BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Berlangsungnya kegiatan magang pada Perusahaan Govokasi selama satu semester, kedudukan ditempatkan sebagai *Co-Head of Tech Department* sebagai Fullstack Developer. Posisi magang dibimbing oleh Andryan Kalmer Wijaya selaku *Head of Tech Department* dan *Supervisor*. Tanggung jawab utama dalam posisi ini adalah menjalankan proyek-proyek teknologi yang telah direncanakan oleh *Head of Tech Department*, serta bertanggung jawab dalam pengembangan aplikasi dan website, baik di sisi *frontend* maupun *backend*. *Head of Tech Department*, yang bertanggung jawab langsung kepada CEO, menyusun perencanaan proyek yang kemudian dilaksanakan oleh *Co-Head of Tech Department*. Koordinasi dilakukan dengan tim untuk memastikan bahwa setiap proyek berjalan sesuai dengan perencanaan dan mencapai tujuan yang telah ditetapkan.

Proses perencanaan dan pelaksanaan proyek dimulai dengan penyusunan perencanaan oleh *Head of Tech Department*, yang kemudian disampaikan kepada CEO. Setelah perencanaan tersebut disetujui, proyek ditindaklanjuti dengan penugasan yang harus dikerjakan, yang kemudian dilaksanakan oleh *Co-Head of Tech Department*. Komunikasi dan koordinasi yang intens antara *Co-Head* dan *Head Department* dilakukan untuk memastikan proyek dapat diselesaikan tepat waktu dan sesuai dengan harapan. Evaluasi kinerja proyek juga dilakukan secara berkala untuk memantau perkembangan dan mengatasi hambatan yang muncul dalam setiap tahapan proyek.

3.2 Tugas yang Dilakukan

Selama periode magang, beberapa tugas utama yang dijalankan diantaranya adalah pembuatan aplikasi *dashboard* untuk tracking proses lamaran menggunakan *frontend* FlutterFlow. Aplikasi ini bertujuan untuk mempermudah pemantauan status lamaran oleh pihak terkait dan memberikan visualisasi yang jelas terkait kemajuan setiap tahapan proses rekrutmen. Sebagai bagian dari pengelolaan data, juga dilakukan pembuatan struktur *database* menggunakan Supabase untuk mendukung aplikasi dan sistem informasi yang berjalan. Struktur *database* ini dirancang agar dapat menyimpan data lamaran, informasi kandidat, serta status

proses rekrutmen dengan efisien dan aman.

Selain itu, tugas lain yang dilaksanakan adalah membuat desain antarmuka website perusahaan di Figma, yang mencakup pembuatan desain untuk halaman utama yang bertujuan untuk menarik perusahaan, halaman talent page untuk menarik talenta, serta halaman tentang tim dan kontak. Desain ini kemudian diimplementasikan ke dalam website perusahaan menggunakan *platform* Zoho, untuk memastikan tampilan yang menarik dan fungsionalitas yang optimal.

3.3 Uraian Pelaksanaan Magang

Proses kerja mengikuti siklus pengembangan mingguan yang terstruktur berdasarkan prioritas proyek dan kebutuhan sistem. Pelaksanaan kerja magang minggu ke-1 hingga 7 berfokus pada pengembangan website perusahaan dengan Zoho Sites. Pelaksanaan kerja magang minggu kedelapan sampai empatbelas baru berfokus ke pengembangan aplikasi yang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1-2	Pengenalan terhadap lingkungan kerja, mempelajari alur kerja perusahaan, dan mengikuti konferensi Zoho untuk memahami aplikasi yang akan digunakan sebagai pembuatan website
3	Memahami *customer profile* dan *Key Visual* dari Govokasi untuk menciptakan desain yang lebih relevan dan menarik. Selain itu, dilakukan juga review terhadap website Govokasi yang sebelumnya, untuk mengevaluasi elemen-elemen yang sudah ada dan mengidentifikasi area yang dapat ditingkatkan, sehingga desain baru dapat memenuhi ekspektasi pengguna dan perusahaan dengan lebih baik.
4	Membuat desain home page di Figma dan mengimplementasikannya ke dalam website yang responsif menggunakan Zoho Sites.

UNIVERSITAS MULTIMEDIA NUSANTARA

Minggu Ke -	Pekerjaan yang dilakukan
5	Membuat desain homepage baru dengan memberikan tiga opsi desain berbeda. Dari ketiga desain tersebut, dipilih satu desain terbaik yang kemudian diimplementasikan ke Zoho Sites, dengan tambahan halaman "*Our Team*" dan "*Contact*".
6	Membuat tiga desain berbeda untuk talent page, yang kemudian dipilih satu desain terbaik untuk diimplementasikan ke Zoho Sites. Proses ini juga mencakup pembuatan versi responsif dari halaman tersebut, memastikan tampilan yang optimal dan aksesibilitas yang baik di berbagai perangkat.
7	Membuat tiga desain landing page di Figma yang disesuaikan dengan top 3 customer profile Govokasi, yaitu dari industri kecantikan, agensi digital, dan marketing. Setiap desain difokuskan untuk memenuhi kebutuhan dan preferensi masing-masing industri, dengan tujuan menarik perhatian dan meningkatkan konversi dari target audiens.
8	Melakukan perancangan pembuatan aplikasi dengan membahas penggunaan tools FlutterFlow dan Supabase. Mempelajari FlutterFlow, termasuk aspek *backend*-nya, seperti pemanggilan data dari Supabase. Proses ini melibatkan pemahaman cara mengintegrasikan Supabase dengan FlutterFlow
9	Membuat flowchart dan membuat struktur *database* dengan merancang skema database. Setelah itu, struktur *database* tersebut diimplementasikan di Supabase.
10	Mengembangkan fitur CRUD *dashboard* internal dan *dashboard* perusahaan mitra

UNIVERSITAS MULTIMEDIA NUSANTARA

Minggu Ke -	Pekerjaan yang dilakukan
11	Mengembangkan fitur merekomendasikan talent pada *dasboard* internal
12	Mengembangkan fitur *posting job* untuk perusahaan mitra
13	Mengembangkan fitur penerimaan talent dengan *swipe* hingga halaman konfirmasi *cart*
14	Merapikan dan menyempurnakan UI aplikasi. Proses ini melibatkan perbaikan tampilan antarmuka pengguna

Pada minggu ke-8 hingga 9, fokus utama adalah perancangan dan pemahaman dasar aplikasi, termasuk pemilihan *tools* FlutterFlow dan Supabase, serta pembelajaran mengenai cara mengintegrasikan keduanya untuk mengelola dan menampilkan data secara dinamis di aplikasi, serta memastikan konektivitas yang lancar antara *frontend* dan *database*. Selama minggu-minggu ini, dilakukan pembuatan flowchart untuk menggambarkan alur aplikasi dan pembuatan skema database untuk merancang struktur database dan menggambarkan hubungan antar tabel yang kemudian diimplementasikan di Supabase.

Pada minggu ke-10 hingga 13, pengembangan aplikasi berfokus pada pembuatan fitur-fitur utama. Pada minggu ke-10, fitur CRUD pada *dashboard* internal dan tampilan *dashboard* perusahaan mitra mulai dikembangkan. Minggu berikutnya (minggu ke-11) dilanjutkan dengan pengembangan fitur rekomendasi talent di *dashboard* internal, sementara pada minggu ke-12, fitur *job posting* untuk perusahaan mitra dikembangkan. Hingga pada minggu ke-13, mengembangkan fitur penerimaan talent dengan cara *swipe*, yang mencakup proses hingga halaman konfirmasi di *cart* oleh perusahaan mitra.

Pada minggu ke-14, pengembangan berfokus pada penyelesaian dan penyempurnaan aplikasi. Merapikan dan menyempurnakan UI aplikasi untuk memastikan desain antarmuka pengguna lebih intuitif, konsisten, dan menarik.

3.3.1 Software dan Hardware yang digunakan

Berikut merupakan *software* dan *hardware* yang digunakan selama kegiatan kerja magang di PT Gerbang Govokasi Indonesia

A Software

Software yang digunakan selama pengujian fungsionalitas fitur dalam kegiatan magang di Govokasi meliputi.

1. Figma, merupakah alat desain berbasis *cloud* yang digunakan untuk merancang antarmuka pengguna (UI) dan pengalaman pengguna (UX). Alat ini memungkinkan kolaborasi secara *real-time*, memungkinkan tim untuk bekerja bersama dalam mendesain, membuat prototipe, dan berinteraksi langsung dengan elemen desain. Figma digunakan untuk membuat wireframe, desain halaman, serta prototipe interaktif yang dapat diuji dan dievaluasi sebelum implementasi. Kelebihan utama Figma adalah kemampuannya untuk diakses secara online, memfasilitasi kolaborasi tim yang lebih efisien, serta fitur berbagi yang memudahkan pengambilan umpan balik dari pemangku kepentingan. Figma mendukung berbagai jenis file dan memungkinkan desain yang responsif, memastikan tampilan yang optimal pada perangkat yang berbeda [4].

2. Zoho Sites, merupakan *platform* pembuat website berbasis *drag-and-drop* yang memungkinkan pengguna untuk membuat dan mengelola situs web dengan mudah. Zoho Sites menawarkan berbagai *template* dan fitur yang dapat disesuaikan untuk membuat situs web responsif dan profesional. *platform* ini memungkinkan integrasi dengan berbagai aplikasi Zoho lainnya, serta mendukung fitur SEO, formulir, dan alat pemasaran untuk meningkatkan visibilitas dan interaksi dengan pengunjung situs. Selain itu, Zoho Sites memberikan fleksibilitas untuk menggunakan custom code, memungkinkan pengguna untuk menyesuaikan berbagai elemen yang tidak tersedia secara default. Dengan fitur ini, pengembangan dan kustomisasi elemen desain dan fungsionalitas yang lebih spesifik dapat dilakukan untuk memenuhi kebutuhan yang lebih kompleks. Zoho Sites digunakan untuk mengimplementasikan desain yang telah dibuat di Figma ke dalam website perusahaan, memastikan bahwa halaman-halaman seperti *homepage*, *talent page*, dan halaman lainnya dapat diakses dengan baik di berbagai perangkat dan memenuhi kebutuhan fungsionalitas perusahaan [5].

3. Asana, merupakan *platform* manajemen proyek yang memungkinkan tim untuk merencanakan, melacak, dan mengelola pekerjaan mereka dengan efisien. Alat ini digunakan untuk mengorganisir tugas, menetapkan

tenggat waktu, dan memonitor progres proyek secara *real-time*. Asana memfasilitasi kolaborasi tim dengan menyediakan fitur seperti penugasan tugas, pencatatan komentar, serta pelaporan status proyek. Dalam konteks proyek pengembangan website dan aplikasi, Asana digunakan untuk merencanakan alur kerja, menetapkan prioritas, dan memastikan setiap anggota tim dapat memantau dan menyelesaikan tugas-tugas mereka tepat waktu. Dengan Asana, manajemen proyek dapat dilakukan secara transparan, mempermudah koordinasi antar tim, dan mengoptimalkan efisiensi dalam menyelesaikan setiap proyek yang ada [6].

4. Flutterflow, merupakan *platform* visual untuk pengembangan aplikasi mobile dan web yang memungkinkan pengguna untuk merancang dan membangun aplikasi secara intuitif. Menggunakan *framework* Flutter, FlutterFlow memungkinkan pengembangan aplikasi yang dapat berjalan di berbagai *platform*, termasuk Android, IOS, dan web. *platform* ini mendukung pembuatan antarmuka pengguna (UI) yang responsif dengan berbagai elemen yang dapat disesuaikan, seperti *widget*, animasi, dan navigasi. FlutterFlow juga memungkinkan integrasi *backend*, seperti Supabase, untuk mengelola dan menyimpan data secara dinamis. Alat ini sangat berguna untuk mempercepat proses pengembangan aplikasi, memudahkan pembuatan aplikasi yang kuat, skalabel, dan berfungsi dengan baik di berbagai perangkat dan *platform* [3].

5. Supabase, merupakan *platform database open-source* yang menyediakan layanan manajemen data berbasis PostgreSQL. Sebagai alternatif open-source untuk Firebase, Supabase memungkinkan pengembang untuk membuat dan mengelola *database* relasional dengan mudah. Dengan fitur *real-time*, Supabase memungkinkan aplikasi untuk melakukan pembaruan data secara langsung dan otomatis di seluruh klien yang terhubung. *platform* ini menyediakan berbagai fitur untuk manajemen data, seperti otentikasi pengguna, pengelolaan file, dan fungsi *server-side*, yang semuanya dapat diakses melalui API sederhana. Supabase juga mendukung kemampuan query SQL yang fleksibel, memberikan kontrol penuh atas struktur dan manipulasi data. Dengan antarmuka yang ramah pengguna, Supabase memungkinkan pengembang untuk membangun *database* yang efisien, aman, dan skalabel.

B Hardware

Hardware yang digunakan selama mengerjakan website dan aplikasi dalam kegiatan magang di Govokasi meliputi.

1. Prosesor: AMD Ryzen 7 7840HS.
2. Memori: 16GB.
3. Penyimpanan: 512GB.
4. Sistem Operasi: Windows 11 Home.

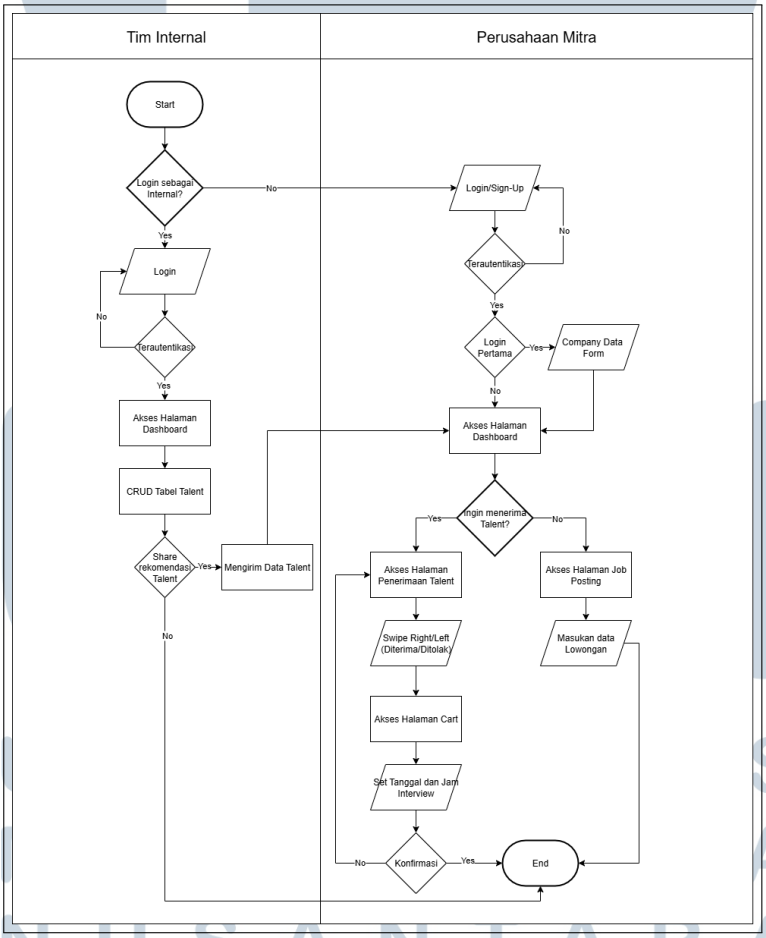
3.3.2 Perancangan Aplikasi

Aplikasi yang dikembangkan terbagi menjadi dua jenis pengguna, yaitu tim internal dan perusahaan mitra. Untuk bagian user tim internal, aplikasi menyediakan sebuah *dashboard* yang menampilkan tabel informasi terkait daftar talenta, dengan kemampuan untuk melakukan operasi CRUD *(Create, Read, Update, Delete)*. Dashboard ini juga menampilkan jumlah talenta, jumlah lowongan, jumlah perusahaan mitra, serta tingkat penerimaan talenta. Selain itu, terdapat tabel yang menampilkan daftar talenta yang telah direkomendasikan kepada perusahaan mitra, yang dapat difilter berdasarkan perusahaan. tim internal juga dapat melihat jadwal wawancara talenta yang telah ditentukan oleh perusahaan mitra.

Untuk bagian perusahaan mitra, aplikasi menyediakan *dashboard* yang menampilkan daftar talenta yang direkomendasikan, serta jadwal wawancara talenta yang telah ditentukan. Selain itu, terdapat halaman *Job Posting* yang memungkinkan perusahaan untuk memposting lowongan magang, serta halaman Penerimaan Talent dengan sistem *swipe*, yang memudahkan perusahaan dalam memilih talenta yang sesuai. Terdapat juga halaman *Cart* yang digunakan untuk konfirmasi akhir sebelum menentukan jadwal wawancara dengan talenta yang telah dipilih. Aplikasi ini dirancang untuk mempermudah interaksi antara tim internal dan perusahaan mitra dalam proses rekrutmen talenta. Adapun pengembangan terhadap aplikasi tersebut dapat dibagi dengan detail sebagai berikut:

A Perancangan Flowchart

Flowchart yang digunakan adalah *flowchart vertical swimlane*, untuk menggambarkan hubungan antar pengguna dalam sistem. Flowchart ini memberikan gambaran alur aplikasi secara keseluruhan, dimulai dari proses *login* yang dilakukan oleh setiap jenis pengguna, baik tim internal maupun Perusahaan Mitra, hingga konfirmasi penerimaan hasil rekomendasi talent. Setiap pengguna memiliki jalur alurnya masing-masing yang terpisah, memudahkan visualisasi langkah-langkah yang dilakukan oleh masing-masing pihak. Dalam flowchart ini, dijelaskan bagaimana data dikirim, diproses, dan akhirnya menghasilkan interaksi yang efisien antara tim internal dan perusahaan mitra, mulai dari akses dashboard hingga tahap konfirmasi penerimaan talent seperti pada gambar 3.1.



Gambar 3.1. Flowchart Aplikasi Dashboard Secara Keseluruhan

Proses pada gambar 3.1 dimulai dengan tim internal yang melakukan *login* ke dalam sistem. Setelah *login* berhasil, mereka dapat mengakses halaman *dashboard* yang menyajikan berbagai informasi penting terkait manajemen data

talent. Di halaman *dashboard*, tim internal memiliki akses penuh untuk melakukan operasi CRUD pada tabel talent. Dengan fitur ini, tim internal dapat menambahkan, memperbarui, atau menghapus data talent yang terdaftar dalam sistem, memastikan data yang dikelola tetap akurat dan *up-to-date*.

Setelah melakukan pengelolaan data talent, tim internal dapat membagikan rekomendasi talent kepada perusahaan mitra. Jika data talent sudah lengkap dan sesuai dengan kriteria, sistem akan memungkinkan pengiriman data tersebut. Jika ada kesalahan dalam data atau talent tidak memenuhi kriteria, sistem akan menonaktifkan *button*. Proses ini memastikan bahwa rekomendasi yang diberikan kepada perusahaan mitra adalah valid dan dapat diproses lebih lanjut.

Proses pada perusahaan mitra dimulai ketika perusahaan mitra melakukan *login* atau *sign-up* untuk mengakses sistem. Jika perusahaan baru, mereka diwajibkan untuk mengisi formulir data perusahaan terlebih dahulu sebelum bisa melanjutkan ke proses berikutnya. Setelah berhasil *login*, perusahaan mitra dapat mengakses halaman *dashboard* yang menyediakan informasi terkait aktivitas mereka dalam sistem, termasuk data terkait talent yang telah dibagikan oleh tim internal dan jadwal *interview*. Sebelum talent dapat direkomendasikan, perusahaan mitra harus terlebih dahulu memposting lowongan pekerjaan. *Job post* ini memberikan informasi terkait posisi yang tersedia dalam perusahaan dan menjadi dasar bagi tim internal untuk merekomendasikan talent yang sesuai.

Di halaman penerimaan talent, perusahaan mitra dapat melakukan seleksi terhadap talent yang diterima atau ditolak dengan cara *swipe* ke kanan atau kiri. Setelah memilih talent yang diterima, perusahaan mitra akan diarahkan ke halaman *cart* untuk memverifikasi pemilihan tersebut dan melanjutkan ke langkah selanjutnya. Selanjutnya, mereka dapat mengatur tanggal dan waktu *interview* untuk talent yang diterima. Proses diakhiri dengan konfirmasi penerimaan hasil rekomendasi talent, memastikan bahwa langkah-langkah yang diambil sesuai dengan kebutuhan perusahaan mitra.

B Perancangan Database

Struktur *database* yang digunakan dalam aplikasi ini dirancang untuk mengelola berbagai entitas yang saling berhubungan, termasuk tabel-tabel utama seperti Auth, Profile, Talent, Company, JobPosition, dan Roles, yang berfungsi untuk menyimpan dan mengatur informasi terkait pengguna, perusahaan, posisi pekerjaan, serta proses rekrutmen dan wawancara. Setiap tabel memiliki relasi

yang terstruktur dengan baik, yang memastikan data dapat diakses, dikelola, dan diperbarui secara efisien, mendukung operasional sistem dalam memberikan layanan yang optimal kepada pengguna dan perusahaan. Sistem ini juga dilengkapi dengan log pencatatan berbagi profil (ShareProfileLog) yang memungkinkan pelacakan aktivitas berbagi informasi antar pengguna. Berikut penjelasan yang lebih merinci mengenai kegunaan tiap tabel pada *database* gambar 3.2:

1. Tabel Auth berfungsi untuk menyimpan data otentikasi pengguna seperti email, username, dan kata sandi yang telah dienkripsi. Tabel ini memastikan bahwa setiap pengguna dapat melakukan *login* ke dalam sistem dengan kredensial yang valid. Tabel ini menjadi tabel utama dalam proses autentikasi pengguna di aplikasi.

2. Tabel Profile menyimpan informasi detail tentang pengguna, seperti nama lengkap dan role. Tabel ini mengelola data pribadi pengguna yang lebih mendalam dan berfungsi untuk menghubungkan informasi otentikasi pengguna yang ada di tabel Auth dan menghubungkan informasi role pengguna.

3. Tabel Roles mendefinisikan peran atau level akses yang dimiliki pengguna dalam sistem. Misalnya, pengguna dapat memiliki peran sebagai internal atau perusahaan mitra. Tabel ini menyimpan informasi mengenai hak akses yang dimiliki setiap pengguna berdasarkan peran yang diberikan.

4. Tabel Talent berfungsi sebagai katalog atau basis data utama para kandidat. Berbeda dengan pengguna, talenta adalah entitas data pasif yang informasi lengkapnya diinput, diperbarui, dan dikelola oleh pengguna tim internal.

5. Tabel Company berfungsi untuk menyimpan data tentang perusahaan yang terdaftar dalam sistem. Ini termasuk nama perusahaan, alamat, industri, dan informasi lainnya. Tabel ini digunakan untuk mengelola profil perusahaan yang berpartisipasi dalam proses rekrutmen atau perekrutan talent.

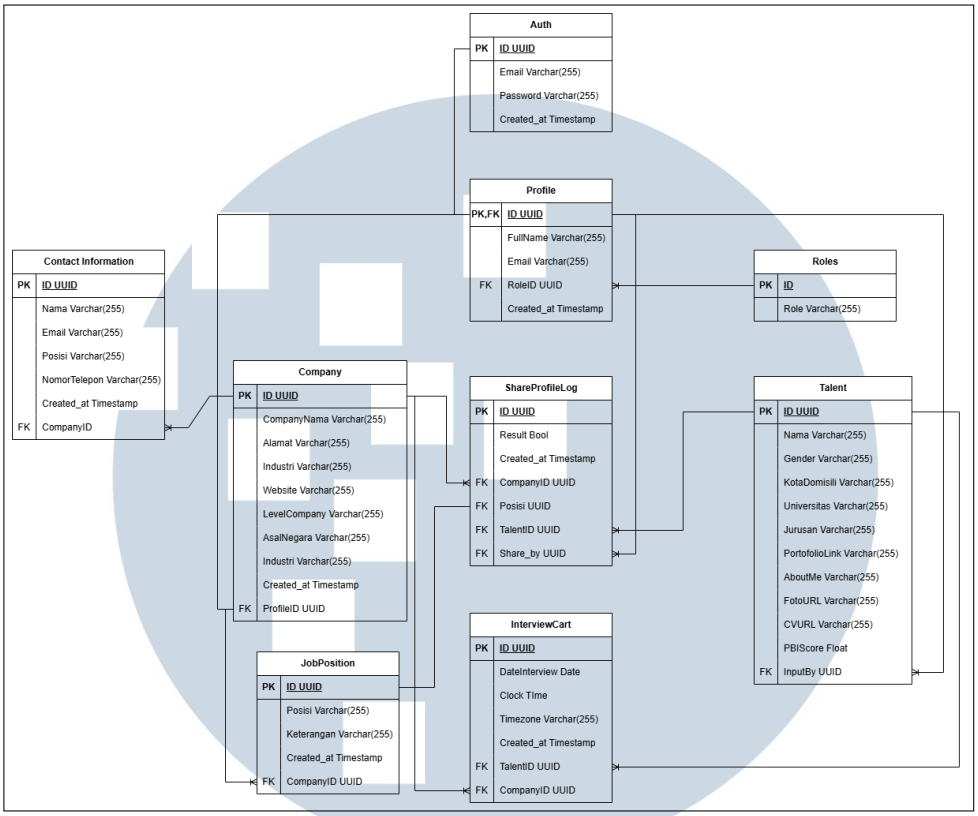
6. Tabel Contact Information didedikasikan untuk menyimpan data kontak detail dari pengguna di suatu perusahaan mitra. Pemisahan informasi kontak ke dalam tabel tersendiri merupakan praktik normalisasi basis data untuk meningkatkan kerapian dan skalabilitas data.

7. Tabel Job Position menampung semua informasi terkait lowongan pekerjaan yang dipublikasikan oleh perusahaan mitra. Pengguna dari perusahaan mitra dapat membuat dan mengelola data di tabel ini.

8. Tabel Interview *cart* menyimpan data terkait dengan jadwal wawancara yang telah ditentukan antara Talent dan Company. Tabel ini berfungsi untuk mengatur dan melacak status wawancara yang melibatkan kandidat dan perusahaan. Setiap entri dalam tabel *Interview Cart* mencatat tanggal dan jam wawancara.

9. Tabel Share Profile Log berfungsi sebagai buku catatan riwayat untuk melacak proses rekomendasi dan penempatan talenta kepada perusahaan mitra. Setiap kali pengguna internal secara resmi merekomendasikan seorang talenta, sebuah catatan akan dibuat di sini. Secara fungsional, setiap baris data di tabel ini merujuk ke tabel Talent untuk mengetahui kandidat mana yang direkomendasikan, terhubung ke tabel Company untuk mengidentifikasi perusahaan mana yang diberikan rekomendasi, menunjuk ke tabel JobPosition untuk merinci lowongan spesifik yang ditawarkan, dan terhubung ke tabel Profile untuk melacak pengguna internal mana yang merekomendasikan talent tersebut.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.2. *database* Aplikasi

Struktur relasi antar tabel pada aplikasi ini dibangun untuk menjamin tiga pilar utama pengelolaan data: konsistensi, integritas, dan efisiensi. Relasi antar tabel dirancang untuk mendukung interaksi antar entitas dalam sistem, mulai dari pengguna hingga perusahaan, serta proses-proses penting seperti rekrutmen dan wawancara. Relasi-relasi ini memastikan bahwa data dapat diproses dengan baik, mengoptimalkan pengambilan informasi, dan menjaga integritas antara entitas yang saling berhubungan. Berikut ini adalah penjelasan lebih rinci mengenai relasi antar tabel yang ada pada aplikasi ini.

1. Tabel Profile

    - *1 to 1* dengan *Auth*: Setiap entri di tabel profile berhubungan dengan satu entri di tabel auth melalui *foreign key*. Ini menunjukkan bahwa setiap profil pengguna harus memiliki data otentikasi yang terhubung (misalnya, username dan password). Relasi *1 to 1* berarti satu entri di profil hanya dapat berhubungan dengan satu entri di Auth, dan sebaliknya.

- *Many to 1* dengan *Roles*: Setiap profil pengguna memiliki satu *role*, yang diambil dari tabel Roles. Banyak profil pengguna dapat terhubung dengan satu *role*. Relasi *Many to 1* ini menunjukkan bahwa banyak pengguna (profil) dapat memiliki peran yang sama, misalnya sebagai pengguna internal atau perusahaan mitra.

2. Tabel Company

   - *1 to 1* dengan *Profile*: Setiap entri di tabel Company memiliki satu profil pengguna yang mengakses aplikasi dari perusahaan tersebut. Relasi *1 to 1* ini mengindikasikan bahwa perusahaan yang tercatat hanya akan berhubungan dengan satu profil pengguna yang mengaksesnya.

3. Tabel Contact Information

   - *Many to 1* dengan *Company*: Setiap detail kontak di tabel Contact Information berhubungan dengan satu entri di tabel Company. Banyak informasi kontak dapat terhubung dengan satu perusahaan, memungkinkan penyimpanan detail kontak yang lebih dari satu untuk perusahaan yang sama.

4. Tabel Job Position

   - *Many to 1* dengan *Company*: Setiap posisi pekerjaan yang diposting di tabel Job Position berhubungan dengan satu perusahaan. Banyak posisi pekerjaan dapat terhubung ke satu perusahaan, yang menunjukkan bahwa perusahaan yang sama dapat memposting banyak posisi pekerjaan.

5. Tabel Interview Cart

   - *Many to 1* dengan *Talent*: Setiap interview cart berhubungan dengan satu entri di tabel talent. Banyak interview cart dapat dikaitkan dengan satu talent, yang berarti banyak kesempatan *interview* dapat diberikan untuk satu kandidat talent.
   - *Many to 1* dengan table *Company*: Setiap interview cart berhubungan dengan satu entri di tabel Company. Banyak interview cart bisa berhubungan dengan satu perusahaan, menunjukkan bahwa satu perusahaan bisa melakukan banyak interview terhadap talent.

6. Tabel Talent

    - *Many to 1* dengan *Profile*: Setiap entri di tabel talent berhubungan dengan satu profil pengguna internal (misalnya, admin atau HR yang memasukkan data talent). Relasi *Many to 1* ini menunjukkan bahwa banyak talent dapat dimasukkan oleh satu profil internal yang sama.

7. Tabel Share Profile Log

    - *Many to 1* dengan *Talent, Company, Job Position,* dan *Profile*: Setiap log berbagi profil berhubungan dengan satu talent, satu perusahaan, satu posisi pekerjaan, dan satu profil pengguna. Relasi *Many to 1* ini menunjukkan bahwa banyak log berbagi profil dapat terjadi untuk satu talent, satu perusahaan, satu posisi pekerjaan, dan satu profil pengguna yang sama, merekam setiap kali profil talent dibagikan ke perusahaan atau posisi pekerjaan tertentu oleh pengguna tertentu.

C Migrasi database

Proses migrasi data dilakukan untuk memastikan kelancaran transfer informasi terkait seluruh data talent yang sebelumnya disimpan dalam sistem SmartSuite. Mengingat pentingnya keberlanjutan data yang akurat dan terintegrasi, migrasi ini bertujuan untuk menghindari kebutuhan input manual data talent secara satu per satu pada aplikasi baru. Dengan adanya migrasi, efisiensi dalam pengelolaan data dapat terjaga, sekaligus meminimalkan risiko kesalahan input yang dapat terjadi jika dilakukan secara manual.

Integrasi antara SmartSuite dan Supabase dilakukan melalui penggunaan webhook, yang berfungsi sebagai mekanisme pengiriman data secara otomatis setiap kali terjadi perubahan atau pembaruan data di SmartSuite. Dengan mengonfigurasi webhook, data yang ada di SmartSuite dapat dipindahkan secara *real-time* ke Supabase tanpa memerlukan intervensi manual, memastikan bahwa sistem terintegrasi dengan baik dan data tetap terbarui secara konsisten. Supabase kemudian menggunakan Edge Function untuk menerima data dari SmartSuite, yang menjalankan kode kustom secara *serverless*. Edge Function ini memproses data yang diterima melalui webhook, melakukan validasi, transformasi format, atau pengecekan integritas sebelum data disimpan dalam *database* Supabase. Dengan menggunakan Edge Function, proses pengolahan data dapat dilakukan secara

otomatis dan efisien, memastikan bahwa data yang masuk selalu sesuai dengan kebutuhan sistem.

```
import { createClient } from 'jsr:@supabase/supabase-js@2';
const supabase = createClient(Deno.env.get('SUPABASE_URL') ?? '',
    Deno.env.get('SUPABASE_ANON_KEY') ?? '');
Deno.serve(async (req)=>{
  // Parse the incoming JSON request body
  const requestBody = await req.json();
  // Extract the 'name' property from the request body
  const name = requestBody.name;
  const fotoURL = requestBody.fotoURL || "";
  const gender = requestBody.gender; // Expecting boolean value (
    true/false)
  const kotaDomisili = requestBody.kotaDomisili;
  const universitas = requestBody.universitas;
  const jurusan = requestBody.jurusan;
  const cvURL = requestBody.cvURL || "";
  const pbiScore = parseFloat(requestBody.pbiScore) || 0.0; //
    Expecting float value
  const portofolioLink = requestBody.portofolioLink || "";
  const aboutMe = requestBody.aboutme || "";
  if (!name) {
    return new Response(JSON.stringify({
      error: 'Name is required'
    }), {
      status: 400,
      headers: {
        'Content-Type': 'application/json'
      }
    });
  }
  // Insert the data into the GVApp table
  const { data, error } = await supabase.from('GVapp').insert([
    {
      Nama: name,
      FotoURL: fotoURL,
      Gender: gender,
      KotaDomisili: kotaDomisili,
      Universitas: universitas,
      Jurusan: jurusan,
      CVURL: cvURL,
      PBIscore: pbiScore,
      PortofolioLink: portofolioLink,
```

```
      Aboutme: aboutMe
    }
  ]);
  if (error) {
    return new Response(JSON.stringify({
      error: error.message
    }), {
      status: 500,
      headers: {
        'Content-Type': 'application/json'
      }
    });
  }
  return new Response(JSON.stringify({
    data
  }), {
    status: 200,
    headers: {
      'Content-Type': 'application/json'
    }
  });
});
```

Kode 3.1: Kode Webhook Supabase

Kode 3.1 menunjukkan implementasi Edge Function di Supabase yang menerima data dari SmartSuite melalui webhook. Pertama, kode ini mengimpor pustaka createClient dari @supabase/supabase-js untuk membuat koneksi dengan *database* Supabase menggunakan URL dan kunci anonim yang disimpan dalam variabel lingkungan (SUPABASE_URL dan SUPABASE_ANON_KEY). Fungsi Deno.serve kemudian digunakan untuk menangani permintaan HTTP yang masuk. Dalam fungsi ini, data dalam format JSON yang diterima dari SmartSuite diparse untuk mengekstrak berbagai informasi seperti name, fotoURL, gender, kotaDomisili, universitas, jurusan, cvURL, pbiScore, portofolioLink, dan aboutMe.

Sebelum data dimasukkan ke dalam *database*, kode ini memeriksa apakah nama (name) telah diberikan. Jika nama tidak ada, maka sistem akan merespons dengan status 400 dan pesan error. Jika data valid, kode ini melanjutkan untuk memasukkan data tersebut ke dalam tabel GVapp di Supabase menggunakan metode *insert*. Jika terjadi kesalahan saat penyimpanan data, sistem akan mengembalikan respons dengan status 500 beserta pesan error yang sesuai. Jika data berhasil dimasukkan, respons dengan status 200 dan data yang dimasukkan

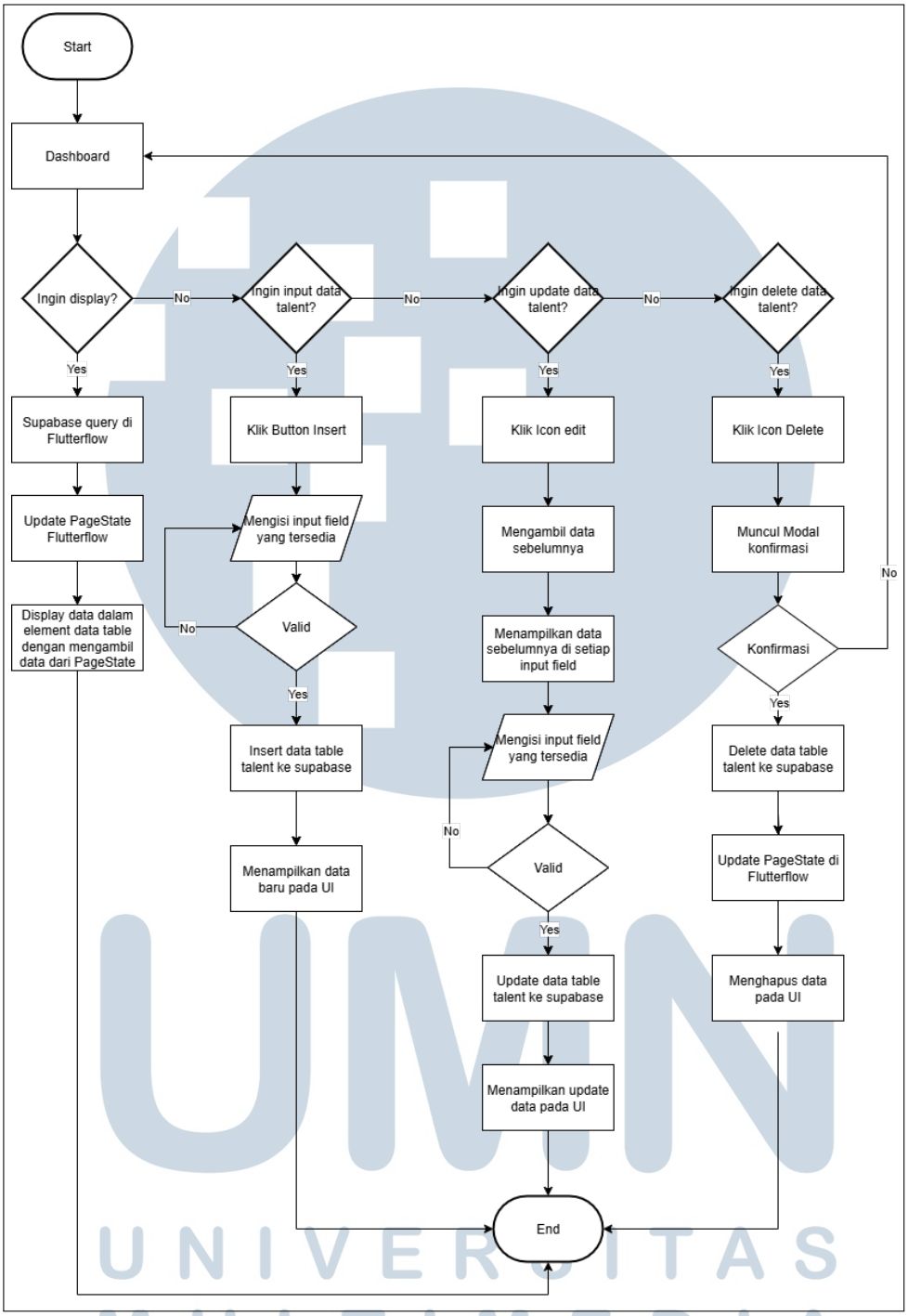
dikembalikan dalam format JSON.

3.3.3 Pengembangan Fitur Aplikasi

A Fitur CRUD Dashboard Internal

dashboard internal dirancang untuk memberikan kemudahan akses dan pengelolaan data talent bagi tim internal. Pengguna dapat melakukan berbagai tindakan melalui fitur yang disediakan, termasuk operasi CRUD (Create, Read, Update, Delete) pada tabel talent. Fitur ini memungkinkan tim internal untuk menambahkan data talent baru, memperbarui informasi yang sudah ada, atau menghapus data talent yang tidak relevan. Selain itu, *dashboard* juga menyediakan fitur untuk share profile talent, yang memungkinkan tim internal untuk membagikan profil talent yang sudah diseleksi kepada perusahaan mitra untuk dipertimbangkan lebih lanjut. Di dalam sistem, status dari setiap talent yang telah direkomendasikan kepada perusahaan mitra disimpan di tabel terpisah yang dapat di filter berdasarkan perusahaan. Tabel ini mencatat hasil dari proses rekomendasi, seperti apakah talent diterima atau ditolak oleh perusahaan mitra. *dashboard* internal tidak hanya berfungsi sebagai alat untuk mengelola data talent, tetapi juga sebagai *platform* untuk memantau dan mengelola hasil rekomendasi talent yang telah diberikan kepada perusahaan mitra.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.3. Flowchart Proses CRUD pada *dashboard* Internal

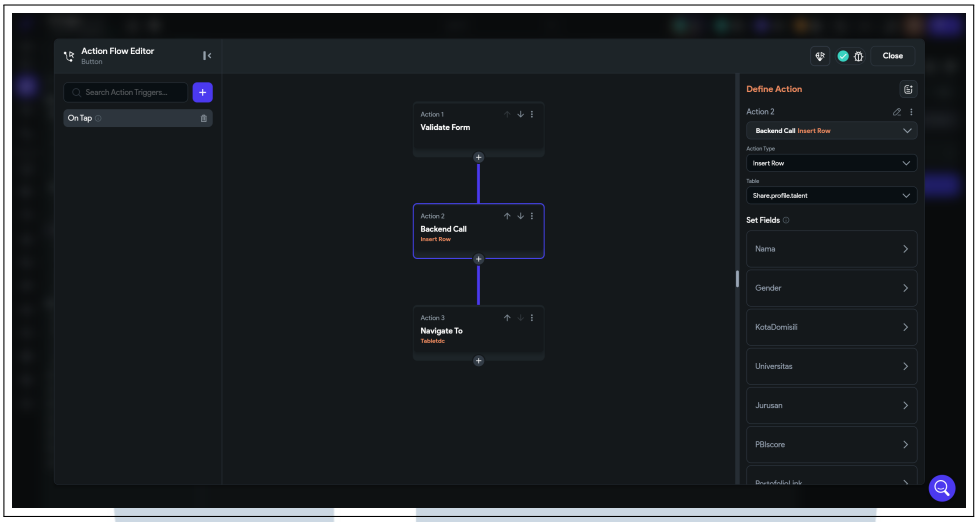
Proses read pada gambar 3.3, sistem akan melakukan query ke *database* Supabase untuk mengambil data yang diperlukan. Proses dimulai dengan menggunakan Supabase query di Flutterflow untuk menarik data yang sudah ada dalam tabel talent. Setelah data berhasil diambil, PageState akan diupdate di Flutterflow. PageState adalah mekanisme yang digunakan untuk menyimpan dan

mengelola status tampilan data di Flutterflow. Karena Flutterflow tidak secara otomatis memuat ulang halaman setiap kali ada perubahan data, penggunaan PageState memungkinkan tampilan antarmuka pengguna untuk terupdate secara *real-time*, menampilkan data terbaru yang sudah diambil dari Supabase ke dalam elemen data tabel.

Proses create pada gambar flowchart 3.3, dimulai ketika pengguna mengklik Button *insert*. Pengguna akan mengisi input field yang tersedia dengan informasi yang diperlukan. Setelah pengguna mengisi data, sistem akan melakukan pengecekan validitas input dengan memastikan bahwa semua field yang wajib diisi telah terisi dengan benar. Jika data valid, maka proses *insert* data table talent ke Supabase akan dijalankan untuk menyimpan data ke dalam tabel talent di Supabase. Setelah data berhasil disimpan, UI akan menampilkan data baru yang telah dimasukkan.

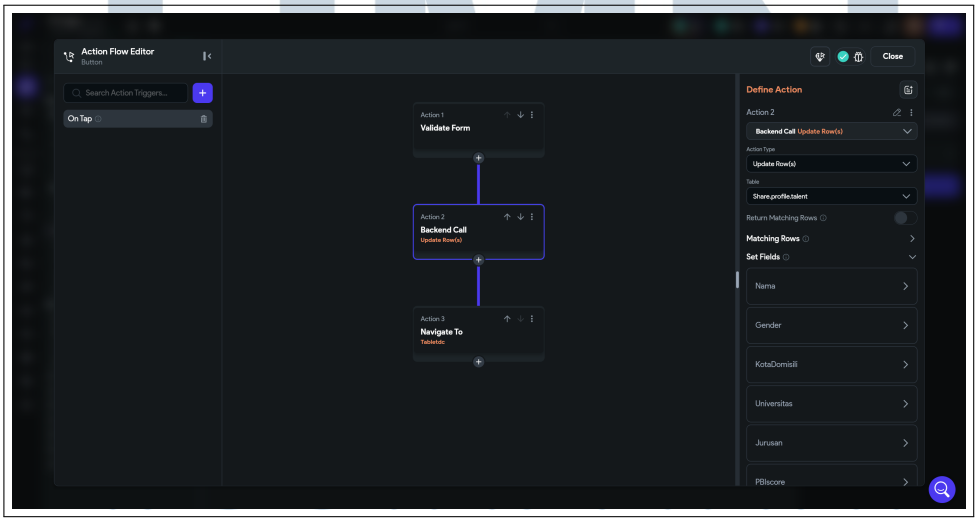
Proses *update* pada gambar flowchart 3.3 data dimulai ketika pengguna mengklik Icon Edit. Sistem akan mengambil data sebelumnya dari *database* dan menampilkannya di input field yang sesuai. Pengguna kemudian dapat mengedit data yang ingin diperbarui. Setelah data diperbarui, sistem akan memverifikasi validitas data yang dimasukkan. Jika valid, proses *update* data table talent ke Supabase akan dijalankan, memperbarui data di *database* Supabase. Setelah pembaruan selesai, tampilan UI akan otomatis menunjukkan data yang telah diupdate.

Proses penghapusan data pada gambar flowchart 3.3, dimulai ketika pengguna mengklik Icon Delete. Sebuah Modal konfirmasi akan muncul untuk memastikan bahwa pengguna ingin menghapus data yang dipilih. Setelah konfirmasi, sistem akan menghapus data dari tabel talent di Supabase dengan menggunakan perintah Delete data table talent ke Supabase. Setelah penghapusan selesai, PageState akan diupdate untuk menghapus tampilan data yang sudah dihapus dari UI secara *real-time*. Karena Flutterflow tidak memuat ulang halaman secara otomatis, PageState memastikan bahwa perubahan data langsung tercermin di UI tanpa memerlukan refresh manual pada halaman.



Gambar 3.4. *Action Flow* Create FlutterFlow

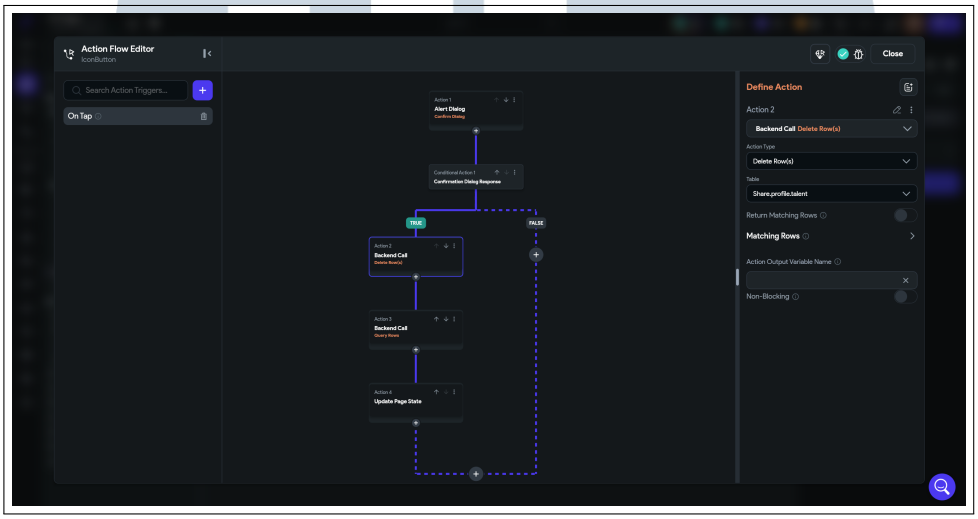
ACtion flow pada gambar 3.4 merupakan urutan langkah-langkah yang dibuat dalam aplikasi FlutterFlow sebagai *logic backend* ketika tombol ditekan. Pertama, sistem akan memvalidasi *form* yang diisi oleh pengguna. Selanjutnya, jika validasi berhasil, data pengguna yang tercatat akan dimasukkan ke dalam tabel di *backend* melalui sebuah *backend call*. Proses ini melibatkan pengisian beberapa field seperti Nama, Gender, KotaDomisili, Universitas, Jurusan, PBIscore, dan lain-lain. Setelah itu, aplikasi akan mengarahkan pengguna ke halaman *dashboard* yang disebut ”Tabletdc.” Proses ini menunjukkan serangkaian tindakan otomatis yang dirancang untuk memproses data dan navigasi dalam aplikasi berbasis FlutterFlow.



Gambar 3.5. *Action Flow Update* FlutterFlow

Action flow pada gambar 3.5 merupakan urutan langkah-langkah yang dibuat dalam aplikasi FlutterFlow sebagai *logic backend* ketika tombol ditekan.

Pertama, sistem akan memvalidasi *form* yang diisi oleh pengguna. Selanjutnya, jika validasi berhasil, data pengguna yang tercatat akan diperbarui dalam tabel di *backend* melalui sebuah *backend call* untuk memperbarui baris yang sesuai. Proses ini melibatkan pengisian beberapa field seperti Nama, Gender, KotaDomisili, Universitas, Jurusan, PBIscore, dan lain-lain. Setelah itu, aplikasi akan mengarahkan pengguna ke halaman *dashboard* yang disebut "Tabletdc." Proses ini menunjukkan serangkaian tindakan otomatis yang dirancang untuk memperbarui data dan navigasi dalam aplikasi berbasis FlutterFlow.



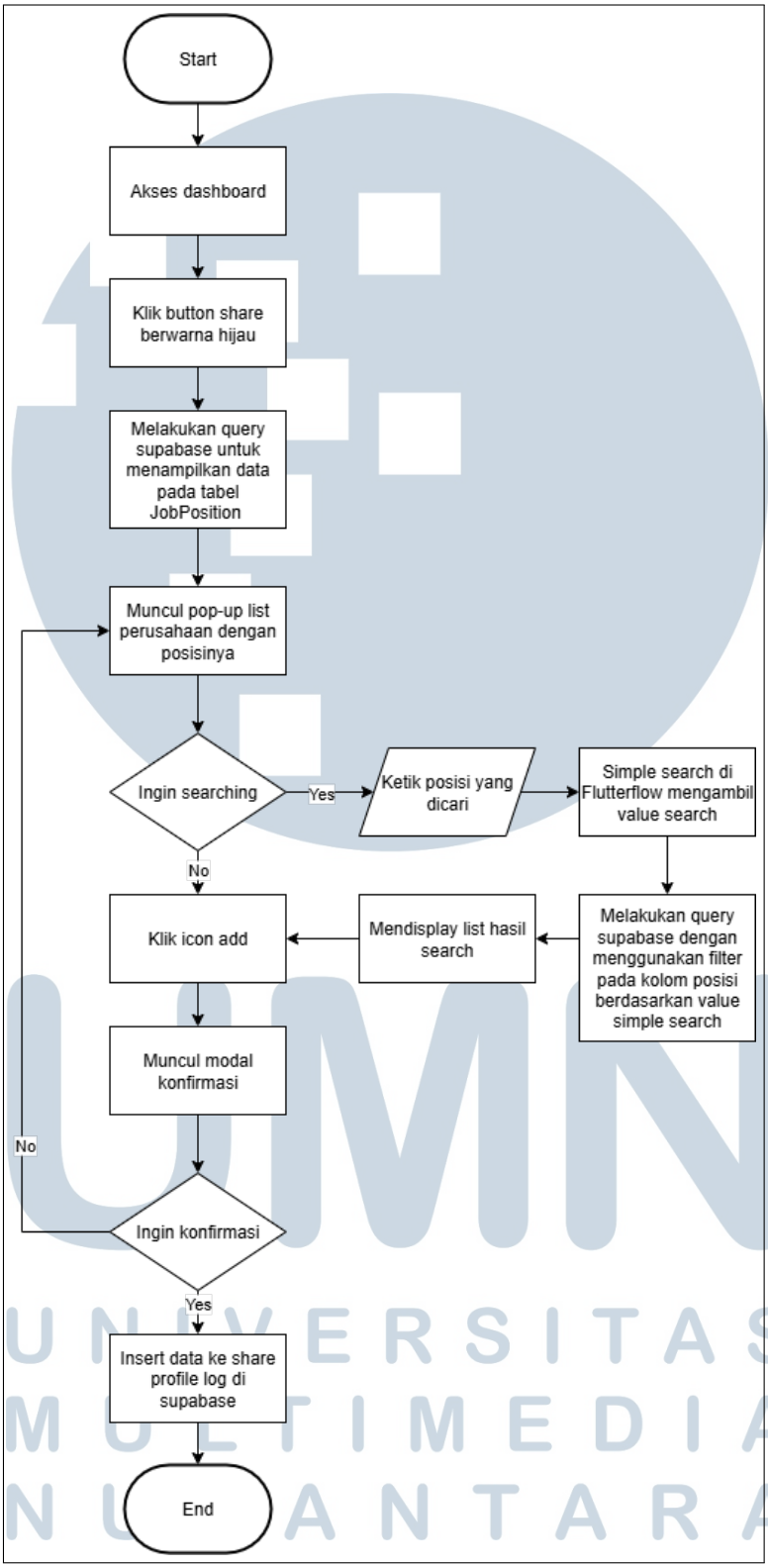
Gambar 3.6. *Action Flow Update* FlutterFlow

Action flow pada gambar 3.6 menggambarkan urutan langkah-langkah yang dibuat dalam aplikasi FlutterFlow untuk menangani aksi penghapusan data. Pertama, sistem akan menampilkan dialog konfirmasi kepada pengguna untuk memastikan bahwa mereka ingin melanjutkan penghapusan data. Jika pengguna mengonfirmasi, data yang sesuai akan dihapus dari tabel di *backend*. Setelah penghapusan, untuk memastikan tampilan antarmuka pengguna (UI) diperbarui sesuai dengan perubahan data, sistem akan mengupdate state halaman. Ini akan memicu pembaruan UI agar mencerminkan status terbaru, seperti menghilangkan data yang sudah dihapus dari daftar yang ditampilkan. Jika pengguna membatalkan penghapusan, proses penghapusan tidak akan dilanjutkan dan tampilan tetap seperti semula. Proses ini memastikan bahwa penghapusan data dilakukan dengan konfirmasi pengguna dan UI selalu terbarui dengan data yang relevan.

B Fitur Merekomendasikan Talent Dashboard Internal

Fitur Merekomendasikan Talent seperti pada gambar 3.9 memungkinkan tim internal untuk memilih dan merekomendasikan talent kepada perusahaan mitra dengan mudah. Pengguna cukup mengklik button share berwarna hijau pada baris talent yang ingin dibagikan. Setelah itu, akan muncul pop-up rekomendasi yang menampilkan daftar perusahaan yang dapat menerima rekomendasi talent tersebut. Di dalam pop-up, pengguna dapat memilih perusahaan yang tepat untuk merekomendasikan talent dengan mengklik icon add di samping nama perusahaan. Setelah talent direkomendasikan, mereka akan masuk ke dalam daftar rekomendasi yang dapat dipantau lebih lanjut. Fitur ini mempermudah proses seleksi dan rekomendasi talent, sekaligus memberikan fleksibilitas kepada pengguna dalam mencari dan memilih kandidat terbaik untuk perusahaan mitra.

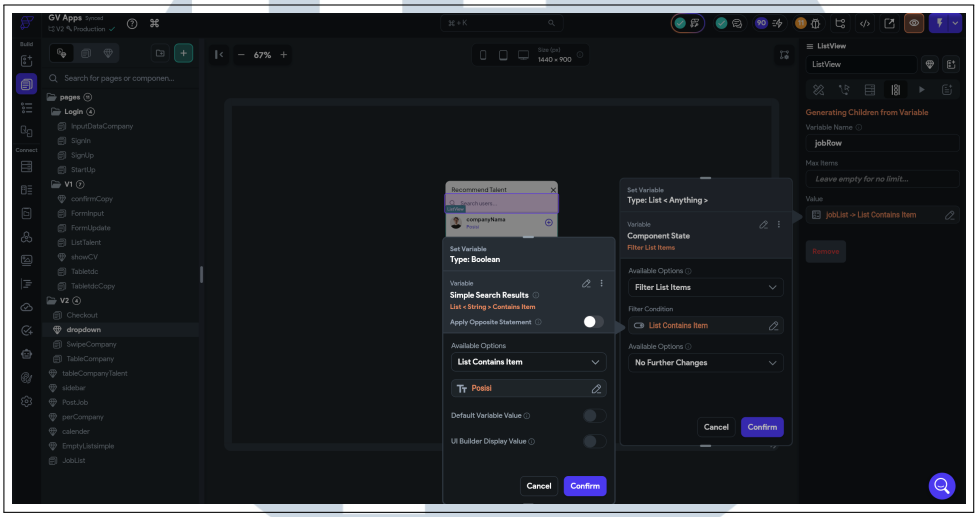
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.7. Flowchart Fitur Share Talent

Proses pada gambar 3.7 dimulai ketika pengguna mengakses *dashboard*

dan mengklik button share berwarna hijau pada baris talent yang ingin dibagikan. Setelah itu, sistem melakukan query ke Supabase untuk menampilkan data yang ada pada tabel JobPosition, yang berisi informasi posisi yang tersedia di perusahaan mitra. Data ini kemudian ditampilkan dalam bentuk pop-up list yang menunjukkan perusahaan-perusahaan beserta posisi yang tersedia.

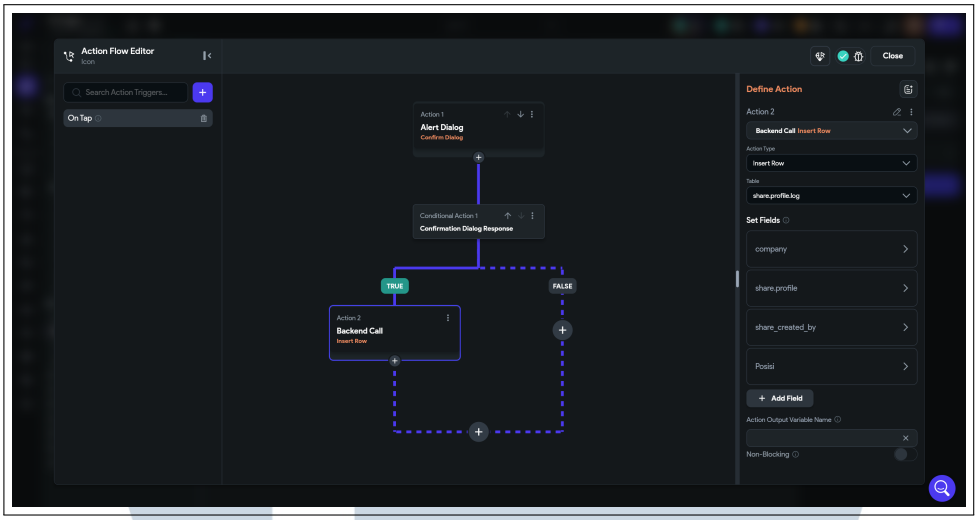


Gambar 3.8. Query untuk Menampilkan Hasil Search

Pada tahap ini, pengguna diberikan opsi untuk search posisi yang diinginkan. Jika pengguna tidak ingin mencari posisi tertentu, mereka dapat langsung mengklik ikon *add* di samping nama perusahaan. Setelah itu, modal konfirmasi akan muncul untuk meminta pengguna memastikan keputusan mereka sebelum melanjutkan proses rekomendasi. Namun, jika pengguna ingin melakukan pencarian, mereka dapat ketik posisi yang dicari, dan sistem menyimpan hasil input pengguna yang dilakukan oleh simple search di Flutterflow. Sistem akan melakukan query ke Supabase dengan menggunakan filter pada kolom posisi pada tabel JobPosition berdasarkan value yang disimpan disimple search seperti pada gambar 3.8 dan hasil pencarian kemudian akan ditampilkan.

Jika pengguna sudah yakin dengan posisi yang dipilih, mereka dapat mengonfirmasi pilihan mereka dengan mengklik konfirmasi pada modal yang muncul. Setelah itu, data akan dimasukkan ke dalam tabel log share profile di Supabase, yang mencatat bahwa talent tersebut telah direkomendasikan ke perusahaan mitra. Proses ini berakhir dengan tindakan memasukkan data ke Supabase dan mengakhiri alur rekomendasi. Proses ini dirancang untuk memastikan bahwa setiap langkah rekomendasi talent dilakukan secara terstruktur, dengan kontrol pencarian dan konfirmasi untuk menghindari kesalahan dalam

memberikan rekomendasi kepada perusahaan mitra.



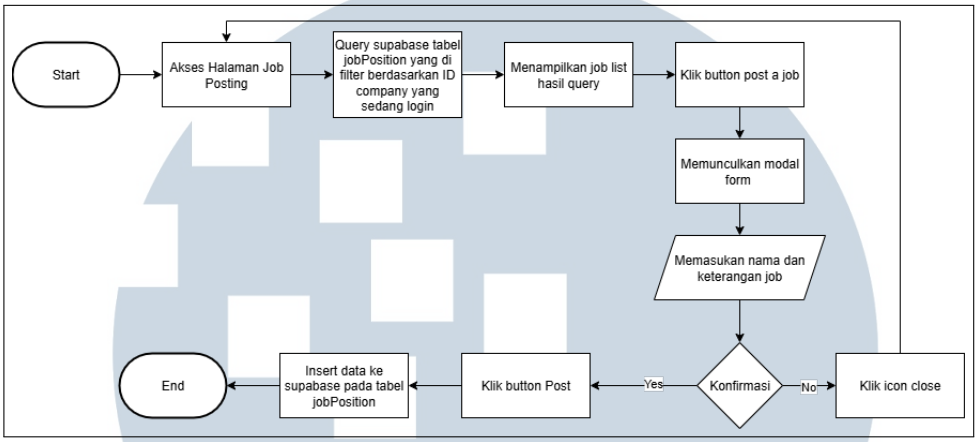
Gambar 3.9. *Action Flow* Merekomendasikan Talent FlutterFlow

Action flow pada gambar 3.9 menggambarkan urutan langkah-langkah yang dibuat dalam aplikasi FlutterFlow untuk menangani aksi berdasarkan konfirmasi pengguna. Pertama, aplikasi akan menampilkan dialog konfirmasi yang meminta persetujuan pengguna untuk melanjutkan suatu tindakan. Jika pengguna mengonfirmasi, maka data akan dimasukkan ke dalam tabel *backend* share.profile.log menggunakan perintah *insert* Row. Data yang dimasukkan mencakup field seperti company, share.profile, share_created_by, dan Posisi. Jika pengguna membatalkan konfirmasi, proses ini tidak akan dilanjutkan, dan tidak ada perubahan yang dilakukan pada *backend*. Proses ini memastikan bahwa data hanya dimasukkan setelah mendapatkan konfirmasi dari pengguna, menjaga akurasi dan kontrol terhadap data yang dimasukkan ke dalam sistem.

C Fitur Job Posting Perusahaan Mitra

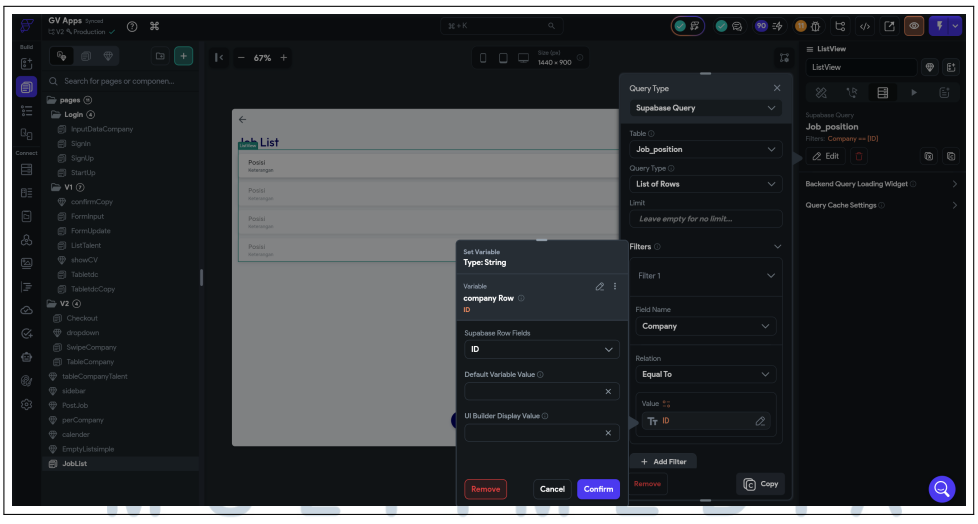
Fitur *Job Posting* memungkinkan perusahaan mitra untuk mengelola dan memposting lowongan pekerjaan dengan mudah. Pada halaman ini, perusahaan mitra dapat melihat list job yang sudah diposting, yang mencakup posisi dan deskripsi pekerjaan yang telah disediakan sebelumnya. Selain itu, perusahaan mitra juga dapat memasukkan job baru dengan mengisi dua kolom utama, yaitu nama posisi dan deskripsi pekerjaan. Untuk menambahkan lowongan pekerjaan baru, pengguna hanya perlu mengklik button *post* setelah memasukkan informasi yang diperlukan dalam kolom Enter job title here untuk nama posisi dan Enter job

description here untuk deskripsi pekerjaan. Setelah dikirim, informasi lowongan pekerjaan baru tersebut akan terdaftar dan akan tampil pada job list.



Gambar 3.10. Flowchart Fitur *Job Posting* Perusahaan Mitra

Proses pada gambar 3.10 dimulai ketika perusahaan mitra mengakses halaman *Job Posting*. Pada halaman ini, sistem akan query ke tabel jobPosition di Supabase, yang difilter berdasarkan ID perusahaan yang sedang *login* seperti pada gambar 3.11. Hasil dari query ini adalah daftar job list yang sudah diposting oleh perusahaan tersebut. Daftar pekerjaan yang ada akan ditampilkan di halaman Job Posting.

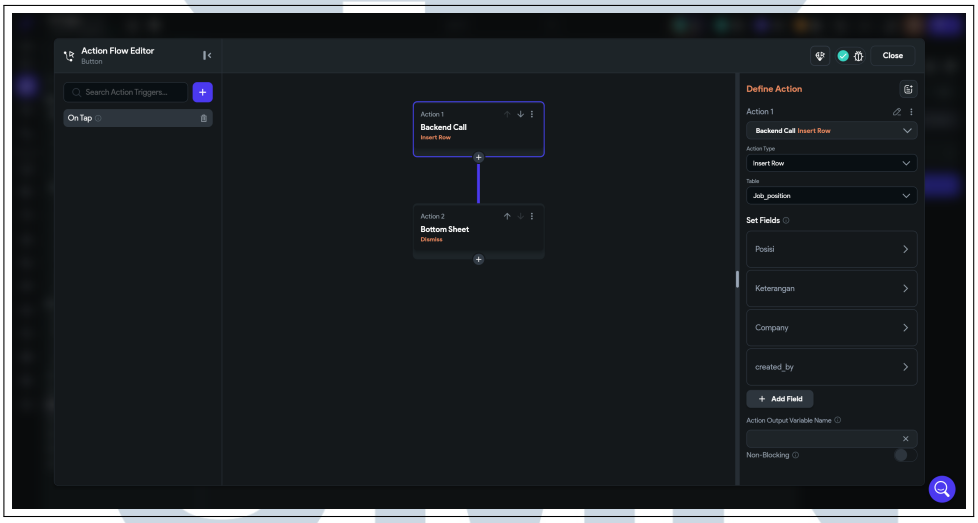


Gambar 3.11. Query untuk Menampilkan Hasil Search

Setelah itu, perusahaan mitra dapat memilih untuk klik button ”*post a job*” untuk menambahkan lowongan pekerjaan baru. Ketika tombol ini diklik, sebuah modal form akan muncul di layar. Di *form* tersebut, perusahaan mitra harus memasukkan nama posisi yang ingin diposting dan deskripsi pekerjaan yang

terkait dengan posisi tersebut. Setelah memasukkan informasi yang diperlukan, perusahaan mitra dapat memilih untuk mengonfirmasi postingan dengan mengklik button *post*.

Namun, sebelum data dimasukkan, sistem meminta konfirmasi dari perusahaan mitra untuk memastikan bahwa informasi yang dimasukkan sudah benar. Jika perusahaan mitra setuju, maka data lowongan pekerjaan akan di*insert* ke dalam tabel jobPosition di Supabase. Jika perusahaan mitra memutuskan untuk membatalkan, mereka dapat klik icon close untuk menutup form dan membatalkan tindakan tersebut. Setelah data berhasil dimasukkan, proses ini berakhir dengan lowongan pekerjaan baru yang terdaftar di sistem dan dapat dilihat oleh perusahaan mitra itu sendiri dan *dashboard* internal ketika ingin melakukan rekomendasi talent. alur proses *Job Posting* ini memastikan bahwa setiap lowongan pekerjaan yang diposting oleh perusahaan mitra terkelola dengan baik dan dapat dipantau secara langsung dalam sistem.



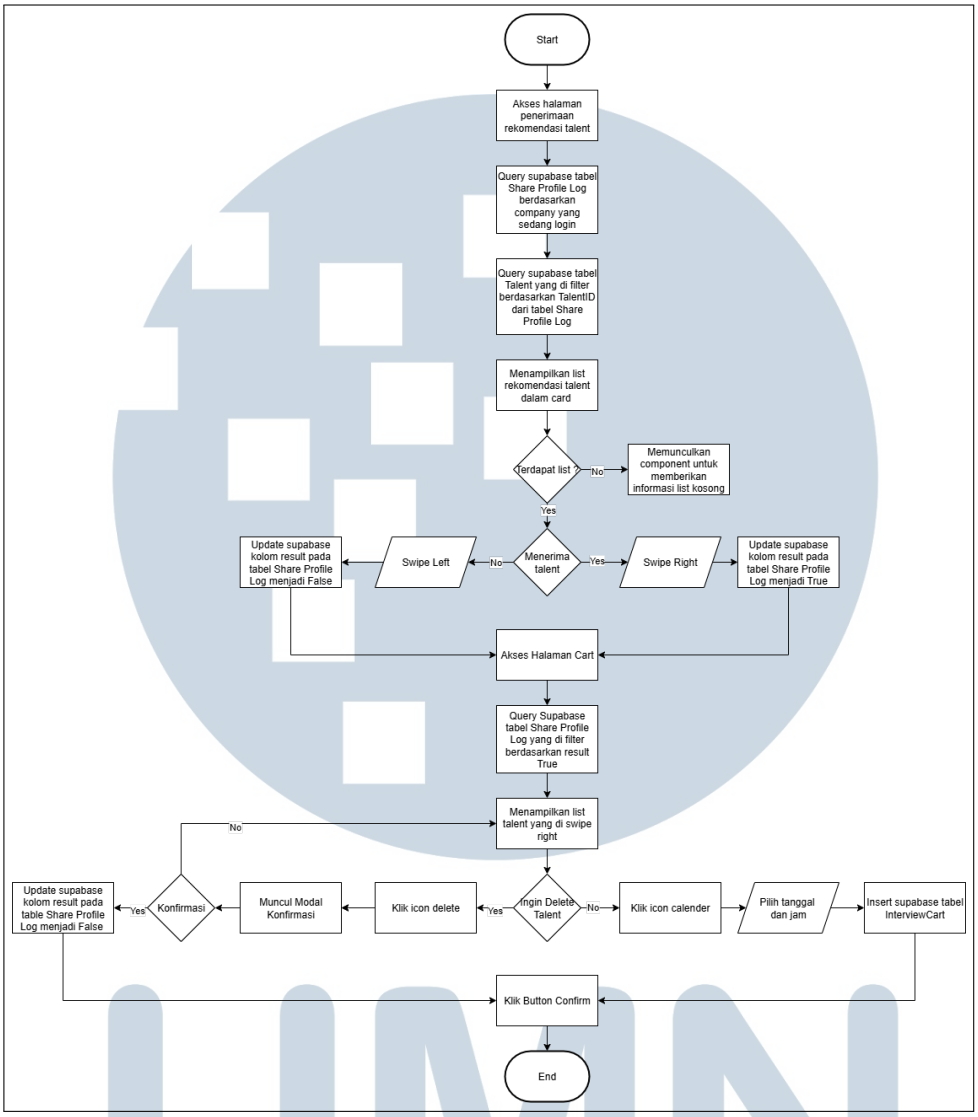
Gambar 3.12. Action Flow *Job Posting* FlutterFlow

Action flow pada gambar 3.12 menggambarkan urutan langkah-langkah yang dibuat dalam aplikasi FlutterFlow ketika tombol ditekan. Pertama, data terkait posisi pekerjaan dimasukkan ke dalam tabel Job_position di *backend* dengan menggunakan perintah *insert Row*. Field yang dimasukkan mencakup Posisi, Keterangan, Company, dan created_by. Setelah data berhasil dimasukkan, aplikasi akan menutup bottom sheet, yang merupakan elemen UI yang sebelumnya terbuka untuk kembali ke halaman sebelumnya. Proses ini memungkinkan aplikasi untuk menambahkan data ke *backend* dan kemudian menyembunyikan elemen UI, memberikan pengalaman pengguna yang mulus dan efisien.

D Fitur Penerimaan Talent Perusahaan Mitra dengan Swipe

Fitur *swipe* memungkinkan perusahaan mitra untuk dengan mudah menerima atau menolak rekomendasi talent yang diberikan oleh tim internal. Dalam fitur ini, perusahaan mitra dapat melihat profil talent yang direkomendasikan dan melakukan *swipe right* untuk menerima talent tersebut, atau *swipe left* untuk menolaknya. Proses ini sangat memudahkan dalam seleksi awal, memberikan pengalaman yang cepat dan efisien. Setelah perusahaan mitra menyelesaikan seleksi dan daftar rekomendasi talent kosong, mereka akan diarahkan untuk mengakses halaman *cart*. Di halaman *cart*, perusahaan mitra dapat menentukan tanggal dan jam *interview* untuk talent yang diterima. Setelah menentukan jadwal *interview*, perusahaan mitra akan mengonfirmasi penerimaan talent untuk melanjutkan ke tahap *interview*, memastikan proses seleksi berjalan lancar dan terorganisir.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.13. Flowchart Penerimaan Talent Perusahaan Mitra dengan *Swipe*

Proses pada gambar 3.13 dimulai ketika perusahaan mitra mengakses halaman penerimaan rekomendasi talent. Sistem kemudian query ke tabel Share Profile Log di Supabase, yang memfilter data berdasarkan perusahaan yang sedang *login*. Selanjutnya, sistem juga akan query ke tabel Talent, yang akan difilter berdasarkan TalentID dari tabel Share Profile Log untuk mengambil data dan menampilkan informasi detail Talent karena di tabel Share Profile Log hanya terdapat TalentID. Hasil dari query ini akan menampilkan list rekomendasi talent dalam bentuk card yang dapat di-*swipe* oleh perusahaan mitra.

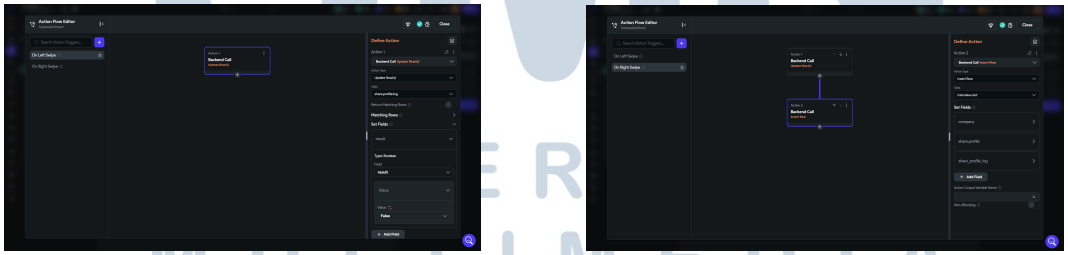
Setelah list talent ditampilkan, jika perusahaan mitra tidak ingin menerima talent tersebut, mereka dapat *swipe left* untuk menolaknya. Jika talent diterima, perusahaan mitra dapat *swipe right*. Ketika talent diterima, sistem akan *update*

kolom result di tabel Share Profile Log menjadi *True*, yang menunjukkan bahwa talent tersebut telah diterima oleh perusahaan mitra. Ketika talent ditolak, sistem akan *update* kolom result di tabel Share Profile Log menjadi *False*. Jika list kosong, sistem akan menampilkan komponen untuk memberi informasi bahwa tidak ada talent yang tersisa untuk dipilih.

Setelah talent diterima, perusahaan mitra akan diarahkan untuk mengakses halaman *cart*. Di halaman *cart*, sistem akan query ke tabel Share Profile Log untuk menampilkan semua talent yang telah diterima (yang memiliki result *True*). Perusahaan mitra kemudian dapat memilih tanggal dan waktu *interview* melalui fitur kalender dan jam di Flutterflow.

Namun, jika perusahaan mitra memutuskan untuk menghapus talent yang sudah ada di dalam *cart*, mereka dapat mengklik icon delete di samping talent yang dipilih. Setelah klik, sistem akan menampilkan konfirmasi penghapusan untuk memastikan keputusan tersebut. Jika perusahaan mitra mengonfirmasi, talent tersebut akan dihapus dari *cart*, dan perubahan tersebut akan diupdate ke dalam tabel Share Profile Log dengan result *False*, yang menandakan bahwa talent tersebut tidak diterima lagi untuk tahap *interview*.

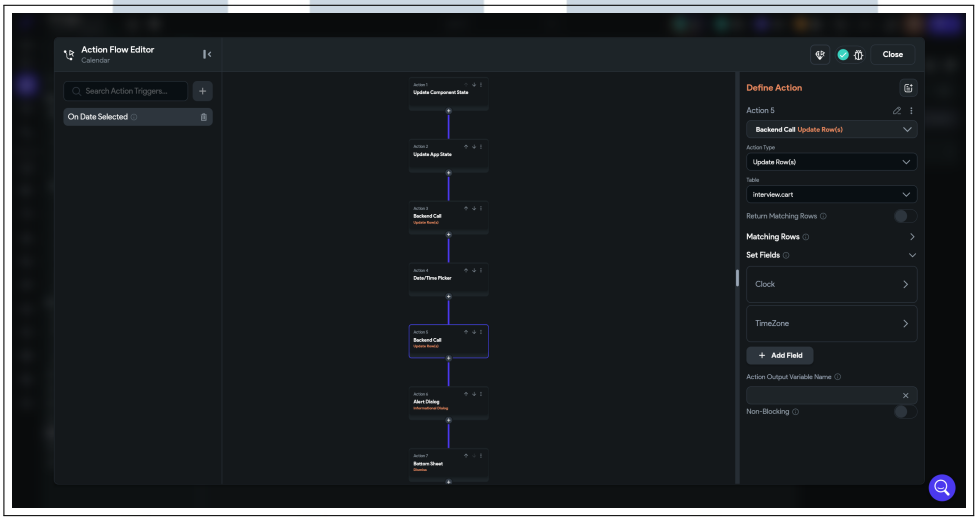
Jika perusahaan mitra tetap ingin melanjutkan, mereka dapat klik button *confirm* untuk menyelesaikan pengaturan *interview*. Proses berakhir setelah data dimasukkan ke dalam tabel InterviewCart di Supabase, yang mencatat jadwal *interview* dan status talent yang diterima. alur ini memungkinkan perusahaan mitra untuk tidak hanya menerima dan mengonfirmasi talent, tetapi juga memberikan fleksibilitas untuk menghapus talent yang tidak diinginkan dari *cart* sebelum melanjutkan ke tahap *interview*.



Gambar 3.14. Action Flow Swipe FlutterFlow

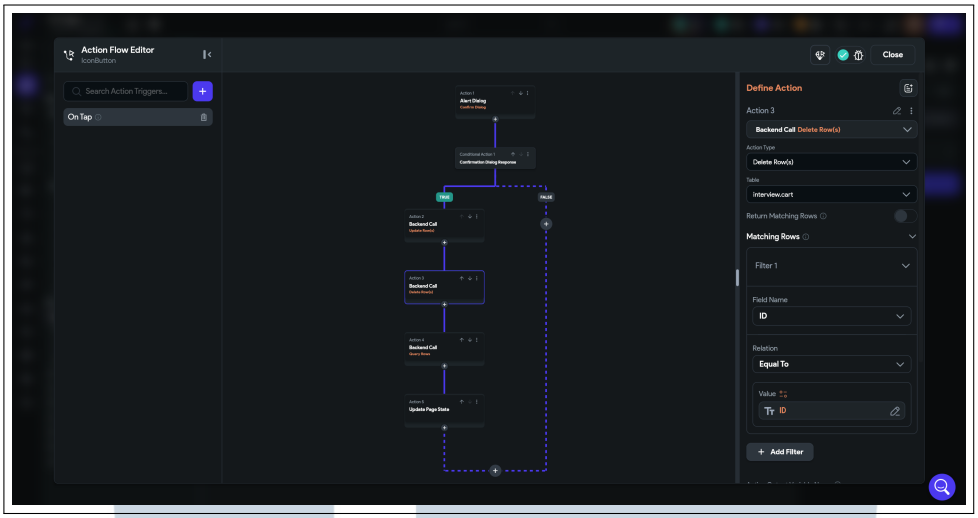
Action flow pada gambar 3.14, ketika pengguna melakukan *swipe* kiri, sistem akan memperbarui baris dalam tabel share.profile.log dan mengatur nilai result menjadi *False*. Menandakan perusahaan mitra menolak rekomendasi

talent. Sebaliknya, ketika pengguna melakukan *swipe* kanan. Data terkait talent yang sedang dilihat akan diperbarui dalam tabel share.profile.log di *backend* menggunakan perintah *Update Row(s)* menjadi *true*. Setelah itu, data baru akan dimasukkan ke dalam tabel interview.cart dengan menggunakan perintah *insert Row*. *Field* yang dimasukkan mencakup company, share.profile, dan share_profile_log. Proses ini memungkinkan aplikasi untuk memperbarui data yang ada dan menambahkan informasi baru ke dalam sistem, memastikan bahwa keputusan perusahaan mitra dalam menerima rekomendasi tercatat dengan baik dalam *backend* dan memberikan pengalaman pengguna yang lancar dan efisien.



Gambar 3.15. *Action Flow* Set Jadwal *Interview* FlutterFlow

Action flow pada gambar 3.15 menggambarkan urutan langkah-langkah yang dibuat dalam aplikasi FlutterFlow ketika pengguna memilih tanggal di kalender. Pertama, sistem akan memperbarui komponen dan aplikasi dengan memproses tanggal yang dipilih di kalender (*Action 1: Update Component State dan Action 2: Update App State*). Kemudian, data tanggal yang dipilih akan diperbarui di tabel interview.cart menggunakan perintah *Backend Call - Update Row(s) (Action 3)*. Selanjutnya, pengguna dapat memilih waktu melalui *Date/Time Picker*, yang akan memperbarui jam yang dipilih di *backend (Action 4)*. Setelah itu, jam yang dipilih akan disimpan dalam tabel interview.cart menggunakan perintah *Backend Call - Update Row(s)* kedua (*Action 5*). Akhirnya, aplikasi akan menampilkan dialog pemberitahuan informasi (*Action 6: Alert Dialog*) dan menutup bottom sheet (*Action 7*). Proses ini memastikan bahwa data terkait dengan tanggal dan waktu yang dipilih diperbarui secara akurat di *backend* dan memberikan pengalaman pengguna yang interaktif dan responsif.



Gambar 3.16. *Action Flow* Menghapus Talent dari FlutterFlow

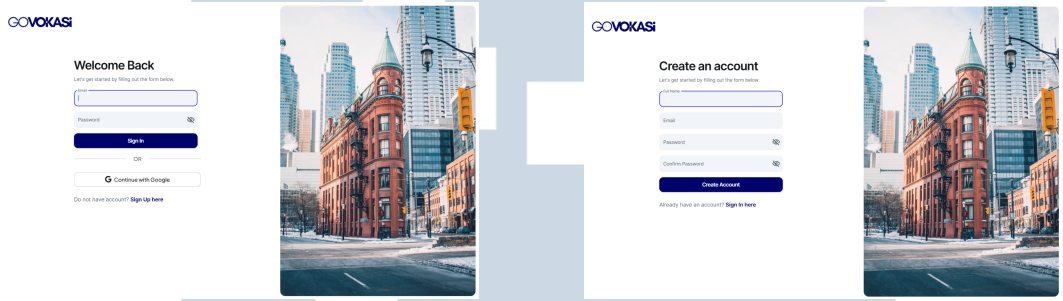
Action flow pada gambar 3.16 menggambarkan urutan langkah-langkah yang dilakukan dalam aplikasi FlutterFlow ketika tombol ditekan. Pertama, sistem akan menampilkan dialog konfirmasi kepada pengguna untuk memastikan apakah mereka ingin melanjutkan aksi yang dimaksud. Jika pengguna mengonfirmasi (*True*), sistem akan melakukan dua tindakan *backend* secara berurutan. Pertama, data terkait result akan diupdate menjadi *False* pada tabel share.profile.log (*Action 2: backend Call - Update Row(s)*). Setelah itu, entitas yang sesuai akan dihapus dari tabel interview.cart menggunakan perintah *Delete Row(s) (Action 3)*. Jika pengguna membatalkan (*False*), tidak ada perubahan yang dilakukan, dan alur berakhir tanpa melakukan aksi apa pun. Setelah penghapusan, aplikasi akan melakukan query untuk memastikan bahwa data yang relevan telah terhapus, dan kemudian memperbarui tampilan halaman (*Action 5: Update Page State*). Proses ini memungkinkan aplikasi untuk secara efisien memperbarui dan menghapus data di *backend* berdasarkan konfirmasi pengguna, serta memastikan tampilan UI mencerminkan perubahan yang terjadi.

3.3.4 Implementasi

Implementasi aplikasi ini menggunakan FlutterFlow, sebuah platform pengembangan aplikasi yang memungkinkan pembuatan aplikasi secara visual dan interaktif. Dengan FlutterFlow, pengembangan aplikasi dilakukan dengan cara mendesain tampilan dan menyusun logika alur aplikasi menggunakan antarmuka grafis. Aplikasi ini kemudian dihosting pada URL https://generasisiapkerja.com/, memberikan akses yang mudah dan cepat kepada pengguna. Hosting melalui

URL ini memastikan bahwa aplikasi dapat diakses secara online dengan kestabilan dan performa yang baik, memungkinkan interaksi pengguna yang lancar dan pengalaman pengguna yang optimal.

A Halaman Login dan Signup

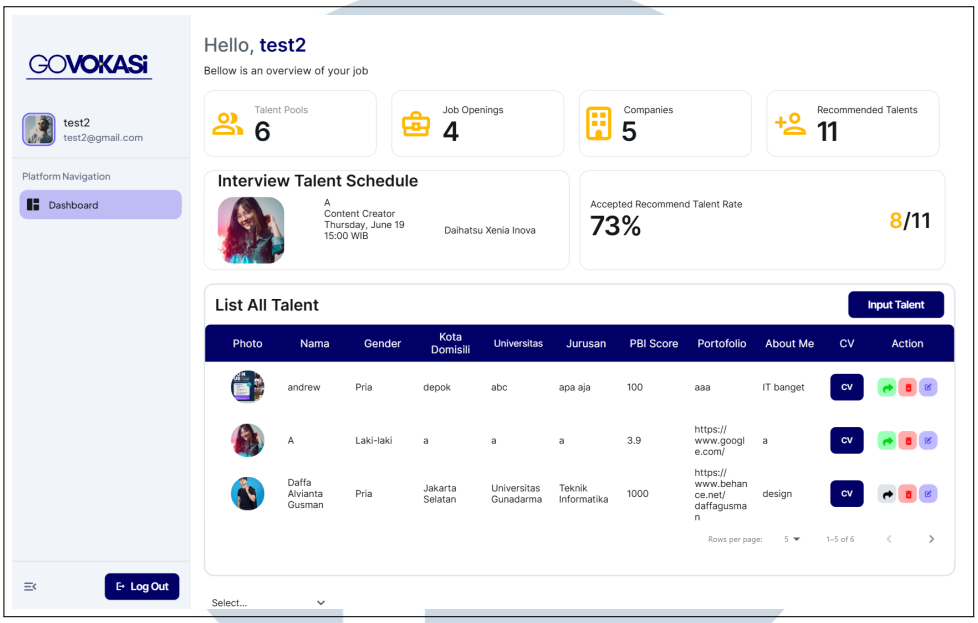


Gambar 3.17. Tampilan Halaman *Login* dan *SignUp*

Pada halaman *login* dan *signup* gambar 3.17, autentikasi dilakukan menggunakan Supabase. tim internal hanya perlu melakukan *login* menggunakan email dan password yang sudah terdaftar sebelumnya. Setelah berhasil *login*, tim internal dapat langsung mengakses *dashboard* aplikasi untuk memonitor dan mengelola data yang ada tanpa perlu melalui proses registrasi tambahan, karena akun sudah dibuatkan sebelumnya. Untuk perusahaan yang belum memiliki akun, proses registrasi dilakukan pada halaman *Create an account*.

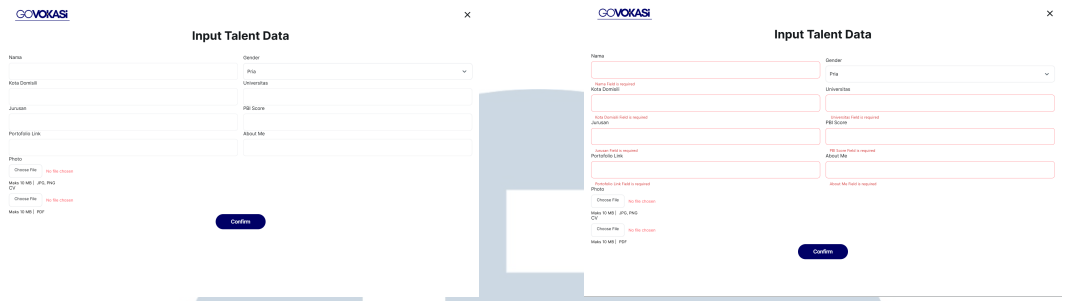
Pengguna dari perusahaan mitra harus mengisi *Full Name, Email, Password, dan Confirm Password*, kemudian menekan tombol *Create Account*. Setelah data pendaftaran dikirim, sistem akan menangani pembuatan akun dan proses verifikasi email. Setelah berhasil mendaftar, perusahaan dapat langsung *login* dengan email dan password yang telah didaftarkan, dan sistem akan memverifikasi kredensial pengguna sebelum memberikan akses ke *dashboard* aplikasi. Halaman ini menyediakan fungsionalitas yang memungkinkan kedua jenis pengguna untuk memperoleh akses yang sesuai, dengan tim internal hanya perlu *login*, sementara perusahaan yang baru mendaftar harus melakukan registrasi terlebih dahulu.

B Halaman Dashboard Internal



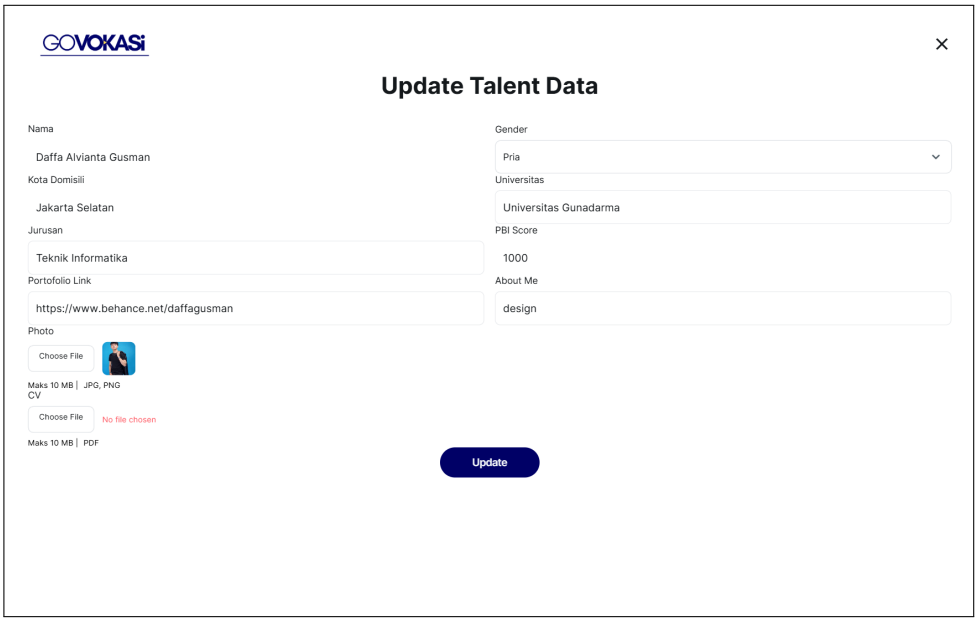
Gambar 3.18. Tampilan Halaman *dashboard* Internal

Pada *dashboard* internal gambar 3.18, tim internal dapat memantau dan mengelola berbagai aspek yang berkaitan dengan talent dan perusahaan mitra. Di bagian Talent Pools, tim internal dapat melihat jumlah talent yang terdaftar dalam sistem, sementara di bagian perusahaan mitra, jumlah perusahaan mitra yang terdaftar juga dapat dipantau. Informasi terkait *Job Openings* menunjukkan jumlah lowongan kerja yang tersedia untuk talent, sementara bagian *Recommended Talents* menampilkan jumlah talent yang sudah direkomendasikan. Tim internal juga dapat melihat Accepted Recommend Talent Rate, yang menunjukkan persentase penerimaan talent berdasarkan rekomendasi yang diberikan. Selain itu, jadwal *interview* untuk semua talent yang diterima dapat dilihat pada *interview Talent Schedule*, lengkap dengan informasi posisi, perusahaan, serta tanggal dan waktu *interview*. Untuk pengelolaan data, tim internal dapat melakukan CRUD pada talent yang terdaftar, termasuk menambah talent baru melalui tombol Input Talent, memperbarui data talent yang ada, serta menghapus talent yang tidak relevan. Dengan berbagai fitur ini, tim internal dapat dengan mudah mengelola dan memantau proses perekrutan, memastikan kelancaran alur kerja dan efisiensi dalam manajemen talent.



Gambar 3.19. Tampilan Halaman Input Talent Form

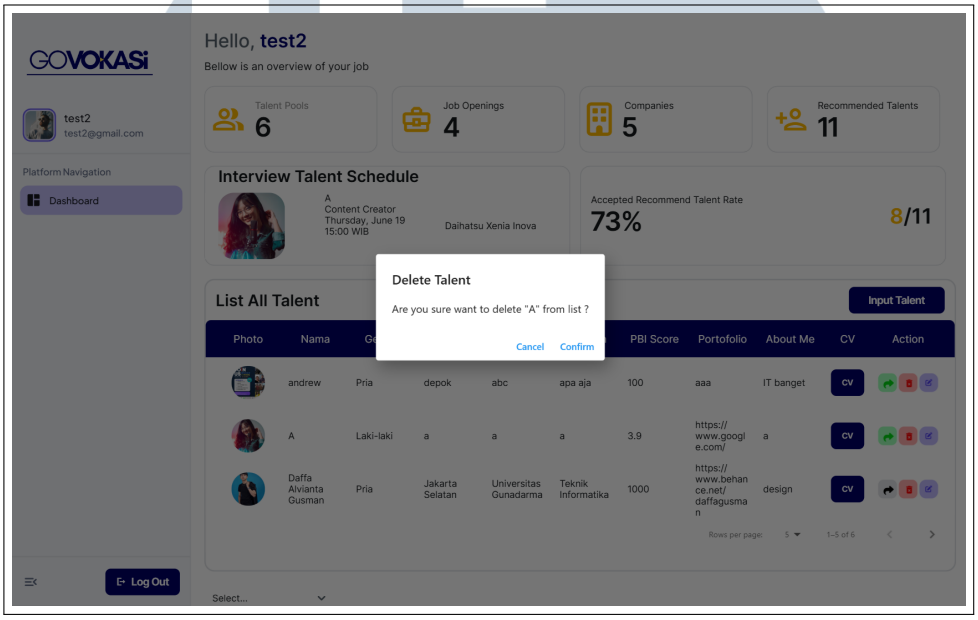
Pada halaman Input Talent Form, tim internal dapat memasukkan berbagai data terkait talent, termasuk nama, jenis kelamin, universitas, jurusan, skor PBI, tautan portofolio, serta foto dan CV. Setiap kolom wajib diisi, dan jika terdapat data yang belum terisi, kolom tersebut akan diberi tanda merah untuk menunjukkan ketidaksesuaian atau kelengkapan data yang belum terpenuhi seperti pada gambar 3.19. Hal ini bertujuan untuk mempermudah pengguna dalam memastikan bahwa semua informasi yang diperlukan telah diinput dengan benar sebelum melanjutkan ke langkah berikutnya.



Gambar 3.20. Tampilan Halaman *Update Talent Form*

Pada halaman *Update Talent Data*, pengguna dapat memperbarui data talent yang telah dimasukkan sebelumnya tanpa perlu mengisi data dari awal. Formulir ini secara otomatis menampilkan informasi yang sudah ada, seperti nama, kota

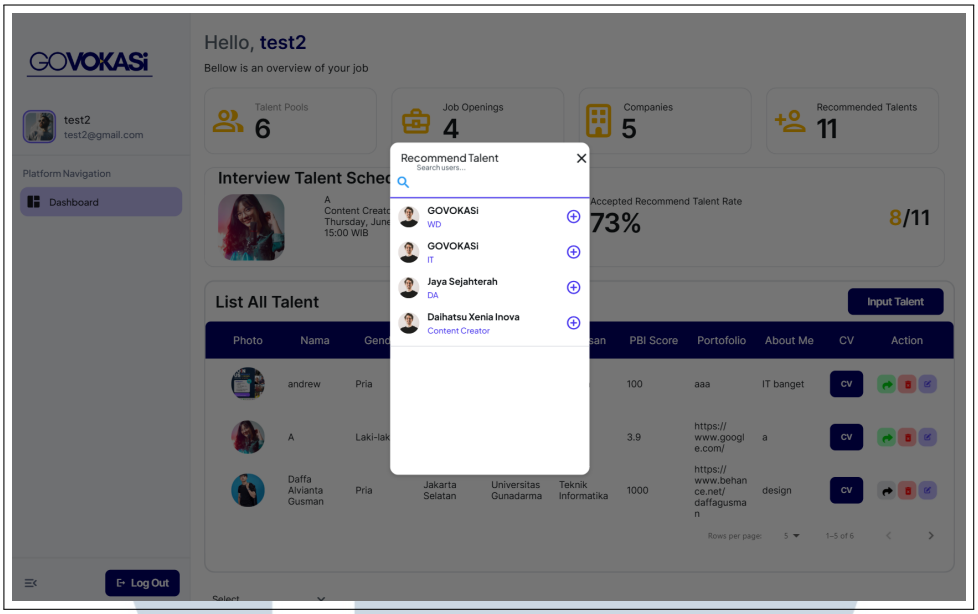
domisili, jurusan, universitas, skor PBI, serta tautan portofolio, yang dapat diedit sesuai kebutuhan. Pengguna hanya perlu mengganti atau memperbarui data yang diperlukan, termasuk foto dan CV, tanpa harus mengisi ulang seluruh formulir seperti pada gambar 3.20. Setelah melakukan perubahan, pengguna dapat menekan tombol ”*update*” untuk memperbarui informasi yang telah disimpan.



Gambar 3.21. Tampilan Modal Konfirmasi Delete Talent

Pada saat pengguna menekan ikon delete pada baris talent, sistem akan menampilkan modal konfirmasi untuk memastikan apakah pengguna benar-benar ingin menghapus talent tersebut dari daftar seperti pada gambar 3.21. Modal ini akan menampilkan pesan konfirmasi dengan pilihan ”*Cancel*” untuk membatalkan aksi penghapusan dan ”*Confirm*” untuk melanjutkan penghapusan talent. Hal ini bertujuan untuk mencegah penghapusan data secara tidak sengaja.

UNIVERSITAS MULTIMEDIA NUSANTARA

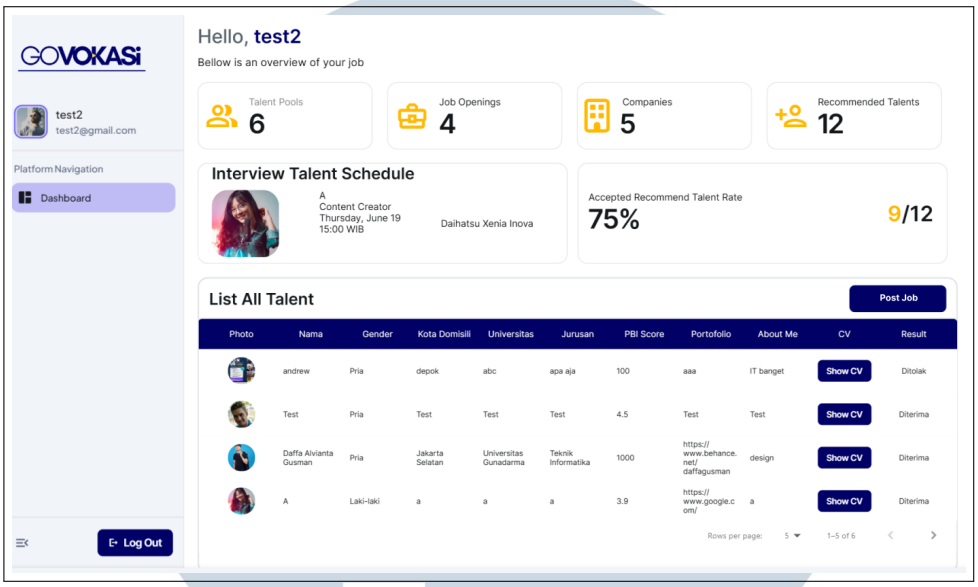


Gambar 3.22. Tampilan Modal Konfirmasi Merekomendasikan Talent

Ketika ikon *share* ditekan, akan muncul modal yang menampilkan daftar perusahaan dan lowongan yang tersedia seperti pada gambar 3.22. Pengguna dapat melakukan pencarian berdasarkan nama perusahaan atau lowongan yang diinginkan. Setelah menemukan yang tepat, pengguna dapat langsung merekomendasikan talent dengan mengklik ikon tambah yang terdapat di samping nama perusahaan atau lowongan. Hal ini mempermudah proses rekomendasi talent tanpa perlu keluar dari tampilan utama.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

C Halaman Dashboard Perusahaan Mitra

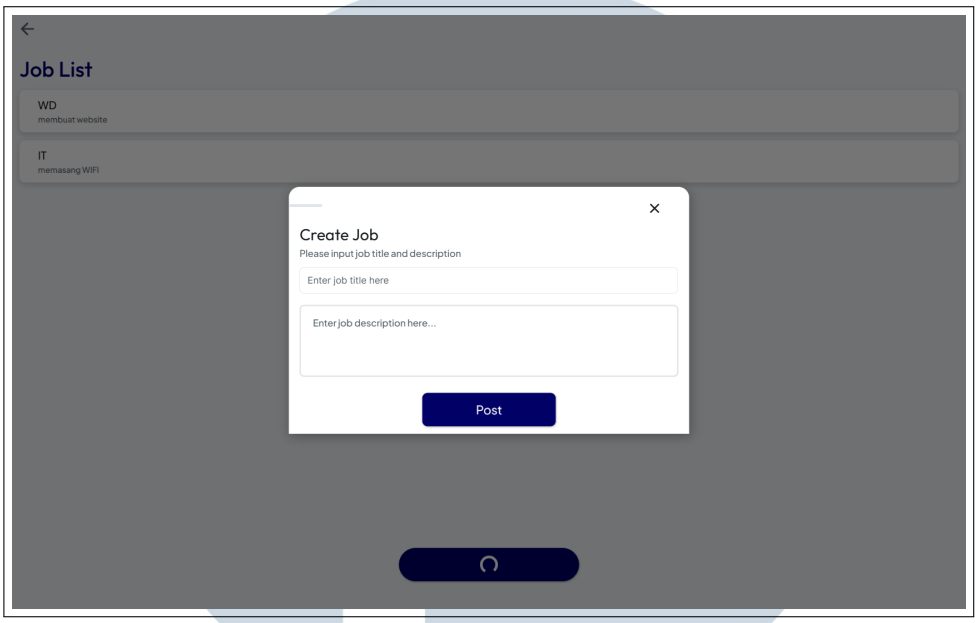


Gambar 3.23. Tampilan *dashboard* Perusahaan Mitra

Pada *dashboard* perusahaan mitra gambar 3.23, tampilan yang ditampilkan serupa dengan *dashboard* internal, namun dengan beberapa perbedaan. Perusahaan mitra dapat melihat informasi terkait talent yang terdaftar, termasuk nama, gender, kota domisili, universitas, jurusan, skor PBI, serta portofolio dan status hasil seleksi, seperti diterima atau ditolak. Namun, perusahaan mitra tidak memiliki akses untuk melakukan operasi CRUD terhadap data talent yang terdaftar. Fungsionalitas utama yang dapat dilakukan oleh perusahaan mitra adalah melakukan "*Post Job*," di mana mereka dapat menambahkan lowongan pekerjaan baru. Fitur lainnya, seperti mengedit atau menghapus data talent, tidak tersedia pada *dashboard* ini untuk perusahaan mitra.

UNIVERSITAS MULTIMEDIA NUSANTARA

D Halaman Job Posting Perusahaan Mitra

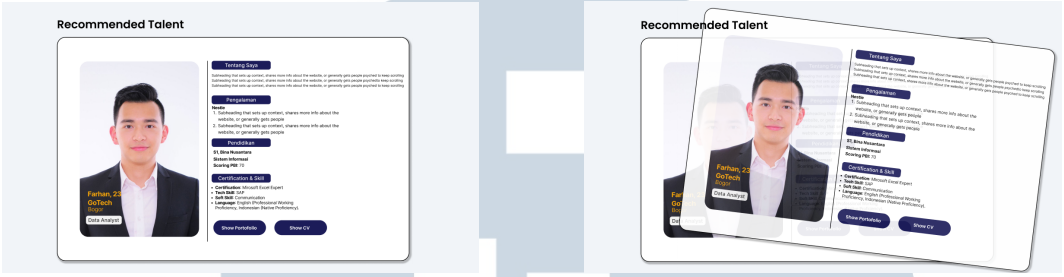


Gambar 3.24. Tampilan Modal Job Posting Perusahaan Mitra

Pada halaman *Job Posting*, perusahaan mitra dapat melihat daftar lowongan pekerjaan yang telah diposting sebelumnya. Selain itu, mereka juga dapat menambahkan lowongan baru dengan mengklik tombol untuk membuat lowongan (Create Job). Sebuah modal akan muncul, yang memungkinkan pengguna untuk mengisi judul pekerjaan dan deskripsi pekerjaan seperti pada gambar 3.24. Setelah informasi tersebut dimasukkan, pengguna dapat menekan tombol "*Post*" untuk mempublikasikan lowongan baru. Halaman ini memudahkan perusahaan mitra dalam mengelola dan menambah lowongan pekerjaan tanpa harus keluar dari tampilan utama.

UNIVERSITAS MULTIMEDIA NUSANTARA

E Halaman Penerimaan Rekomendasi Talent Perusahaan Mitra

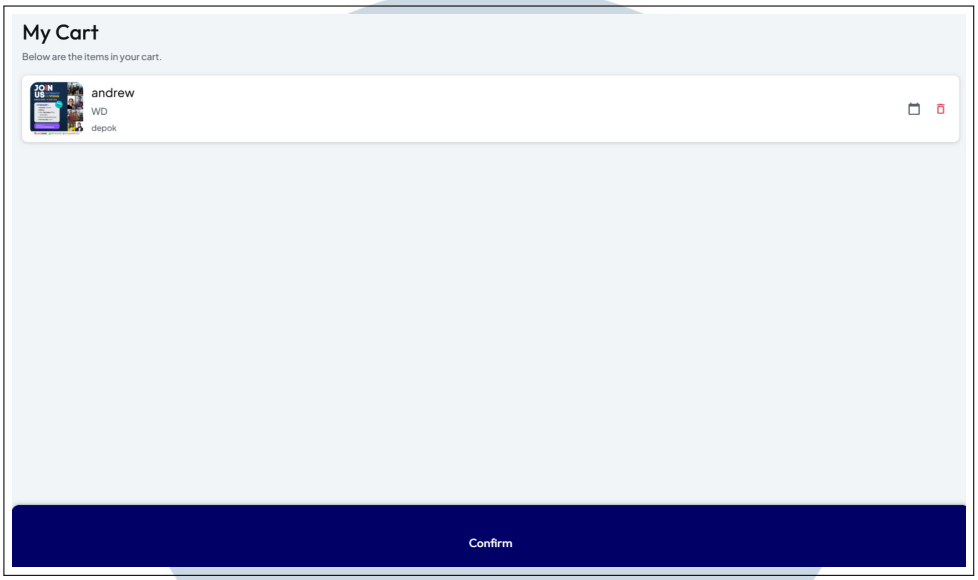


Gambar 3.25. Tampilan Halaman Penerimaan Rekomendasi Talent Perusahaan Mitra

Pada bagian penerimaan rekomendasi talent oleh perusahaan mitra, tampilan menggunakan sistem *swipe* yang memudahkan proses penilaian terhadap talent yang direkomendasikan. Setiap kartu talent yang muncul akan menampilkan foto, informasi terkait, dan beberapa detail seperti pengalaman, pendidikan, serta keterampilan. Perusahaan mitra dapat melakukan *swipe* untuk memberikan keputusan terhadap talent tersebut seperti pada gambar 3.25. *Swipe* ke kanan menunjukkan bahwa rekomendasi talent diterima, sementara *swipe* ke kiri berarti rekomendasi talent ditolak. Fitur ini memungkinkan perusahaan mitra untuk dengan cepat menilai dan memutuskan apakah akan melanjutkan dengan rekomendasi tersebut atau tidak.

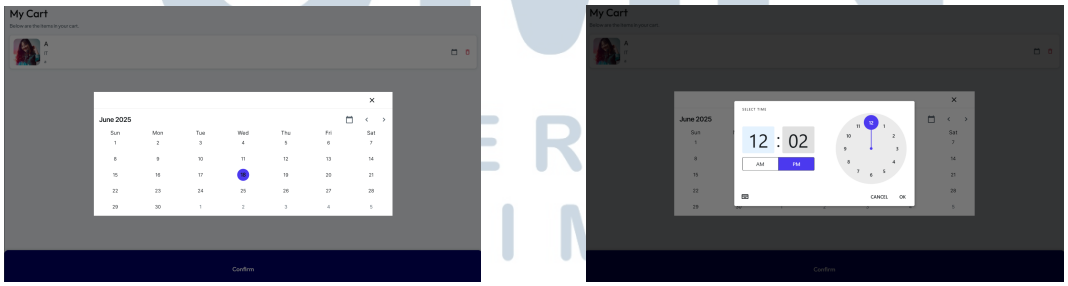
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

F Halaman Cart Konfirmasi Rekomendasi Talent Perusahaan Mitra



Gambar 3.26. Tampilan Halaman *Cart* Konfirmasi Rekomendasi Talent Perusahaan Mitra

Pada halaman *cart* Konfirmasi Rekomendasi Talent gambar 3.26, perusahaan mitra dapat melihat talent yang telah diterima rekomendasinya. Daftar talent yang muncul di *cart* ini dapat dikelola lebih lanjut, misalnya dengan mengatur jadwal wawancara, seperti memilih tanggal dan jam yang sesuai untuk *interview* dengan talent. Selain itu, pengguna juga dapat menghapus talent dari *cart* jika diperlukan, menggunakan ikon hapus yang tersedia di samping setiap nama talent. Setelah semua pengaturan selesai, perusahaan mitra dapat menekan tombol "*Confirm*" untuk mengonfirmasi tindakan tersebut dan melanjutkan ke langkah selanjutnya dalam proses rekrutmen.



Gambar 3.27. Tampilan Set Jadwal *Interview* Perusahaan Mitra

Perusahaan mitra dapat mengatur jadwal wawancara dengan talent yang telah diterima rekomendasinya. Untuk melakukannya, perusahaan mitra dapat

memilih tanggal melalui kalender yang muncul setelah mengklik ikon kalender seperti pada gambar 3.27. Setelah memilih tanggal yang diinginkan, perusahaan mitra dapat melanjutkan dengan memilih waktu wawancara dengan mengklik ikon jam. Sebuah interface waktu akan muncul, di mana perusahaan mitra dapat memilih waktu yang sesuai, termasuk memilih antara AM atau PM, serta menentukan menit dengan menggunakan roda waktu. Setelah selesai memilih tanggal dan waktu, perusahaan mitra dapat menekan tombol ”OK” untuk mengonfirmasi jadwal wawancara, atau membatalkannya jika perlu.

3.4 Kendala yang Ditemukan

Selama pelaksanaan magang di PT Gerbang Govokasi Indonesia, terdapat sejumlah tantangan yang muncul baik dari sisi teknis maupun proses kerja. Beberapa kendala utama yang dihadapi meliputi:

1. Belum menguasai dan memiliki pengalaman menggunakan FlutterFlow dan Supabase, yang membutuhkan waktu untuk memahami dan menguasai kedua *platform* tersebut.
2. Ketidaksesuaian dalam pemberian nama tabel dan kolom di Supabase, yang menimbulkan kebingungan dan kesulitan saat proses pembuatan aplikasi, terutama dalam pengelolaan dan integrasi data.

3.5 Solusi Atas Kendala yang Ditemukan

1. Meluangkan waktu untuk mempelajari FlutterFlow dan Supabase dengan memanfaatkan sumber daya online, seperti tutorial dan dokumentasi resmi, untuk memahami cara menggunakan kedua *platform* tersebut. Selain itu, bisa juga dengan meminta bantuan langsung dari supervisor yang lebih berpengalaman untuk mendapatkan bimbingan dan dukungan teknis.
2. Menetapkan konvensi penamaan yang konsisten dan jelas untuk tabel dan kolom, sehingga meminimalisir kebingungannya. Setelah konvensi penamaan diterapkan, perlu melakukan review dan audit struktur *database* untuk memastikan bahwa nama-nama tabel dan kolom sudah sesuai dengan standar yang disepakati. Dokumentasi *database* yang lengkap juga sangat penting, mencakup deskripsi tabel, kolom, tipe data, dan relasi antar tabel agar mudah dipahami. Jika ada perubahan pada struktur *database*, pembuatan

skrip pembaruan untuk memastikan data tetap aman dan aplikasi tidak terganggu adalah langkah yang perlu dilakukan. Terakhir, uji coba dan validasi perlu dilakukan untuk memastikan aplikasi berjalan dengan lancar setelah perubahan tersebut.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA