Peserta dinyatakan lulus apabila skor total telah mencapai ambang batas minimal yang telah ditentukan oleh admin melalui sistem pengaturan. Pendekatan penilaian ini memberikan standar evaluasi yang objektif, konsisten, serta mencerminkan tingkat ketelitian peserta secara akurat. Berikut adalah diagram alur dari logika tes mengetik :



Gambar 3.5 Diagram Logika Tes Mengetik

Tujuan nya adalah untuk memberikan pengalaman ujian yang lebih interaktif dan informatif bagi karyawan, sekaligus memudahkan admin dalam melakukan evaluasi hasil tes. Berikut halaman Tes Mengetik :

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.6 Halaman Tes MarkazTest

The screenshot shows the 'MKZ\_UMUM-L1.xlsx' file. On the left, there is a table titled 'User Input' with columns 'No', 'Isi Materi', 'INF30', and 'INDOSAT REGULER'. The table has 20 rows, each containing a number from 1 to 20 and a corresponding material name. On the right, there is a table titled 'MKZ\_UMUM-L1.xlsx' with columns 'No', 'Indosat', 'INF30', and 'INDOSAT REGULER'. This table also has 20 rows, matching the user input table. The data in both tables is identical.

No	Isi Materi	INF30	INDOSAT REGULER
1	Indosat	INF30	INDOSAT REGULER
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

Gambar 3.7 Halaman User Input Tes Mengetik

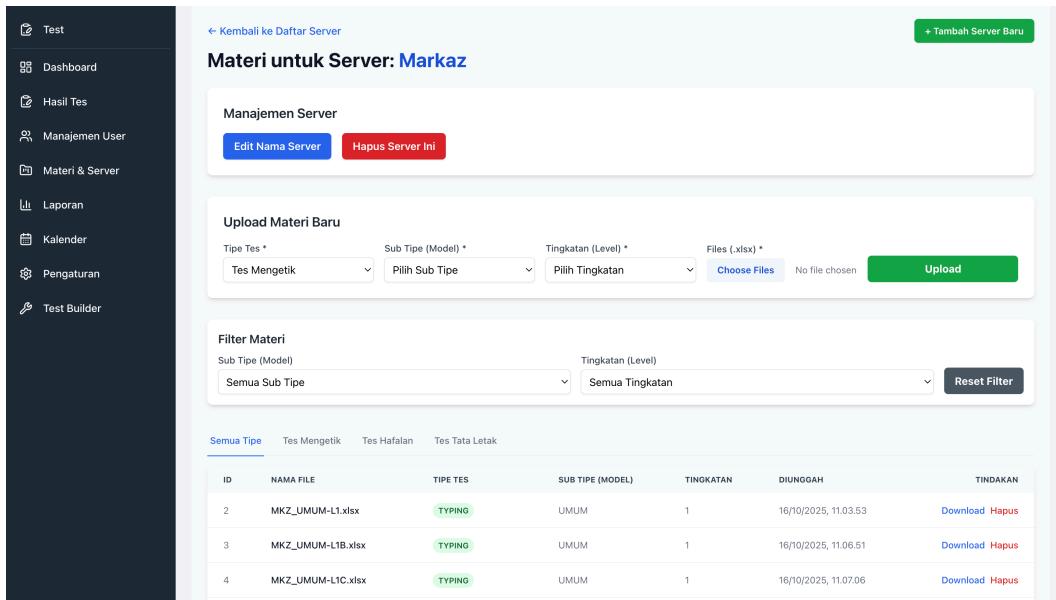
## B. Fase Manajemen Data dan Integrasi Backend (Minggu 5–7)

Memasuki fase kedua pelaksanaan kerja magang, fokus pengembangan sistem MarkazTest beralih dari pembangunan logika dasar ke pengelolaan data, integrasi backend, serta peningkatan skalabilitas sistem. Pada tahap ini, penulis mulai terlibat lebih dalam pada pengolahan data materi ujian, struktur database, serta optimalisasi komunikasi antara frontend dan backend. Fase ini

menjadi krusial karena sistem ujian online yang dikembangkan ditujukan untuk digunakan secara berkelanjutan dengan jumlah peserta dan materi yang terus bertambah.

Pada minggu kelima, penulis mengembangkan fitur input materi ujian berbasis file Excel. Pengembangan fitur ini dilatarbelakangi oleh kebutuhan admin agar tidak perlu memasukkan soal secara manual satu per satu melalui antarmuka aplikasi, mengingat jumlah materi tes yang cukup besar dan bersifat dinamis. Format Excel yang digunakan disesuaikan dengan standar internal perusahaan, yaitu terdiri dari kolom penyedia (provider), kode produk, dan deskripsi produk.

Secara teknis, proses upload file dilakukan menggunakan middleware pada backend yang bertugas menangani file berformat .xlsx. File yang diunggah kemudian diproses menggunakan library parser Excel untuk mengonversi data mentah menjadi struktur JSON. Data hasil parsing ini selanjutnya divalidasi sebelum disimpan ke database. Validasi mencakup pengecekan format data, jumlah kolom, serta konsistensi nilai untuk menghindari error saat proses ujian berlangsung. Dengan adanya fitur ini, proses manajemen bank soal menjadi lebih efisien dan terstandarisasi.

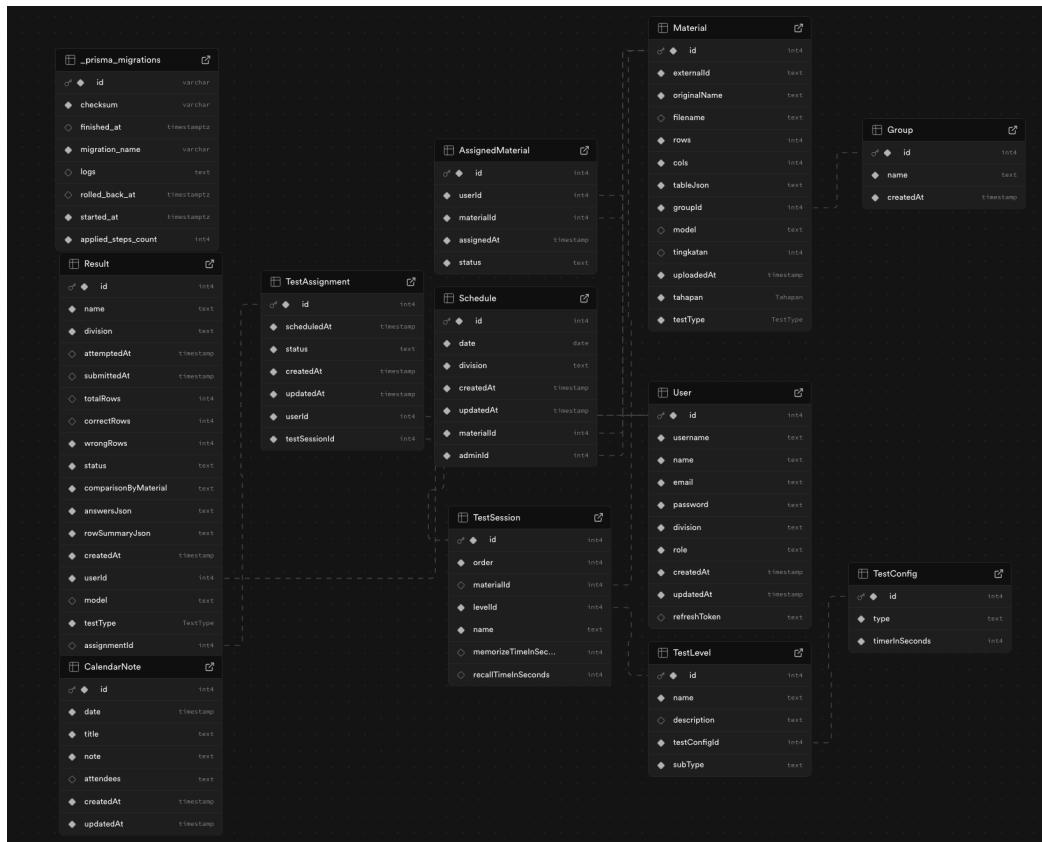


Gambar 3.8 Tampilan Halaman Unggah Materi

Pada minggu keenam, penulis melakukan migrasi sistem penyimpanan data ke database relasional menggunakan Prisma ORM. Sebelumnya, beberapa data masih dikelola secara lokal dan belum memiliki struktur relasional yang optimal. Migrasi ini bertujuan untuk meningkatkan integritas data, kemudahan maintenance, serta kesiapan sistem untuk penggunaan skala besar.

Penulis merancang skema database yang mencakup entitas utama seperti User, Material, Group, SubGroup, dan Result. Relasi antar entitas dirancang agar setiap hasil tes dapat ditelusuri kembali ke pengguna, jenis tes, serta materi yang digunakan. Prisma digunakan sebagai Object Relational Mapping (ORM) untuk mempermudah proses query data sekaligus menjaga konsistensi struktur database antara environment pengembangan dan produksi.

Proses migrasi dilakukan secara bertahap, dimulai dari pembuatan skema, menjalankan migrasi, hingga penyesuaian endpoint backend agar sesuai dengan struktur data yang baru. Selama proses ini, penulis juga melakukan debugging terhadap beberapa error relasi dan penyesuaian tipe data agar kompatibel dengan frontend.

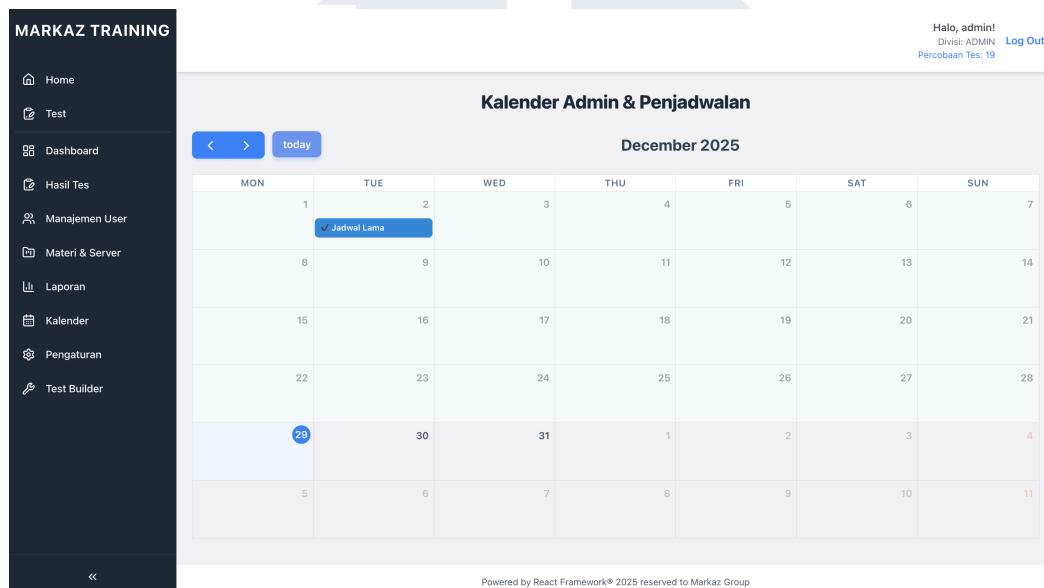


Gambar 3.9 Skema Database MarkazTest

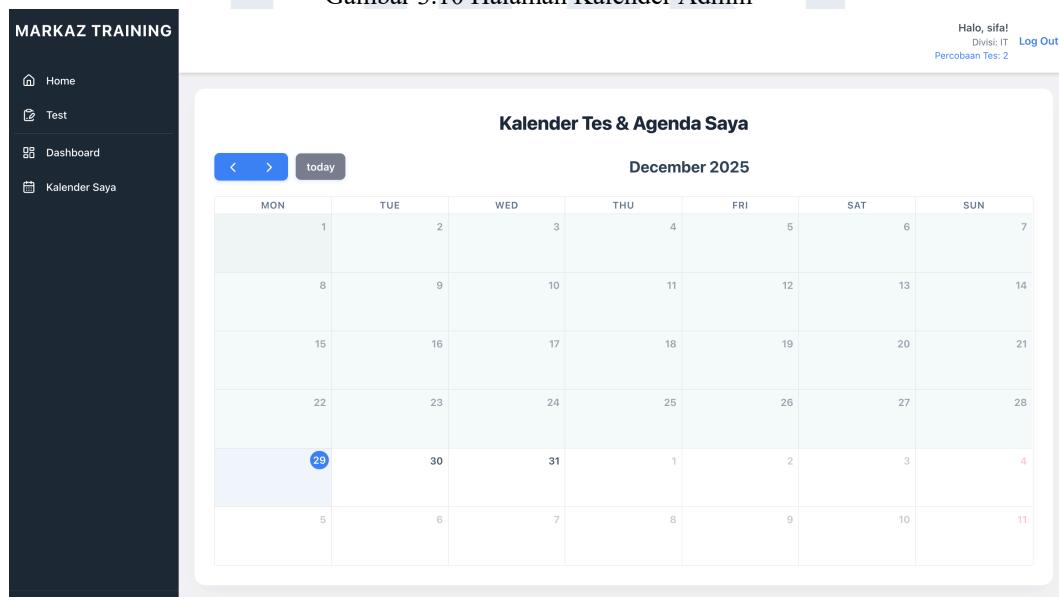
Pada minggu ketujuh, pengembangan difokuskan pada manajemen data berbasis pengelompokan serta penjadwalan ujian. Penulis menambahkan fitur grouping materi, yaitu pengelompokan soal ke dalam Group dan SubGroup agar admin dapat mengatur materi ujian sesuai kategori tertentu. Fitur ini sangat membantu dalam pembuatan ujian dengan variasi materi yang berbeda-beda untuk setiap sesi.

Selain itu, penulis mengembangkan fitur kalender ujian yang dapat diakses oleh admin dan user. Kalender ini digunakan untuk mengatur jadwal pelaksanaan tes, menampilkan tanggal ujian, serta membatasi akses peserta sesuai waktu yang telah ditentukan. Implementasi kalender ini juga melibatkan sinkronisasi data waktu antara frontend dan backend untuk memastikan konsistensi jadwal, terutama ketika sistem diakses dari berbagai perangkat.

Pada tahap ini, penulis mulai memahami pentingnya sinkronisasi data lintas modul, karena perubahan pada satu bagian sistem (misalnya materi atau jadwal) dapat berdampak langsung pada alur ujian dan hasil tes peserta. Oleh karena itu, pengujian integrasi dilakukan secara menyeluruh untuk memastikan setiap modul berjalan sesuai fungsinya.



Gambar 3.10 Halaman Kalender Admin



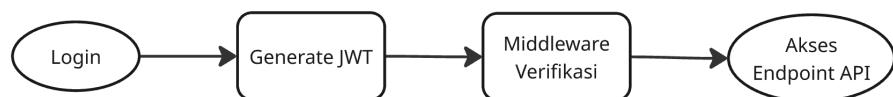
Gambar 3.11 Halaman Kalender User

### C. Fase 3: Keamanan Sistem dan Deployment (Minggu 8 – 10)

Memasuki fase ketiga, fokus pengembangan aplikasi MarkazTest beralih dari pembangunan fitur keamanan untuk memastikan data yang tersimpan di dalam web tetap terjaga dan aman.

Pada minggu kedelapan, penulis mulai mengimplementasikan sistem autentikasi dan otorisasi pengguna secara menyeluruh. Sistem login dan registrasi dikembangkan menggunakan pendekatan JSON Web Token (JWT) sebagai mekanisme autentikasi utama. Setiap pengguna yang berhasil login akan menerima token autentikasi yang digunakan untuk mengakses endpoint backend sesuai dengan hak aksesnya. Untuk menjaga keamanan data pengguna, password tidak disimpan dalam bentuk teks asli, melainkan melalui proses hashing menggunakan algoritma Bcrypt sebelum disimpan ke dalam database.

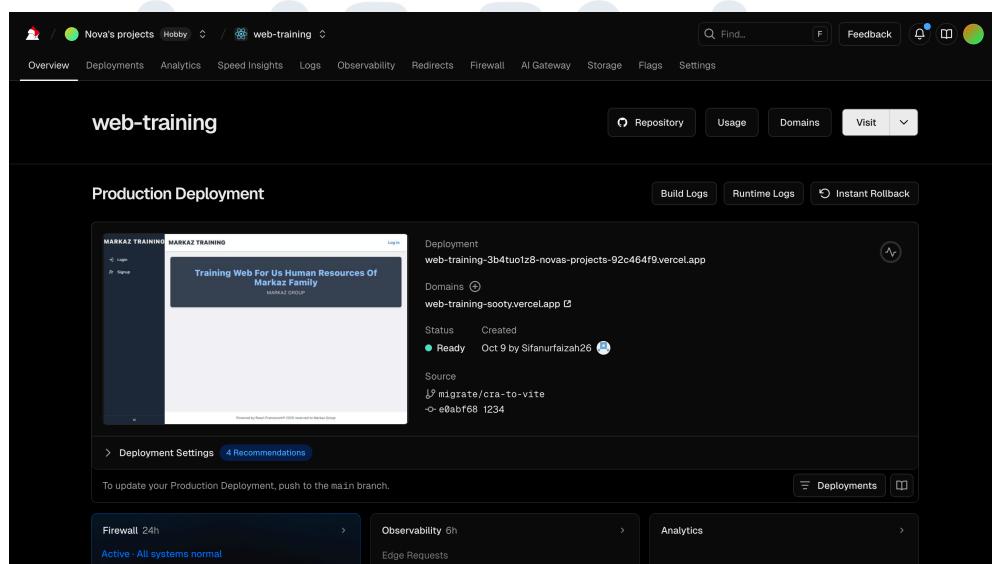
Selain autentikasi, penulis juga menerapkan konsep Role-Based Access Control (RBAC) untuk membedakan hak akses antara pengguna dengan peran Admin dan User. Admin memiliki akses penuh terhadap fitur manajemen data, pengaturan tes, serta laporan hasil ujian, sedangkan User hanya dapat mengakses fitur yang berkaitan dengan pelaksanaan tes dan melihat hasil pribadi. Implementasi RBAC ini dilakukan melalui middleware pada backend yang memvalidasi token JWT sekaligus memeriksa peran pengguna sebelum request diproses lebih lanjut.



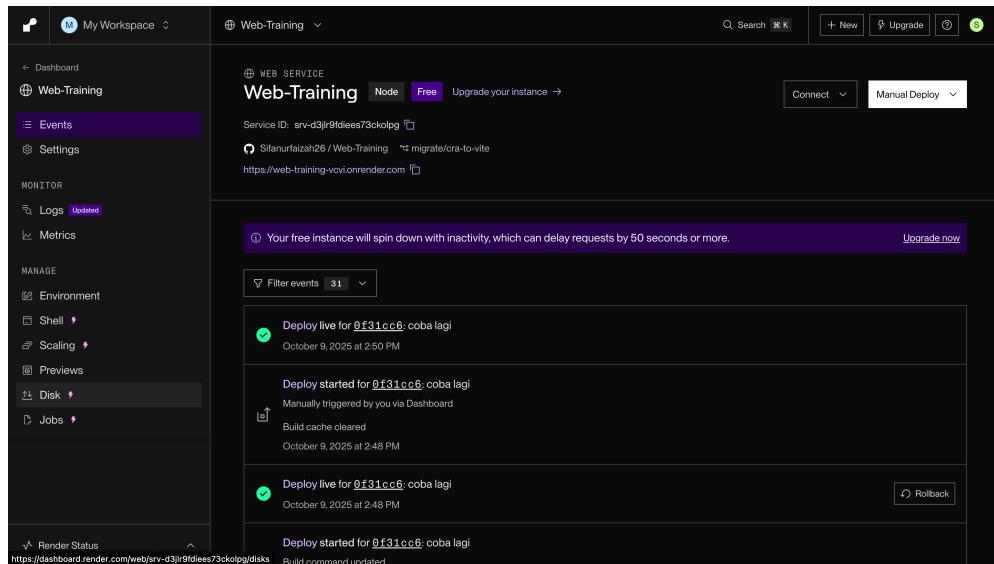
Gambar 3.12 Diagram Alur Autentikasi MarkazTest

Pada minggu kesembilan, aplikasi mulai dipersiapkan untuk deployment ke lingkungan produksi. Frontend aplikasi MarkazTest di-deploy menggunakan platform Vercel, sedangkan backend ditempatkan pada layanan Render. Proses deployment ini memerlukan penyesuaian konfigurasi environment, seperti pengaturan variabel lingkungan (environment variables), URL API, serta koneksi database (Supabase) agar sistem dapat berjalan dengan baik di luar lingkungan lokal.

Salah satu tantangan utama pada tahap ini adalah penanganan Cross-Origin Resource Sharing (CORS). Karena frontend dan backend berjalan pada domain yang berbeda, penulis perlu melakukan konfigurasi CORS pada backend agar komunikasi data dapat berlangsung secara aman tanpa melanggar kebijakan keamanan browser. Selain itu, penulis juga melakukan pengujian koneksi frontend-backend secara menyeluruh untuk memastikan seluruh endpoint API dapat diakses dengan benar setelah sistem berada di server produksi.



Gambar 3.13 Status Deployment Frontend



Gambar 3.14 Status Deployment Backend

Pada minggu kesepuluh, fokus pengembangan diarahkan pada manajemen pengguna dan pelaporan data. Penulis mengembangkan fitur manajemen user pada sisi admin, yang memungkinkan admin untuk melihat daftar pengguna, melakukan reset password, serta mengubah peran pengguna sesuai kebutuhan operasional. Selain itu, sistem laporan hasil tes juga diperbaiki dengan menambahkan fitur paginasi data, sehingga admin dapat mengelola dan meninjau hasil ujian dalam jumlah besar secara lebih efisien.

Pada tahap ini juga dilakukan berbagai pengujian fungsional dan non-fungsional, seperti pengujian keamanan endpoint, pengujian stabilitas sistem ketika diakses secara bersamaan, serta pengecekan alur penggunaan dari sisi user dan admin. Proses debugging dilakukan berdasarkan log server dan feedback dari supervisor untuk memastikan aplikasi berjalan stabil sebelum memasuki fase pengembangan fitur lanjutan.

#### D. Fase 4: Pengembangan Fitur Lanjutan (Minggu 11–14)

Pada fase keempat, fokus pengembangan aplikasi MarkazTest diperluas untuk mendukung jenis ujian selain Tes

Mengetik, yaitu Tes Hafalan dan Tes Tata Letak. Pengembangan pada fase ini bersifat lebih kompleks karena melibatkan variasi alur penggerjaan ujian, metode penilaian yang berbeda, serta penyesuaian struktur data agar sistem dapat menangani berbagai tipe tes dalam satu platform yang terintegrasi.

Tes Hafalan dirancang untuk menguji kemampuan peserta dalam mengingat materi tertentu dalam jangka waktu yang telah ditentukan. Oleh karena itu, sistem tes ini dibagi menjadi dua tahap utama, yaitu fase menghafal dan fase mengingat (recall).

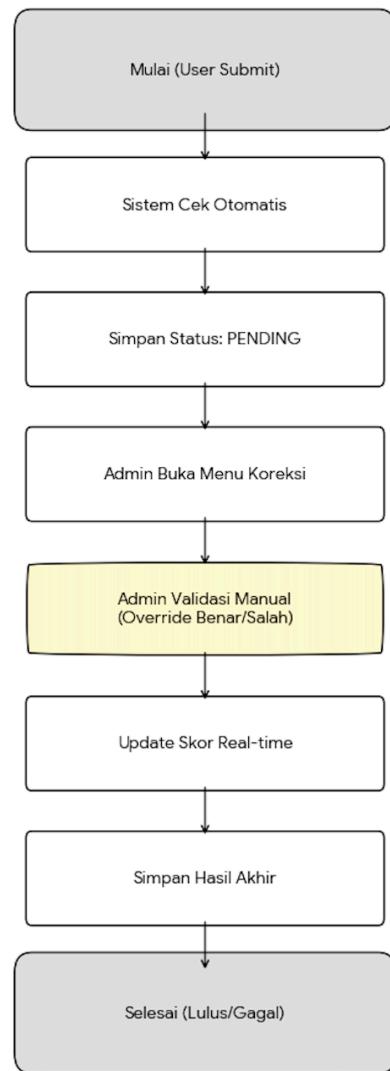
Pada fase menghafal, peserta diberikan waktu untuk membaca dan memahami materi yang ditampilkan oleh sistem. Setelah waktu habis, sistem secara otomatis berpindah ke fase recall, di mana peserta diminta menuliskan kembali materi yang telah dipelajari tanpa bantuan tampilan soal. Mekanisme penilaian pada Tes Hafalan menggunakan pendekatan semi-otomatis yang mengombinasikan pemeriksaan sistem dan validasi manual oleh admin. Pada tahap awal, sistem melakukan pengecekan otomatis dengan membandingkan jawaban peserta terhadap kunci jawaban menggunakan metode *exact match*. Namun, hasil pengecekan ini tidak langsung dijadikan nilai akhir, melainkan disimpan dengan status pending.

Selanjutnya, admin melakukan proses koreksi manual melalui halaman khusus yang menampilkan jawaban peserta berdampingan dengan materi referensi. Melalui mekanisme ini, admin dapat menilai apakah perbedaan jawaban masih dapat ditoleransi, seperti perbedaan huruf kapital, singkatan umum, atau variasi penulisan yang tidak mengubah makna. Jika dinilai benar, admin dapat melakukan *override* terhadap hasil sistem.

Setelah seluruh jawaban diverifikasi, sistem akan menghitung ulang skor akhir secara otomatis dan menentukan status kelulusan peserta. Pendekatan ini diterapkan untuk menjaga

keseimbangan antara ketelitian sistem dan fleksibilitas penilaian manusia, sehingga hasil evaluasi menjadi lebih adil dan akurat.

Berikut diagram logika dari tes hafalan:



Gambar 3.15 Diagram Alur Logika Tes Hafalan

Berbeda dengan Tes Mengetik yang menggunakan penilaian otomatis penuh, penilaian pada Tes Hafalan bersifat lebih subjektif. Oleh karena itu, penulis mengembangkan fitur koreksi manual, yang memungkinkan admin untuk meninjau jawaban peserta dan menentukan status benar atau salah secara langsung melalui dashboard admin. Perubahan hasil koreksi ini secara otomatis memperbarui skor akhir peserta di sistem.

**Tes Hafalan Manual (Level 1)**

Hafalkan materi di bawah ini. Waktu Anda:

1:57

1	Indosat	IDPUNF25	DATA ISAT PURE 2,5GB ALL NET 24 JAM 30HR NF
2	XL	XXCF5	DATA XL COMBO Flex XXL (65GB + Lokal) 28H
3	Three	TH#2	DATA THREE HAPPY 2,5GB+0,5GB/LK 1HR
4	Telkomsel	TDM1A	DATA TSEL MINI 1 GB 24 JAM 1 HR
5	Three	TRIDB2500	DATA BULK THREE 2,5GB 14HR
6	Smart	SVOL60	DATA 10GB+2GB(01-05)+Telp/Ses, 30Hr
7	Indosat	IDPM215	DATA ISAT FREEDOM INTERNET HARIAN 1GB 2HARI
8	XL	XDP4000	DATA XL Kuota Pure 4GB 30 Hari
9	Axis	IDSS533	DATA ISAT SACHET 5GB 3HR
10	Axis	AKP500	DATA AXIS PURE 500MB 30 HARI
11	Smart	SMTNF26	TRANSFER PULSA SMARTFREN NF 26K
12	Smart	SULID1B	SMART DATA U 1GB/H 14HR

Saya Sudah Hafal, Lanjut Mengerjakan

Gambar 3. 16 Tes Hafalan MarkazTest

**Tes Hafalan Manual (Level 1)**

Tulis kembali materi yang Anda hafal. Waktu Anda:

4:51

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

Selesai & Kumpulkan

Gambar 3.17 User Input Tes Hafalan  
Tampilan Koreksi yang hanya bisa diakses oleh admin dapat dilihat pada gambar berikut :

NO	KUNCI JAWABAN (REFERENSI)	JAWABAN USER (EDITABLE)	KOREKSI
1	Indosat IDPUNF25 DATA ISAT PURE 2,5GB ALL NET 24 JAM 30HR NF	<input type="text"/> X <input type="text"/> X	X
2	XL XXXCF5 DATA XL COMBO Flex XXL (65GB + Lokal) 28H	<input type="text"/> X <input type="text"/> X	X
3	Three THM2 DATA THREE HAPPY 2,5GB+0,5GBLKL 1HR	<input type="text"/> X <input type="text"/> X	X
4	Telkomsel TDM1A DATA TSEL MINI 1 GB 24 JAM 1 HR	<input type="text"/> X <input type="text"/> X	X
5	Three TRIDB2500 DATA BULK THREE 2,5GB 14HR	<input type="text"/> X <input type="text"/> X	X
6	Smart SVOL60 DATA 10GB+20GB(01-05)+TeleSes,30Hr	<input type="text"/> X <input type="text"/> X	X

Gambar 3.18 Tampilan Halaman Koreksi Oleh Admin

Seiring bertambahnya jenis tes, struktur database awal yang hanya mendukung Tes Mengetik tidak lagi memadai. Pada tahap ini, dilakukan restrukturisasi database untuk mendukung konsep Test Builder Dinamis, yaitu sistem yang memungkinkan admin membuat dan mengatur berbagai tipe tes dalam satu platform.

Penulis memisahkan konfigurasi tes berdasarkan tipe (Mengetik, Hafalan, dan Tata Letak), termasuk pengaturan durasi, metode penilaian, dan struktur soal. Perubahan ini juga diikuti dengan penyesuaian endpoint backend serta sinkronisasi tampilan frontend admin agar pengelolaan tes menjadi lebih fleksibel dan terstruktur.

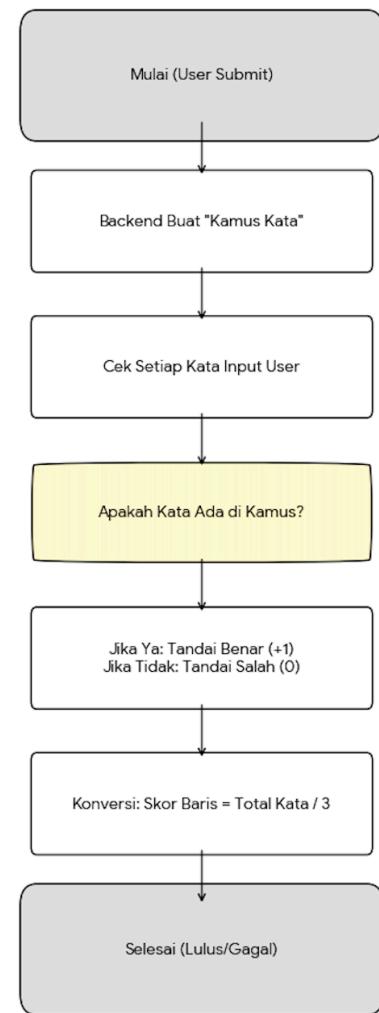
Tes Tata Letak dikembangkan untuk menguji pemahaman peserta terhadap susunan atau posisi elemen tertentu berdasarkan materi yang diberikan. Penilaian pada Tes Tata Letak menggunakan pendekatan *flexible keyword mapping* yang bertujuan memberikan fleksibilitas dalam penilaian meskipun urutan soal telah diacak. Pada tahap awal, backend membangun struktur data berupa *kamus jawaban (frequency map)* dengan cara memecah seluruh materi soal

menjadi kumpulan kata kunci dan menghitung jumlah kemunculannya.

Ketika peserta menginput jawaban, sistem akan mencocokkan setiap kata yang diketik ke dalam kamus tersebut. Jika kata ditemukan dan masih memiliki stok ketersediaan, maka jawaban tersebut dinilai benar, dan stok kata akan dikurangi untuk mencegah penghitungan ganda. Sebaliknya, jika kata tidak ditemukan atau stok telah habis, maka jawaban dinilai salah.

Jumlah kata yang berhasil dijawab dengan benar kemudian dikonversi menjadi satuan baris sesuai dengan jumlah kolom dalam satu baris soal. Metode ini memungkinkan sistem memberikan penilaian yang adil meskipun peserta mengerjakan soal secara tidak berurutan, sekaligus menjaga akurasi evaluasi terhadap penguasaan materi.

Selain itu, sistem penilaian pada Tes Tata Letak menggunakan pendekatan penilaian fleksibel berbasis kata kunci. Backend tidak hanya membandingkan jawaban secara persis, tetapi juga menggunakan konsep *kamus jawaban*, di mana jawaban dianggap benar apabila mengandung kata kunci penting yang sesuai dengan materi, meskipun urutan penulisannya berbeda, berikut diagram logika dari tes tata letak:



Gambar 3.19 Diagram Alur Logika Tes Tata Letak  
 Pengembangan fitur ini diuji secara menyeluruh melalui pengujian end-to-end untuk memastikan seluruh tipe tes dapat berjalan dengan baik dalam satu sistem yang terintegrasi.

UNIVERSITAS  
 MULTIMEDIA  
 NUSANTARA

Materi Referensi			Lembar Jawaban
1 Indosat	IDPUNF25	DATA ISAT PURE 2,5GB ALL NET 24 JAM 30HR NF	1
2 XL	XXCF5	DATA XL COMBO Flex XXL (65GB + Lokal) 28H	2
3 Three	THM2	DATA THREE HAPPY 2,5GB+0,5GBLKL 1HR	3
4 Telkomsel	TDM1A	DATA TSEL MINI 1 GB 24 JAM 1 HR	4
5 Three	TRIDB2500	DATA BULK THREE 2,5GB 14HR	5
6 Smart	SVOL60	DATA 10GB+20GB(61-05)+TeleSes,30H	6
7 Indosat	IDFM215	DATA ISAT FREEDOM INTERNET HARIAN 1GB 2HARI	7
8 XL	XDP4000	DATA XL Kuota Pure 4GB 30 Hari	8
9 Axis	IDSS533	DATA ISAT SACHET 5GB 3HR	9
10 Axis	AXP500	DATA AXIS PURE 500MB 30 HARI	10
11 Smart	SMTNF26	TRANSFER PULSA SMARTFREN NF 26K	11
12 Smart	SULID018	SMART DATA U 1GB/H 14HR	12
13 Telkomsel	SANF4	TELKOMSEL REGULER 4K	13
14 Indosat	INF12	INDOSAT REGULER 12K	14
15 XL	XXCF1A	DATA XL COMBO Flex S+ (6,5GB + Lokal) 28H	15
16 Three	TRIDB750	DATA BULK THREE 750MB 14HR	16
17 Smart	HSMNNF70	REGULER SMART NF 70K	17
18 XL	XBPA13	DATA XL BEBAS PIAS 3,25GB 13H	18

Gambar 3. 20 Tampilan Halaman User Input Tes Tata Letak  
E. Fase 5: Finalisasi dan Serah Terima (Minggu 15–17)

Memasuki fase akhir pelaksanaan kerja magang, fokus kegiatan diarahkan pada proses finalisasi sistem MarkazTest sebelum diserahkan kepada pihak perusahaan. Pada tahap ini, penulis tidak lagi mengembangkan fitur utama, melainkan memastikan seluruh fungsi sistem berjalan dengan stabil, konsisten, dan siap digunakan secara operasional. Fase ini menjadi tahap krusial karena mencerminkan kesiapan aplikasi untuk digunakan dalam lingkungan produksi nyata.

Pada minggu ke-15 hingga minggu ke-16, penulis melakukan quality assurance (QA) dan pengujian sistem secara menyeluruh. Pengujian dilakukan dengan mensimulasikan berbagai skenario penggunaan, baik dari sisi pengguna (user) maupun administrator (admin). Penulis menguji alur penggerjaan tes mulai dari proses login, pemilihan jadwal ujian, penggerjaan tes, hingga penampilan hasil akhir. Selain itu, pengujian juga dilakukan pada sisi admin, seperti pengelolaan materi, pengaturan jadwal, koreksi manual tes hafalan, serta pengelolaan data pengguna.

Salah satu fokus utama pada tahap ini adalah sinkronisasi zona waktu (timezone) pada sistem kalender dan timer ujian. Sebelumnya, ditemukan perbedaan waktu antara server dan sisi klien yang menyebabkan ketidaksesuaian jadwal ujian. Penulis melakukan penyesuaian dengan memastikan seluruh pengolahan waktu menggunakan standar waktu yang konsisten, baik pada backend maupun frontend. Perbaikan ini bertujuan agar jadwal ujian, countdown timer, dan waktu penyimpanan hasil tes berjalan secara akurat.

Selain itu, penulis juga melakukan perbaikan bug (bug fixing) yang ditemukan selama proses pengujian, seperti kesalahan tampilan UI, error validasi input, serta penanganan edge case pada proses penilaian. Beberapa perbaikan juga dilakukan pada performa aplikasi agar lebih responsif, terutama pada halaman admin yang menampilkan data dalam jumlah besar. Untuk mendukung keamanan dan kejujuran peserta ujian, penulis menerapkan fitur anti-cheat sederhana, seperti menonaktifkan fungsi salin (copy) dan tempel (paste) selama pengeraan tes.

Pada minggu ke-17, kegiatan difokuskan pada penulisan laporan dan penyelesaian administrasi magang. Penulis juga melakukan deployment final untuk uji testing finalisasi menandakan aplikasi web MarkazTest resmi selesai. Dilakukan juga monitoring yang memastikan aplikasi berjalan sesuai keinginan yang sudah disepakati.

Selain itu, penulis juga menyelesaikan laporan kerja magang, yang berisi seluruh rangkaian kegiatan selama periode magang, mulai dari pengenalan perusahaan, pelaksanaan tugas, hingga evaluasi dan pembelajaran yang diperoleh. Proses ini menandai berakhirnya kegiatan magang dan menjadi bentuk pertanggungjawaban akademik atas pekerjaan yang telah dilakukan.

Secara keseluruhan, fase finalisasi dan serah terima ini memastikan bahwa aplikasi MarkazTest berada dalam kondisi stabil, fungsional, dan layak digunakan. Tahap ini juga memberikan pengalaman berharga bagi penulis dalam memahami pentingnya pengujian, dokumentasi, dan kesiapan sistem sebelum diterapkan dalam lingkungan operasional perusahaan.

### 3.3.2 Limitasi

Meskipun aplikasi MarkazTest telah berhasil dikembangkan dengan berbagai fitur utama, seperti *Test Builder*, pengaturan tes (*Admin Settings*), serta mekanisme penilaian otomatis dan semi-otomatis, sistem ini masih memiliki beberapa keterbatasan yang perlu diperhatikan. Batasan ini disebabkan oleh keterbatasan waktu pelaksanaan kerja magang, ruang lingkup proyek yang telah ditentukan perusahaan, serta pertimbangan stabilitas sistem pada tahap implementasi awal.

Adapun beberapa limitasi sistem MarkazTest adalah sebagai berikut:

1. Keterbatasan pada Fitur Test Builder

Fitur *Test Builder* dirancang untuk mempermudah admin dalam mengelola materi ujian dan menyusun paket tes secara dinamis. Namun, pada implementasinya saat ini, sistem masih memiliki beberapa batasan, antara lain:

- Sistem hanya mendukung unggahan materi melalui file berformat (Excel) dengan struktur kolom tertentu, yaitu *provider*, *kode produk*, dan *deskripsi*.
- Admin belum dapat menambahkan soal secara manual satu per satu melalui antarmuka aplikasi. Hal ini dilakukan untuk menjaga konsistensi format data dan efisiensi pengolahan soal dalam jumlah besar.
- Jenis tes yang tersedia masih terbatas pada tiga tipe, yaitu Tes Mengetik, Tes Hafalan, dan Tes Tata Letak,

sehingga belum mendukung pembuatan tipe tes baru tanpa pengembangan lanjutan di sisi backend.

## 2. Keterbatasan pada Pengaturan Tes (Admin Settings)

Menu *Admin Settings* digunakan untuk mengatur konfigurasi global ujian, seperti durasi tes, metode penilaian, dan ambang batas kelulusan. Beberapa keterbatasan pada fitur ini antara lain:

- Pengaturan timer bersifat global per jenis tes, sehingga perubahan durasi akan berdampak pada seluruh sesi ujian dengan tipe yang sama, kecuali dilakukan pengaturan khusus di tingkat sesi.
- Pengaturan skoring masih bersifat terstandarisasi dan belum mendukung konfigurasi penilaian yang lebih fleksibel per kelompok peserta atau per paket soal.

Dengan adanya limitasi tersebut, diharapkan pengembangan MarkazTest di masa mendatang dapat terus dilakukan secara bertahap guna meningkatkan fleksibilitas, skalabilitas, serta kenyamanan penggunaan sistem, baik dari sisi administrator maupun peserta ujian.

### 3.3.3 Kendala yang Ditemukan

Selama pelaksanaan kerja magang dan proses pengembangan aplikasi ujian online MarkazTest, penulis menemui beberapa kendala baik dari sisi teknis maupun non-teknis. Kendala-kendala tersebut merupakan hal yang wajar dalam proses pengembangan sistem berbasis web, terutama pada proyek yang dikembangkan secara bertahap dan terus mengalami penyesuaian sesuai kebutuhan perusahaan.

Pada tahap awal pelaksanaan magang, penulis menghadapi kendala dalam memahami keseluruhan ekosistem sistem yang digunakan oleh perusahaan. Markaz Group memiliki beberapa sistem

internal, seperti OtomaX dan sistem training, yang saling berkaitan secara tidak langsung dengan pengembangan aplikasi ujian online. Banyaknya sistem dan alur bisnis yang harus dipahami membuat penulis membutuhkan waktu untuk menyesuaikan diri dan memahami konteks penggunaan aplikasi secara menyeluruh.

Selain itu, kebutuhan sistem ujian online pada awalnya belum terdokumentasi secara rinci. Beberapa fitur mengalami perubahan konsep seiring berjalannya waktu, sehingga penulis harus sering melakukan penyesuaian desain dan logika aplikasi.

Dalam proses pengembangan, penulis mengalami beberapa kendala teknis, antara lain:

- Perbedaan struktur data antara frontend dan backend yang menyebabkan error saat integrasi API.
- Kompleksitas algoritma penilaian, khususnya pada penerapan row-based scoring dan highlight karakter pada Tes Mengetik.
- Kendala parsing file Excel dengan format tertentu, seperti perbedaan jumlah kolom, baris kosong, atau format data yang tidak konsisten.
- Masalah koneksi antara frontend dan backend setelah proses deployment, khususnya terkait konfigurasi CORS dan environment variable.

Pada saat dilakukan migrasi backend ke Prisma ORM dan database relasional, penulis menghadapi kendala dalam penyesuaian skema database. Beberapa relasi antar tabel harus diubah agar mendukung fitur lanjutan seperti multi-tipe tes, multi-sesi, serta pengelompokan materi. Proses migrasi ini berisiko menyebabkan data tidak sinkron apabila tidak dilakukan dengan hati-hati. Kendala lain yang dihadapi adalah manajemen waktu antara penggerjaan fitur, perbaikan bug, serta koordinasi dengan supervisor dan tim. Beberapa pekerjaan membutuhkan diskusi lanjutan sebelum dapat diimplementasikan, sehingga waktu pengembangan terkadang harus

disesuaikan dengan kebutuhan pada saat-saat tertentu dan dikerjakan sendiri

### 3.3.4 Solusi atas Kendala yang Ditemukan

Untuk mengatasi berbagai kendala yang muncul selama pelaksanaan kerja magang, penulis menerapkan beberapa solusi dan pendekatan yang dilakukan secara bertahap dan terstruktur.

Penulis secara aktif melakukan diskusi dengan supervisor dan terkait untuk memperjelas alur bisnis dan tujuan pengembangan sistem. Selain itu, penulis mempelajari dokumentasi internal perusahaan serta mengamati langsung penggunaan sistem existing seperti OtomaX dan sistem training. Pendekatan ini membantu penulis memahami kebutuhan perusahaan secara lebih komprehensif sebelum mengimplementasikan fitur baru.

Untuk mengatasi kendala teknis, penulis melakukan beberapa langkah berikut:

- Menyeragamkan struktur data API antara frontend dan backend melalui penyesuaian response JSON.
- Memecah logika penilaian menjadi fungsi-fungsi kecil agar lebih mudah diuji dan diperbaiki.
- Menambahkan validasi input pada proses upload file Excel untuk meminimalkan kesalahan format data.
- Melakukan debugging secara bertahap menggunakan log server dan tools developer pada browser.

Dalam proses migrasi database, penulis menggunakan fitur migration dan schema validation pada Prisma ORM. Setiap perubahan skema dilakukan secara bertahap dan diuji terlebih dahulu pada environment pengembangan sebelum diterapkan ke server produksi. Dengan cara ini, risiko kehilangan data dan inkonsistensi database dapat diminimalkan.

Untuk mengatasi kendala manajemen waktu, penulis membuat daftar prioritas pekerjaan dan membagi tugas pengembangan ke dalam

fase-fase mingguan. Koordinasi rutin dengan supervisor dilakukan untuk memastikan pekerjaan yang dikerjakan sudah sesuai dengan kebutuhan perusahaan dan tidak menyimpang dari tujuan awal proyek.



**UMN**  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA