

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

3.1.1 Kedudukan Magang

Selama periode magang di PT. Zeelandia Indonesia, mahasiswa magang menempati posisi sebagai Full-Stack Developer. Posisi ini berada di bawah departemen ICT (Information and Communication Technology) perusahaan. Sebagai Full-Stack Developer, tugas utama mahasiswa magang adalah mengembangkan, memelihara, dan memperbarui aplikasi HRIS (Human Resources Information System) perusahaan.

3.1.2 Atasan dan Rekan Kerja

mahasiswa magang melapor langsung kepada Pak Achmad Muzani, yang merupakan Head di departemen ICT. Beliau menjadi pemimpin tim pengembangan aplikasi HRIS dan bertanggung jawab atas pengawasan proyek dan arahan pekerjaan sehari-hari. mahasiswa magang juga bekerja bersama dengan Mas Bintang dalam membangun proyek HRIS ini. Kolaborasi dengan rekan kerja ini adalah bagian penting dalam pelaksanaan tugas dan pencapaian proyek-proyek yang ditetapkan.

3.1.3 Struktur Organisasi

PT. Zeelandia Indonesia memiliki struktur organisasi yang terstruktur dengan baik. Departemen ICT, di bawah naungan yang lebih besar, memiliki tanggung jawab khusus dalam pengelolaan sistem informasi perusahaan. Departemen ini bekerja erat dengan departemen lain, terutama Departemen Sumber Daya Manusia, untuk memastikan aplikasi HRIS berfungsi secara efektif dalam mendukung manajemen sumber daya manusia

3.2 Tugas yang Dilakukan

3.2.1 Analisis Kebutuhan

Peran dalam tugas ini mencakup interaksi yang intensif dengan tim HR perusahaan, di mana tujuan utamanya adalah memahami kebutuhan dan permasalahan yang dihadapi oleh pengguna aplikasi HRIS. Dalam konteks ini, terlibat dalam proses mengumpulkan persyaratan dan menyusun dokumen analisis kebutuhan sebagai langkah awal dalam pengembangan aplikasi. Selain itu, aktif melakukan wawancara dengan anggota tim HR untuk mendapatkan pemahaman mendalam tentang proses manajemen sumber daya manusia yang terkait. Upaya juga melibatkan interpretasi dan analisis data yang diperlukan untuk mendukung pengembangan fitur-fitur baru dalam aplikasi HRIS, memastikan bahwa solusi yang dihasilkan sesuai dengan kebutuhan dan tujuan yang telah ditetapkan.

3.2.2 Pengembangan Aplikasi

Dalam peran sebagai Full-Stack Developer, tanggung jawab melibatkan pengembangan, pemeliharaan, dan pembaruan aplikasi HRIS. Tugasnya mencakup menerjemahkan persyaratan ke dalam kode pemrograman menggunakan teknologi dan bahasa pemrograman yang relevan. Selain itu, mahasiswa magang juga bertanggung jawab untuk mengembangkan fitur-fitur baru sesuai dengan pedoman desain yang telah ditetapkan. Pekerjaannya mencakup pemrograman di kedua sisi, baik di sisi depan (*frontend*) maupun sisi belakang (*backend*) aplikasi. Fokus tambahan diberikan pada aspek desain antarmuka pengguna, dengan upaya memastikan bahwa aplikasi memiliki antarmuka yang responsif dan mudah digunakan bagi pengguna akhir.

3.2.3 Uji Coba

Dalam rangka memastikan kinerja optimal sebelum peluncuran dan selama pembaruan, dilibatkannya mahasiswa magang adalah untuk membuat skenario uji coba dan melaksanakan pengujian fungsional aplikasi. Tugas ini juga mencakup deteksi dan pelaporan bug atau masalah yang teridentifikasi selama proses pengujian. Mahasiswa magang bertanggung jawab untuk berkoordinasi dengan tim pengujian, memastikan setiap fitur diuji secara menyeluruh, dan menjaga kualitas keseluruhan aplikasi HRIS.

3.2.4 Pemeliharaan

Dalam lingkup tugas pemeliharaan, terlibat dalam perbaikan rutin, penanganan masalah teknis, dan pembaruan aplikasi HRIS menjadi fokus utama. Ini mencakup langkah-langkah seperti merespons permintaan perbaikan dan perubahan yang diajukan oleh tim HR atau manajemen. Selain itu, melakukan pemantauan secara berkala terhadap kinerja aplikasi, serta menangani gangguan atau masalah teknis yang mungkin muncul selama penggunaan. Pemeliharaan juga melibatkan pelaksanaan pembaruan keamanan dan perangkat lunak secara berkala untuk menjaga kehandalan dan keamanan aplikasi..

3.2.5 Pelaporan dan Dokumentasi

mahasiswa magang bertanggung jawab untuk menyusun laporan berkala tentang kemajuan proyek, perubahan yang diimplementasikan, dan masalah yang telah diatasi. Selain itu, mahasiswa magang juga harus mendokumentasikan kode, konfigurasi, dan prosedur kerja untuk aplikasi HRIS agar informasi tersebut dapat diakses oleh tim teknis di masa depan.

3.3 Uraian Pelaksanaan Magang

Dalam pelaksanaan magang, terdapat pekerjaan yang harus diselesaikan mahasiswa magang dalam suatu periode tertentu. Maka dari itu, pemegang hendak menguraikan apa saja pekerjaan yang dilakukan dalam periode setiap minggu kerja magang di PT. Zeelandia Indonesia. Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Arsitektur aplikasi HRIS yang akan dikembangkan dan bahasa pemrograman yang digunakan dalam proyek ini dipelajari lebih dalam antara lain <i>Laravel, Ionic React, Postman, MySQL Workbench</i> , dan beberapa <i>server requirements</i> seperti <i>WAMPP server</i> . Pada minggu pertama ini, juga terdapat on boarding dimana rekan-rekan kerja diperkenalkan dan belajar mengenal lingkungan kerja di PT. Zeelandia Indonesia.
2	Kebutuhan dan persyaratan pengguna dalam proyek HRIS diidentifikasi serta skema basis data yang diperlukan mulai dirancang. Setelah mengenal apa saja tools yang digunakan oleh PT. Zeelandia Indonesia, maka pertemuan pertama dilakukan guna membahas proyek dan pengenalan SOP (<i>Standard Operating Procedure</i>) yang terdapat di divisi <i>Human Resource</i> PT. Zeelandia Indonesia. Pada pertemuan ini dijelaskan kebutuhan rekan HR dalam mendigitalisasi <i>Performance Management</i> karyawan. Syarat-syarat atau ketentuan penilaian yang dibahas dalam pertemuan ini antara lain <i>Performance Appraisal</i>
3	Kebutuhan dan persyaratan pengguna dalam proyek HRIS masih diidentifikasi dengan melakukan berbagai pertemuan dengan pihak <i>Human Resources</i> . Hasil diskusi diputuskan dan pengembang diminta berfokus pada sistem penilaian karyawan, yakni penilaian KPI, PA, dan CGA.
4	Proses <i>detailing</i> skema basis data yang diperlukan. Pada proses ini, dilakukan diskusi dengan pengembang senior dan pihak <i>human resource</i> mengenai apa saja basis data yang sekiranya diperlukan oleh aplikasi. Skema basis data dirancang dan direvisi sehingga menghasilkan skema basis data yang final. Pada awalnya, pertanyaan dibuat kolom menyamping pada database dikarenakan awalnya diperkirakan bahwa seluruh form berisikan jumlah pertanyaan yang sama, namun ternyata setiap karyawan bisa saja berbeda, maka dibuatlah kolom <i>question id</i> yang digunakan untuk <i>mapping</i> <i>question id</i> dari tabel <i>response</i> ke tabel <i>question</i> .

Minggu Ke -	Pekerjaan yang dilakukan
5	<p><i>Frontend</i> aplikasi HRIS mulai dirancang dengan fokus pada antarmuka pengguna yang mudah digunakan. Proses perancangan <i>frontend</i> menggunakan figma untuk mengeksekusi <i>wireframe</i> dari tampilan keseluruhan aplikasi. Ionic React dijadikan <i>framework</i> yang digunakan untuk mengeksekusi rancangan <i>frontend</i> yang telah dibuat sebelumnya. Hal ini dikarenakan permintaan dari HR yang memiliki visi ke depan untuk menambahkan fitur <i>request leaves</i> melalui <i>smartphone</i>, sehingga <i>developer</i> menggunakan <i>framework</i> yang mampu membentuk aplikasi hybrid. Sedangkan, untuk <i>framework</i> CSS yang digunakan adalah Tailwind CSS <i>framework</i>.</p>
6	<p><i>Frontend</i> aplikasi HRIS mulai dikembangkan dengan fokus pada antarmuka pengguna yang mudah digunakan menggunakan Ionic React dan Tailwind. <i>Developer</i> mulai membuat <i>login form</i>, <i>sign up form</i>, dan <i>sidebar menu</i> yang lebih familiar bagi admin dari HR untuk melakukan navigasi halaman dan fitur.</p>
7	<p>Back-end sistem HRIS, termasuk pengaturan database, pengelolaan API, dan otentikasi pengguna mulai dikembangkan berdasarkan skema yang telah dibuat. Sistem <i>backend</i> dibangun dengan menggunakan <i>framework</i> Laravel dan menggunakan Res API yang menghubungkan <i>framework frontend</i> ke basis data. Berawal dari penghubungan CRUD (<i>Create, Read, Update, Delete</i>) karyawan, hingga penyimpanan hasil penilaian karyawan melalui form yang dibangun.</p>
8	<p>Back-end sistem HRIS, termasuk pengaturan database, pengelolaan API, dan otentikasi pengguna diintegrasikan dengan <i>front-end</i> dari aplikasi HRIS PT. Zeelandia. <i>Developer</i> mulai melakukan integrasi <i>frontend</i> dengan <i>backend</i> menggunakan Res API dari Laravel dan <i>library</i> Axios yang digunakan pada Ionic React untuk melakukan <i>fetch data</i> menggunakan API untuk ditampilkan pada <i>frontend</i> aplikasi HRIS PT. Zeelandia.</p>

Minggu Ke -	Pekerjaan yang dilakukan
9	Uji coba aplikasi HRIS dilakukan untuk mengidentifikasi dan memperbaiki masalah teknis dan antarmuka pengguna. Uji coba ini dilakukan oleh <i>user</i> admin dari departemen HR PT. Zeelandia Indonesia dan manager ICT. Pada saat ini, terdapat miskomunikasi dari departemen HR dengan <i>developer</i> dikarenakan misinformasi penilaian sikap atau perilaku dari setiap karyawan meskipun dalam satu departemen dapat berbeda. Bahkan kategori kompetensinya bisa berubah baik itu <i>soft skill competencies</i> ataupun <i>technical competencies</i> .
10	Pengembangan aplikasi dilanjutkan dengan menambahkan fitur-fitur tambahan yang dibutuhkan oleh tim HR Zeelandia Indonesia. Yakni fitur untuk <i>user</i> melakukan pemilihan kompetensi karyawan yang akan dinilai. Proses pengembangan ini menggunakan fungsi yang memisahkan setiap kategori dan membuat selectionnya.
11	Pengembangan aplikasi dilanjutkan dengan mengubah berbagai fitur dari awalnya memilih satu atau dua penilai melalui tombol menjadi <i>mapping</i> otomatis dari id <i>supervisor</i> karyawan, sehingga diperlukan kolom dan data tambahan pada basis data.
12	<p>Evaluasi selanjutnya dilakukan dengan tujuan meninjau ulang keperluan <i>user</i> dan alur sistem yang sudah sesuai serta revisi yang diminta segera dilaksanakan. beberapa kesalahan yang ditemukan antara lain :</p> <ol style="list-style-type: none"> 1. user yang telah mengisi formulir tabel target kerja tidak bisa melakukan penyuntingan input jika kolom sudah tidak aktif padahal formulir belum disimpan. 2. Terdapat kesalahan pada sistem pemilihan kompetensi dimana jika terdapat nilai nol di setiap pertanyaan di kategori kompetensi maka pilihan tidak ditampilkan sistem, padahal jika form masih kosong maka seharusnya semua kategori ditampilkan. 3. Terdapat beberapa penambahan kolom pada tabel pertanyaan PA (<i>Performance Appraisal</i>) dikarenakan bobot ternyata merupakan input dan bukan <i>mapping</i> dari master pertanyaan.

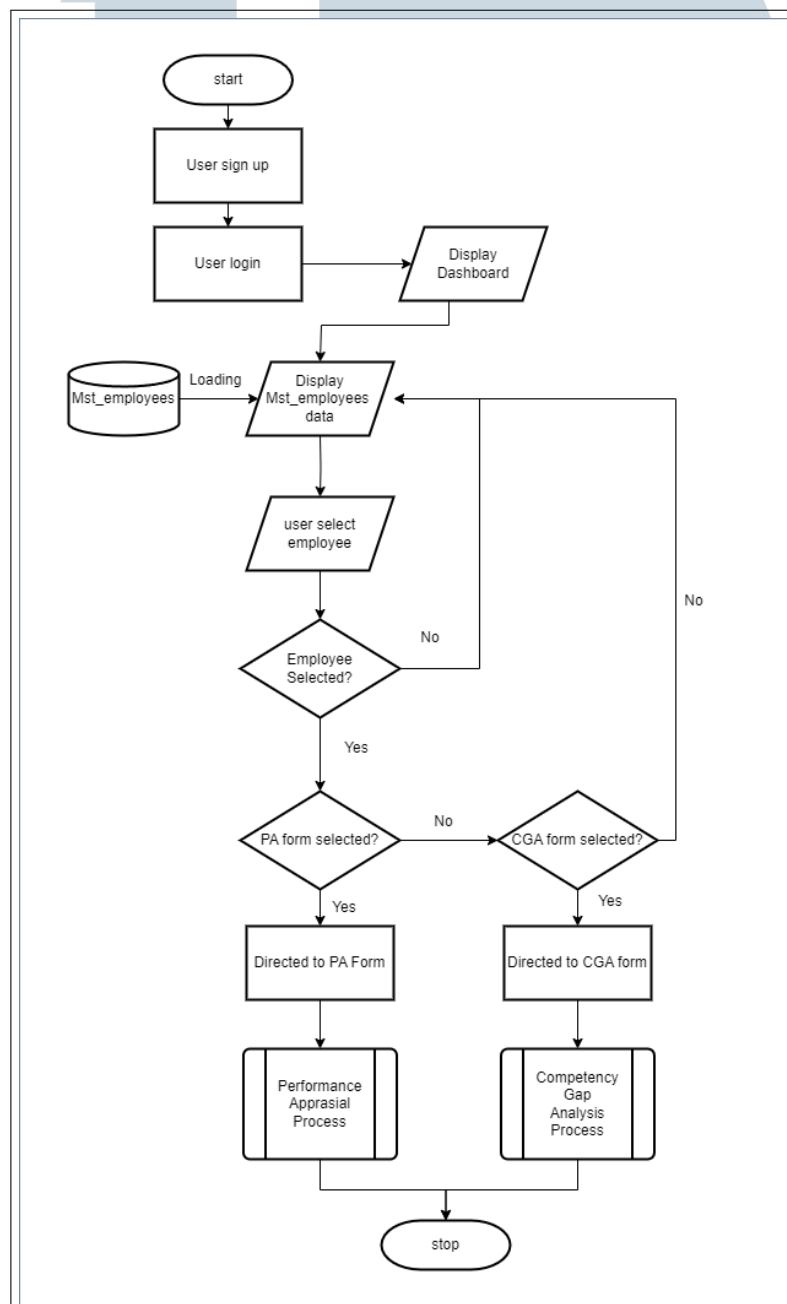
Minggu Ke -	Pekerjaan yang dilakukan
13	Pengembangan dan perbaikan aplikasi HRIS PT. Zeelandia dilanjutkan dengan menambahkan permintaan dari <i>user</i> departemen <i>human resources</i> . <i>Developer</i> menyunting fungsi pengecekan nilai nol dan <i>selection competencies</i> pada <i>frontend</i> aplikasi HRIS PT. Zeelandia Indonesia
14	Perbaikan dilaksanakan dan ditinjau ulang oleh <i>supervisor</i> dan dilakukan tes kembali bersama dengan <i>user</i> dari departemen <i>human resource</i> . Pada perbaikan kali ini, seluruh komponen penilaian sudah hampir final kecuali form penilaian untuk tingkat <i>supervisor</i> ke atas.
15	Integrasi data yang diperlukan mulai dilakukan bersama pihak departemen <i>human resources</i> . Dan secara bersamaan dilakukan pengembangan formulir penilaian karyawan tingkat <i>supervisor ke atas</i> . Data yang perlu diintegrasikan adalah <i>master employee</i> dan <i>master competencies (soft and technical)</i>
16	Peninjauan ulang setelah data diintegrasikan dilakukan serta dilakukan tahap finalisasi pengembangan aplikasi HRIS PT. Zeelandia untuk form penilaian karyawan tingkat <i>supervisor</i> ke atas dan fitur penilaian karyawan siap diluncurkan.



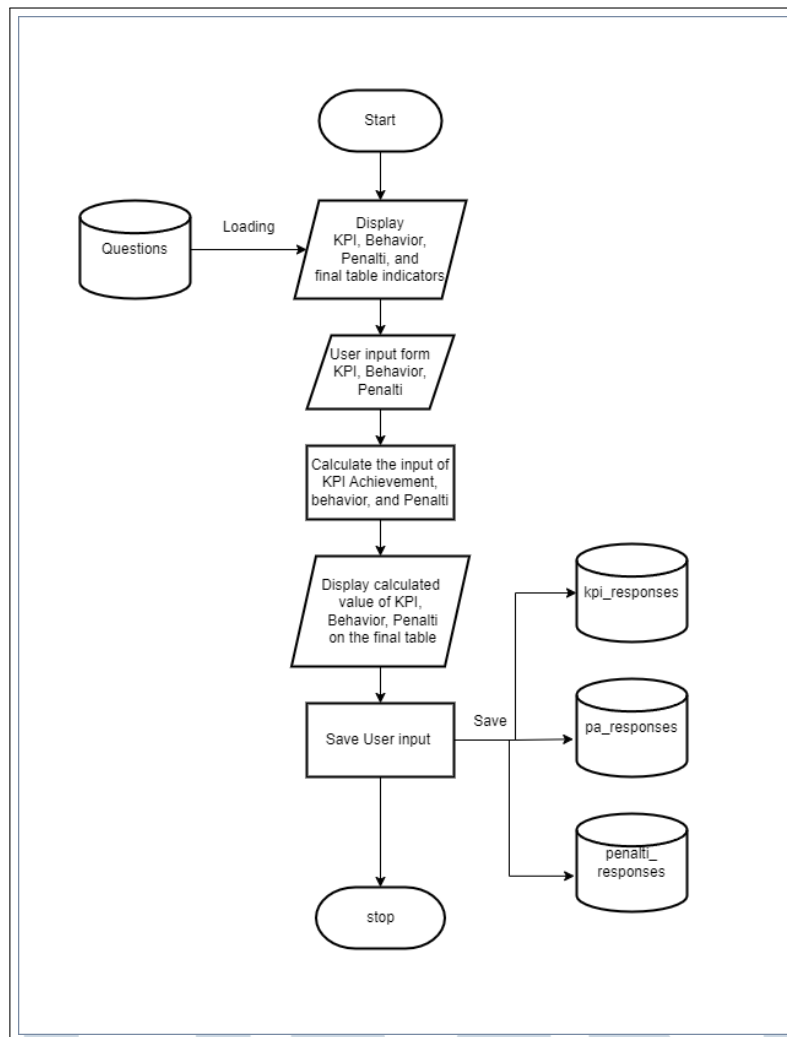
3.4 Bukti Hasil Kerja Magang

3.4.1 Proses Implementasi tampilan *front-end* aplikasi *hybrid* HRIS PT. Zeelandia Indonesia

A. Perancangan sistem aplikasi *hybrid* HRIS PT. Zeelandia Indonesia

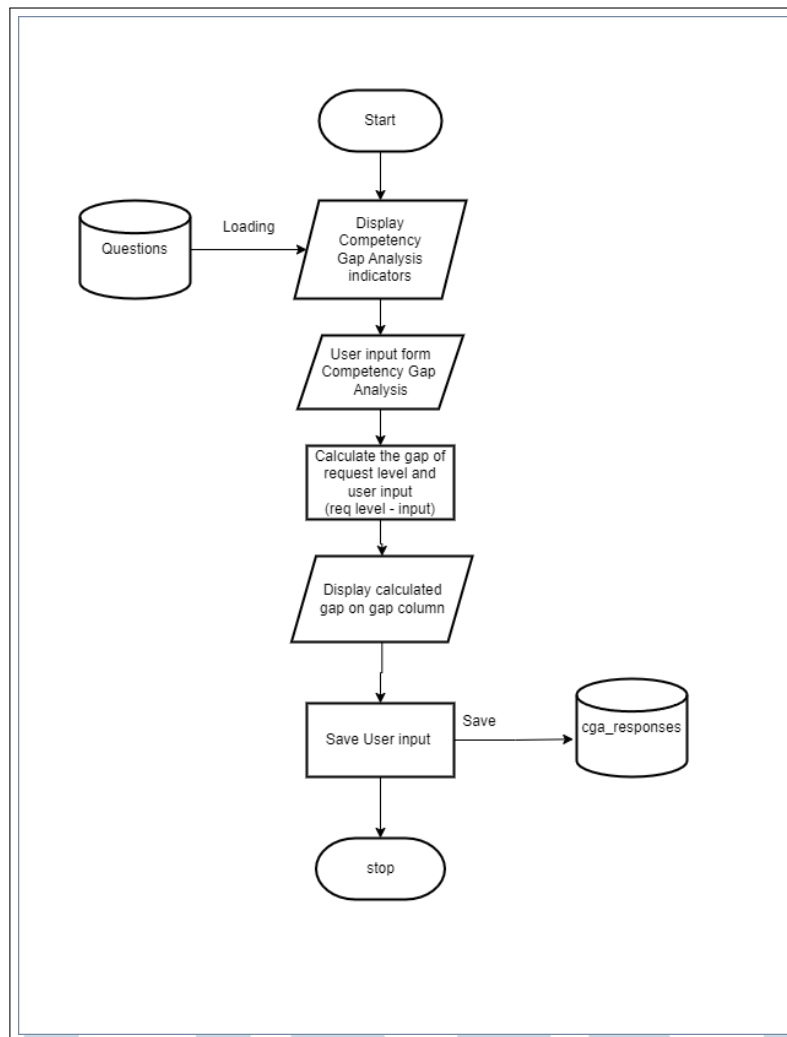


Gambar 3.1. Alur sistem utama fitur penilaian dalam aplikasi



Gambar 3.2. Alur sistem modul *Performance Appraisal Process* dalam aplikasi

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



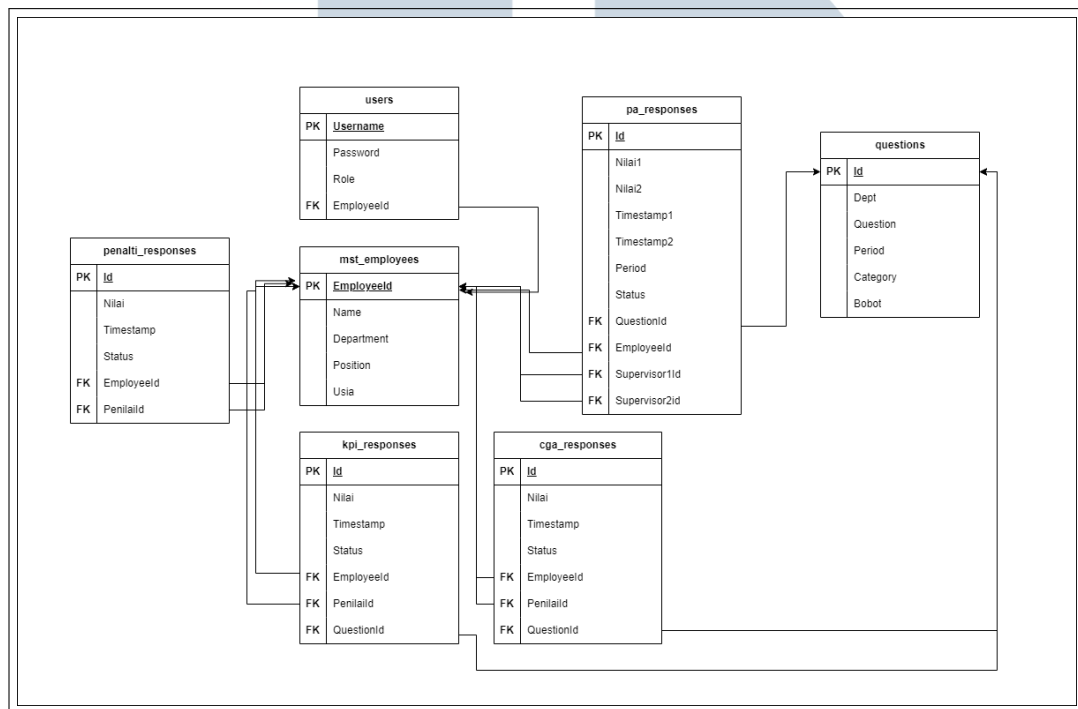
Gambar 3.3. Alur sistem modul *Competency Gap Analysis Process* dalam aplikasi

Gambar 3.1, 3.2, dan 3.3 menggambarkan alur sistem fitur penilaian dalam aplikasi HRIS PT. Zeelandia Indonesia, yang melibatkan beberapa proses berikut:

1. Sistem dimulai dengan mendaftarkan pengguna dan melakukan *login*.
2. Setelah menuju *dashboard*, pengguna memilih *employee database* untuk menampilkan daftar karyawan.
3. Daftar karyawan yang ditampilkan terdiri dari data *user* yang sedang *login* dan karyawan yang merupakan bawahan dari *user*.
4. *User* perlu memilih karyawan yang ingin dinilai atau dievaluasi.
5. Selanjutnya, *user* perlu memilih formulir penilaian yang ingin diterapkan pada karyawan yang dipilih, seperti *Competency Gap Analysis* atau

Performance Appraisal.

6. Setelah mengisi formulir, data penilaian dikirimkan ke basis data melalui API.



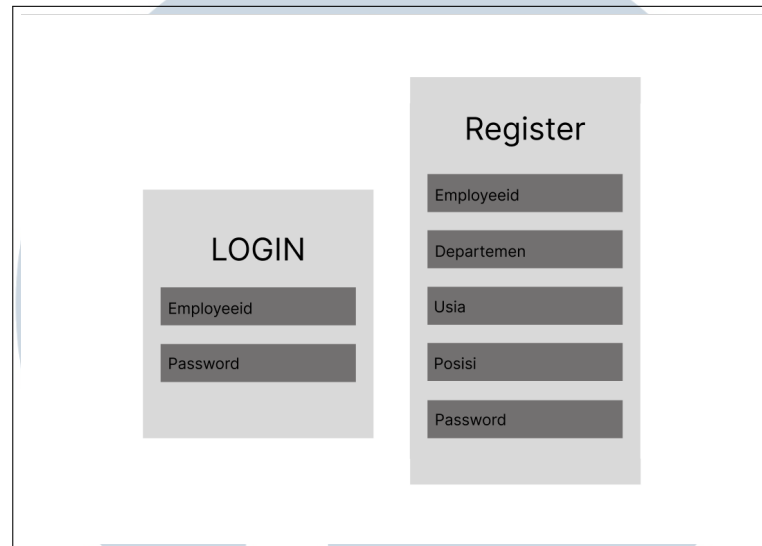
Gambar 3.4. Diagram basis data aplikasi HRIS PT. Zealandia Indonesia

Basis data dalam gambar 3.4 dirancang dengan struktur yang terorganisir, terdiri dari beberapa tabel utama yang masing-masing memainkan peran penting dalam fungsi aplikasi ini. Tabel pertama adalah "Master User," yang bertanggung jawab atas manajemen pengguna sistem, mencakup informasi seperti nama pengguna (*username*) dan kata sandi (*password*).

Selanjutnya, terdapat tabel "Master Employee" yang menyimpan data lengkap mengenai setiap karyawan di perusahaan. Informasi seperti nama, jabatan, dan unit kerja dapat diakses dan dikelola melalui tabel ini. Tabel "Master Questions" digunakan untuk menyimpan pertanyaan-pertanyaan yang akan diajukan dalam proses penilaian karyawan. Setiap pertanyaan memiliki atribut seperti jenis pertanyaan dan kriteria penilaian yang terkait.

Terakhir, tabel berbagai Responses (KPI, PA, Penalti) mencatat hasil penilaian yang diberikan oleh pengguna HR (*Human Resources*). Informasi ini mencakup tanggapan atau nilai yang diberikan untuk setiap pertanyaan penilaian terhadap karyawan. Dengan struktur tabel ini, aplikasi HRIS dapat efektif

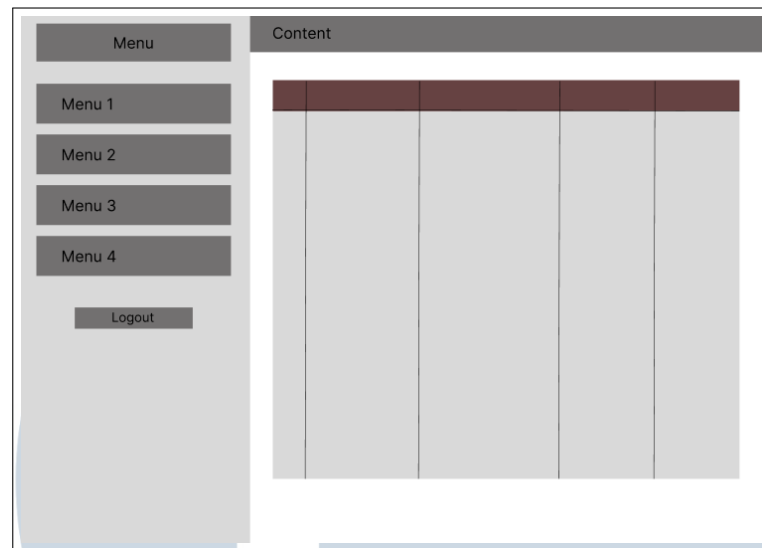
mengelola dan menyimpan data terkait pengguna, karyawan, pertanyaan penilaian, dan hasil penilaian, memberikan dasar yang kokoh untuk operasional sistem manajemen sumber daya manusia.



Gambar 3.5. Wireframe formulir login dan register aplikasi

Wireframe formulir login dan sign up pada gambar 3.5 dirancang dengan tujuan menyediakan antarmuka yang sederhana dan intuitif, memudahkan pengguna untuk login atau mendaftar dengan efisien. Form login memiliki kolom input untuk ID karyawan dan kata sandi, serta tombol "Login" untuk memvalidasi data input dan data dari basis data. Form sign up mencakup kolom input tambahan untuk informasi pribadi dan tombol "Signup" untuk menambahkan data user baru pada basis data. Desain ini bertujuan agar pengguna dapat membantu *developer* dalam mendapatkan gambaran rancangan *frontend* yang hendak dibangun.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

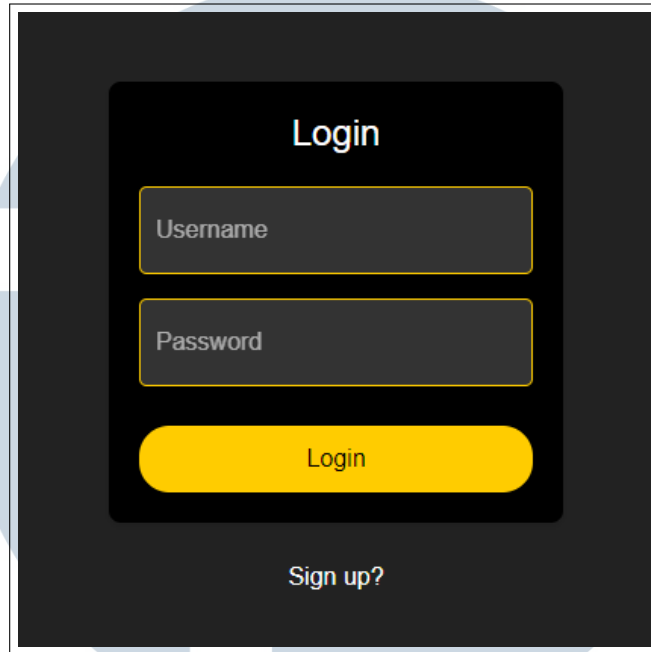


Gambar 3.6. Wireframe tata letak menu dan konten aplikasi

Pada ilustrasi wireframe yang dapat dilihat pada Gambar 3.6, terlihat bahwa aplikasi HRIS PT. Zeelandia Indonesia dirancang dengan menggunakan *sidebar layout*. Pilihan desain ini memposisikan *navigation bar* pada bagian samping tata letak aplikasi. *Sidebar* tersebut bertujuan untuk memberikan akses cepat dan navigasi yang intuitif bagi pengguna dalam menjelajahi berbagai fitur dan menu yang tersedia dalam aplikasi. Dengan menyematkan *navigation bar* pada bagian samping, pengguna dapat dengan mudah mengakses dan beralih antar halaman atau modul tanpa mengganggu ruang konten utama aplikasi. Hal ini bertujuan untuk meningkatkan efisiensi dan pengalaman pengguna dalam berinteraksi dengan sistem HRIS.

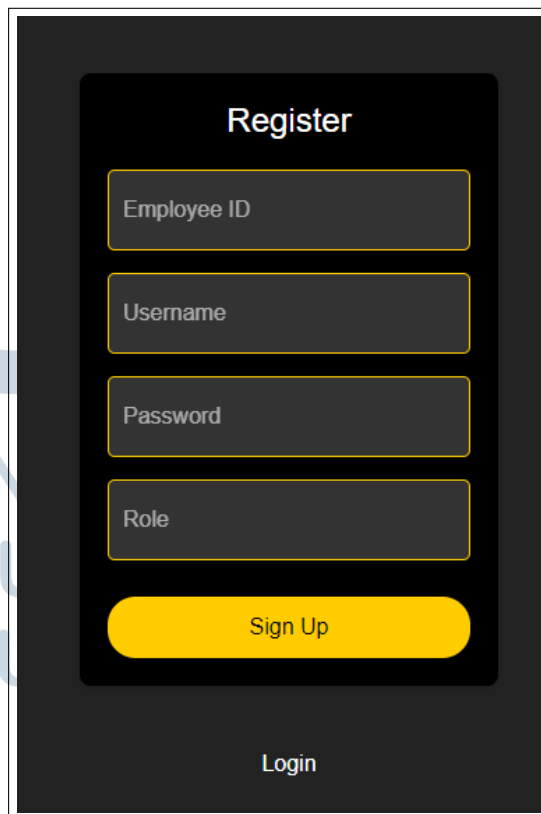
U M M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

B. Tampilan *front-end* aplikasi *hybrid* HRIS PT. Zeelandia Indonesia



The image shows a login form with a dark background. At the top, the word "Login" is written in white. Below it are two input fields: "Username" and "Password", both with yellow borders. A yellow "Login" button is positioned below the password field. At the bottom, there is a "Sign up?" link in white text.

Gambar 3.7. Formulir login pengguna aplikasi HRIS PT. Zeelandia

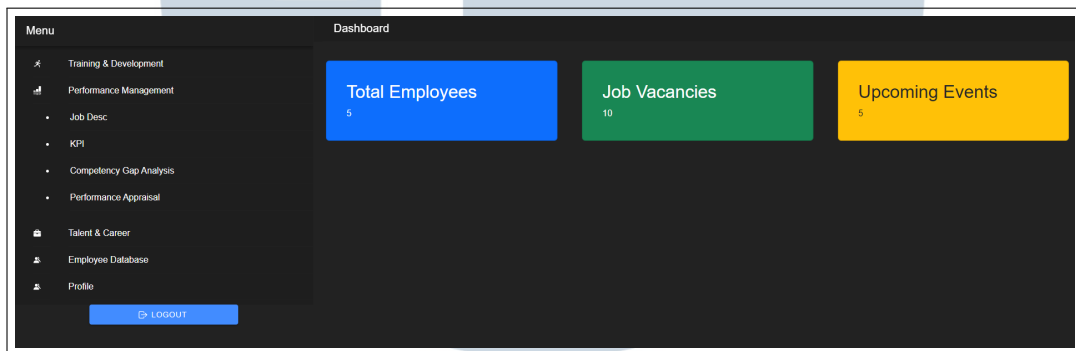


The image shows a register form with a dark background. At the top, the word "Register" is written in white. Below it are four input fields: "Employee ID", "Username", "Password", and "Role", all with yellow borders. A yellow "Sign Up" button is positioned below the "Role" field. At the bottom, there is a "Login" link in white text.

Gambar 3.8. Formulir sign-up pengguna aplikasi HRIS PT. Zeelandia

Gambar 3.7 dan gambar 3.8 menunjukkan tampilan login form serta sign-up form pada aplikasi HRIS PT. Zeelandia Indonesia didesain dengan sederhana dan intuitif. Login form menyediakan *input field* untuk *username* dan *password*, serta tombol "Login". Sign-up form juga sederhana dengan kolom input untuk *username*, *password*, dan informasi pendaftaran lainnya, dilengkapi tombol "Sign Up". Desainnya menekankan keterbacaan dan kemudahan penggunaan, memberikan pengalaman yang baik bagi pengguna.

C. Tampilan dashboard dan konten aplikasi HRIS PT. Zeelandia



Gambar 3.9. Dashboard aplikasi HRIS PT. Zeelandia

Dashboard dalam aplikasi HRIS PT. Zeelandia Indonesia seperti pada gambar 3.9, yang merupakan halaman awal setelah user berhasil login, memberikan user akses terhadap berbagai menu dan informasi penting. Melalui tampilan ini, user dapat dengan mudah memilih menu-menu yang terdapat pada sidebar. Sidebar menyajikan navigasi memungkinkan user untuk menjelajahi dan mengakses fitur-fitur yang relevan, seperti penilaian KPI, Behavior, dan informasi karyawan lainnya. Dengan demikian, dashboard bertujuan memberikan pengalaman pengguna yang efisien dan menyediakan akses cepat ke berbagai fungsi utama aplikasi.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.10. Tampilan *user profile*

Menu "Profile" seperti pada gambar 3.10 memberikan *user* akses penuh untuk melihat informasi terkait profil pribadi mereka. Di dalam menu ini, user dapat dengan mudah melihat berbagai data yang relevan dengan identitas dan posisi mereka di perusahaan. Informasi yang tersedia meliputi nama lengkap, posisi jabatan, nomor identifikasi karyawan, departemen tempat bekerja, serta nama supervisor 1 dan 2. Hal ini memberikan kemudahan bagi user untuk memeriksa dan memverifikasi data pribadi mereka secara cepat, memastikan akurasi informasi, dan merasa terhubung dengan struktur organisasi di sekitar mereka.

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Employee Database

+
Search

No	ID	Name	Gender	Position	Actions	Evaluation
1	1001	jezreel	male	magang	EDIT DELETE	PA CGA
2	1002	Jeri	male	staff	EDIT DELETE	PA CGA
3	1003	Jeriko	male	staff	EDIT DELETE	PA CGA
4	1004	Jessica	female	manager	EDIT DELETE	PA CGA
5	1005	Jessi	female	G. manager	EDIT DELETE	PA CGA
6	1006	Gery Susanto	male	manager	EDIT DELETE	PA CGA
7	1007	Jason Kurnaia	male	magang	EDIT DELETE	PA CGA
8	1008	Jerry Kadroinisa	male	Developer	EDIT DELETE	PA CGA

Page 1 of 3 PREVIOUS 1 2 3 NEXT

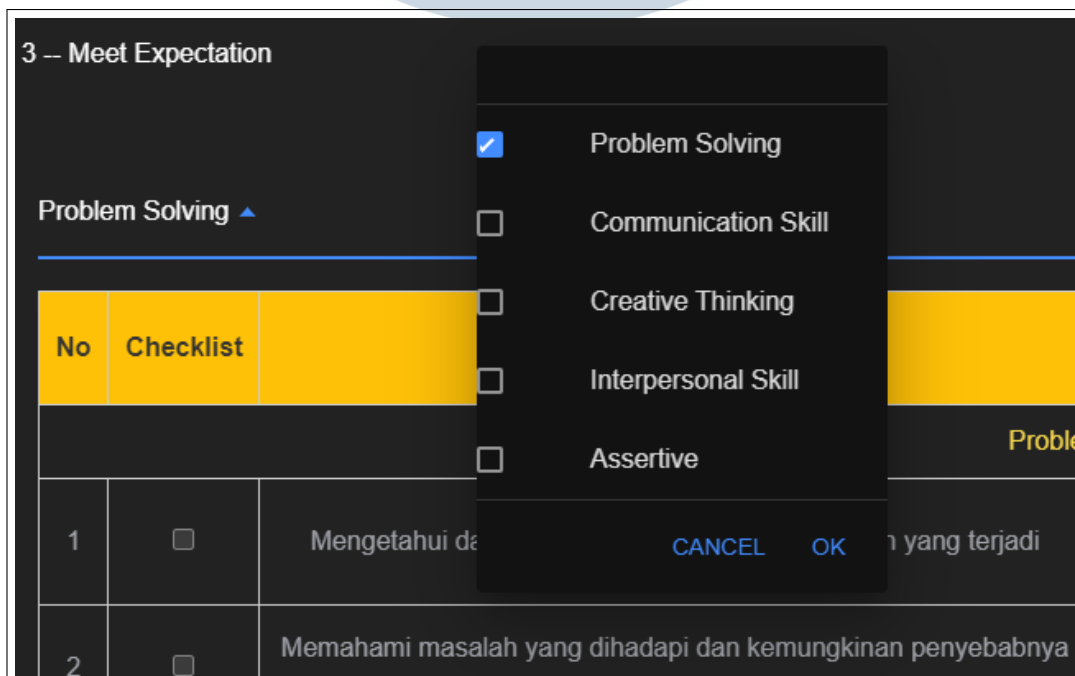
Gambar 3.11. Tampilan Basis data karyawan PT. Zeelandia Indonesia

Tampilan basis data karyawan pada gambar 3.11 memiliki peran utama dalam menyediakan platform pendataan karyawan yang terintegrasi. Basis data ini tidak hanya berfungsi sebagai tempat penyimpanan informasi dasar karyawan, seperti nama, jabatan, dan departemen, tetapi juga menjadi fondasi untuk pelaksanaan penilaian karyawan. Dengan basis data ini, sistem dapat dengan efisien mengakses dan memanfaatkan data karyawan yang diperlukan selama proses penilaian. Dengan kata lain, tampilan basis data karyawan ini memungkinkan integrasi yang lancar antara data karyawan dan fungsi penilaian, menciptakan pendekatan yang terkoordinasi dan terpusat untuk manajemen sumber daya manusia.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada gambar 3.12 dan gambar 3.13, terdapat dua buah tabel yakni tabel target kerja yang menilai KPI dari karyawan dan tabel *Behavior* yang menilai *soft competencies* dan *technical competencies* masing-masing karyawan. Tabel penilaian target kerja digunakan untuk merekam dan menilai kinerja karyawan berdasarkan indikator kinerja kunci yang telah ditentukan. Tabel ini mencakup parameter-parameter seperti produktivitas, pencapaian target, dan kualitas kerja. Tabel akan menerima input KRA, KPI, Target, Bobot, dan Pencapaian dari *user*. Kolom nilai akan melakukan pengambilan keputusan *scoring* berdasarkan nilai dari kolom pencapaian.

Sementara itu, tabel penilaian *Behavior* digunakan untuk menilai aspek perilaku karyawan di tempat kerja, termasuk kemampuan berkomunikasi, kerjasama tim, dan etika profesional. Tabel ini menerima input bobot dan nilai dari setiap kompetensi, dari setiap kategori, hanya satu kompetensi yang dinilai sesuai dengan *level* dari karyawan yang dievaluasi, *user* mampu memilih kompetensi mana yang hendak dinilai dengan memilih kolom checkbox pada baris kompetensi yang ingin dipilih. Kedua tabel ini berperan penting dalam menyediakan gambaran holistik tentang kinerja dan kontribusi karyawan terhadap perusahaan.



Gambar 3.14. Pilihan kompetensi pada tabel penilaian *Behavior*

Gambar 3.14 menunjukkan bahwa tabel penilaian Behavior memiliki fitur yang memungkinkan pengguna untuk memilih kompetensi atau aspek perilaku yang akan dinilai. Dengan adanya opsi ini, pengguna dapat menyesuaikan fokus

penilaian sesuai dengan kebutuhan dan tujuan evaluasi yang spesifik. Kemampuan untuk memilih kompetensi yang akan dinilai memberikan fleksibilitas kepada pengguna dalam menentukan aspek-aspek perilaku tertentu yang ingin ditekankan atau dievaluasi selama proses penilaian. Hal ini memberikan pendekatan yang lebih terpersonal dan disesuaikan dalam mengevaluasi aspek perilaku karyawan.

Problem Solving, Communication Skill						
No	Checklist	Pertanyaan	Bobot (%)	Penilai 1	Rata-rata	Nilai x Bobot
Problem Solving						
1	<input type="checkbox"/>	Mengetahui dan mampu mengidentifikasi masalah yang terjadi	0		0.00	0.00
2	<input type="checkbox"/>	Memahami masalah yang dihadapi dan kemungkinan penyebabnya dengan membuat diagnosa masalah	0		0.00	0.00
3	<input type="checkbox"/>	Mampu mengembangkan standar dan alternatif solusi atau mendesain solusi baru dari masalah yang dihadapi	0		0.00	0.00
4	<input type="checkbox"/>	Mampu memilih, membuat kemudian mengimplementasikan solusi yang tepat	0		0.00	0.00
5	<input type="checkbox"/>	Mampu menganalisa resiko yang mungkin ditimbulkan dan mengembangkan rencana antisipasinya	0		0.00	0.00
Communication Skill						

Gambar 3.15. Formulir *Performance Appraisal* jika terdapat satu penilai

Problem Solving, Communication Skill							
No	Checklist	Pertanyaan	Bobot	Penilai 1	Penilai 2	Rata-rata	Nilai x Bobot
Problem Solving							
1	<input type="checkbox"/>	Mengetahui dan mampu mengidentifikasi masalah yang terjadi	0			0.00	0.00
2	<input type="checkbox"/>	Memahami masalah yang dihadapi dan kemungkinan penyebabnya dengan membuat diagnosa masalah	0			0.00	0.00
3	<input type="checkbox"/>	Mampu mengembangkan standar dan alternatif solusi atau mendesain solusi baru dari masalah yang dihadapi	0			0.00	0.00
4	<input type="checkbox"/>	Mampu memilih, membuat kemudian mengimplementasikan solusi yang tepat	0			0.00	0.00
5	<input type="checkbox"/>	Mampu menganalisa resiko yang mungkin ditimbulkan dan mengembangkan rencana antisipasinya	0			0.00	0.00
Communication Skill							

Gambar 3.16. Formulir *Performance Appraisal* jika terdapat dua penilai

Dalam konteks penilaian karyawan di Zeelandia, terdapat dua skenario yang perlu diperhatikan karena variasi dalam jumlah supervisor yang dimiliki oleh karyawan. Ada karyawan yang memiliki satu supervisor dan ada juga yang memiliki dua supervisor. Oleh karena itu, perlu dilakukan pengecekan data karyawan untuk memastikan jumlah supervisor yang dimiliki. Jika seorang

karyawan memiliki dua supervisor, maka tabel perilaku (*behavior*) yang muncul akan mengakomodasi kedua penilai dengan menampilkan dua kolom penilai (dapat dilihat pada gambar 3.16). Dengan pendekatan ini, sistem dapat menanggapi perbedaan struktur organisasional di Zeelandia dan memberikan fleksibilitas yang dibutuhkan dalam proses penilaian karyawan.

Selain mengembangkan bagian tampilan atau *front-end* dari aplikasi HRIS PT. Zeelandia, pengembangan *back-end* aplikasi turut dilaksanakan. Tugas tersebut mencakup pemenuhan kebutuhan fitur yang diminta oleh *user* atau pengguna aplikasi. Dengan fokus pada *back-end*, mahasiswa magang bertanggung jawab dalam merancang, mengimplementasikan, dan memastikan keberlanjutan fungsi-fungsi penting di belakang layar aplikasi. Hal ini mencakup pengaturan database, manajemen API, serta proses otentikasi pengguna. Kolaborasi antara *front-end* dan *back-end* menjadi kunci dalam memastikan bahwa aplikasi HRIS PT. Zeelandia memberikan pengalaman pengguna yang menyeluruh dan memenuhi kebutuhan bisnis yang diinginkan.

```
1  const handleButtonClick = () => {
2    if (data.supervisor2id == null) {
3      setShowContent1(true);
4      setShowContent2(false);
5    } else {
6      setShowContent2(true);
7      setShowContent1(false);
8    }
9    return "";
10 };
11
12 useEffect(() => {
13   handleButtonClick();
14 }, [data.supervisor2id]);
15
16 // contoh return
17 {showContent2 && (
18   <div style={{ marginRight: "20px", marginLeft: "20px",
19     marginTop: "5%" }}>
20     <h4> II. Behavior </h4>
21     <h5>Ketentuan penilaian target kerja sebagai
22     berikut:</h5>
23     </div>
24     <div className="ketentuan-nilai">
```

24
25

```
...  
    )}
```

Kode 3.1: Potongan kode penentu kolom penilai

Potongan kode 3.1 mencakup fungsi `handleButtonClick`, yang bertanggung jawab untuk menentukan tampilan berdasarkan kondisi tertentu dalam aplikasi. Saat tombol diklik, fungsi ini mengevaluasi nilai `data.supervisor2id`. Jika `data.supervisor2id` tidak terdefinisi (`null`), variabel `showContent1` diatur menjadi `true`, sedangkan `showContent2` diatur menjadi `false`. Sebaliknya, jika `data.supervisor2id` terdefinisi, `showContent2` diatur menjadi `true`, dan `showContent1` menjadi `false`.

Selanjutnya, pada `useEffect`, fungsi `handleButtonClick` dipanggil setiap kali terjadi perubahan pada `data.supervisor2id`. Hal ini memastikan bahwa tampilan aplikasi selalu sesuai dengan kondisi terkini. Penggunaan kondisi ini diilustrasikan dalam blok `return`, di mana elemen HTML hanya akan dirender jika `showContent2` memiliki nilai `true`.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

No	Competency	Req Level	Current Level	Gap
Soft				
1	Leadership	4	3	1
2	Analytical thinking	3	2	1
3	Communication skill (incl. English)	3	1	2
4	Customer orientation	4	2	2
5	Assertive	4	2	2
Technical				
6	Warehouse technical knowledge	4	2	2
7	ERP knowledge	4	1	3
8	Computer literacy (Ms. Office, etc)	4	2	2
9	Data accuracy	4	2	2
10	Reporting skill	3	2	1
11	Business Process Knowledge	3	2	1
12	Product & Raw Material knowledge	4	2	2
13	QMS, Government & Corporate regulation (ISO, SJH, HACCP, GMP, Safety)	4	2	2

Kirim Penilaian

Gambar 3.17. Formulir *Competency Gap Analysis* aplikasi

Sebagai analisis kesenjangan kompetensi, CGA (Competency Gap Analysis) di aplikasi HRIS PT. Zeelandia Indonesia memiliki peran kunci dalam menilai perbedaan antara kompetensi yang dimiliki oleh karyawan dan kompetensi yang diinginkan atau diperlukan oleh perusahaan. Form CGA memberikan gambaran detail tentang sejauh mana karyawan telah mencapai kompetensi yang diharapkan dan mengidentifikasi area-area di mana perbaikan atau pengembangan lebih lanjut diperlukan. Melalui CGA, perusahaan dapat membuat rencana pengembangan karyawan yang lebih terarah dan sesuai dengan kebutuhan organisasi. Data hasil CGA ini juga dapat menjadi dasar untuk pengambilan keputusan terkait pelatihan dan pengembangan sumber daya manusia.

			employeeid	username	role	password	
<input type="checkbox"/>	Edit	Copy	Delete	1001	jezreel	manager	\$2y\$10\$bhr5aGkz2KaqGsgnnOEnSOQ7S.CQqu1ruh.fxBSDeMm...
<input type="checkbox"/>	Edit	Copy	Delete	1007	leo	employee	\$2y\$10\$D1sR0dUZCSmTjYXHvyAFmenii3SEVP08Y3WWL.ouzWJ...
<input type="checkbox"/>	Edit	Copy	Delete	1008	lili	employee	\$2y\$10\$KCy2h6/MQq9hZN9cxJz04.bUXjgrgf7NCu9NAUuPE5s...
<input type="checkbox"/>	Edit	Copy	Delete	1009	lolo	employee	\$2y\$10\$WKjJ2oT1mSuxn9gDiY1OBOJwYdk/j7kU4J4tkTMHgl...
<input type="checkbox"/>	Edit	Copy	Delete	1111	lolo	employee	\$2y\$10\$.7wYAK8g11moEwckFYpxx.O9igWH7RBzkPLs7rDK7V...
<input type="checkbox"/>	Edit	Copy	Delete	1112	lolo	employee	\$2y\$10\$IsQ6/i0p7PcT5jpmHWHInuy1GAJVPUWSsQWK2xcBccv...
<input type="checkbox"/>	Edit	Copy	Delete	1123	spesial	employee	\$2y\$10\$Br14eHyYTWEgtD3UdC0QCebMakaCTgmFw/MwHLSf1...
<input type="checkbox"/>	Edit	Copy	Delete	1222	reee	employee	\$2y\$10\$XxZmrcb0lc vazDttjysUguCDi5ImqYIt6p9yXToSj4o...
<input type="checkbox"/>	Edit	Copy	Delete	10045	sfsf	employee	\$2y\$10\$Z3sv46Nm7c.hkyiDqREK2uadUgQk3V3hnnTVDIsprnd...
<input type="checkbox"/>	Edit	Copy	Delete	10046	sfsf	employee	\$2y\$10\$Sq.E3cT9EhhRO8TbVFLjOOmwrfsE8WgnYFxxGG/vL9jj...
<input type="checkbox"/>	Edit	Copy	Delete	1006	gerysusanto	manager	\$2y\$10\$Yl2lmkPvWAb8SpjP/dY5U.1m5fuxdnaQzhkmp/yRuu4...

Gambar 3.18. *Back-end* basis data user

Tabel user pada database aplikasi HRIS PT. Zeelandia Indonesia memiliki beberapa kolom kunci, yaitu `employeeid`, `username`, `role`, dan `password` (dalam bentuk *hash*). Kolom `employeeid` digunakan sebagai identifikasi unik untuk setiap karyawan, `username` adalah nama pengguna yang digunakan untuk login, `role` menunjukkan peran atau hak akses pengguna (seperti *admin* atau *user*), dan kolom `password` berisi *hash* dari kata sandi pengguna untuk keamanan. Tabel ini akan digunakan oleh API dari form login untuk pengecekan input login form pada aplikasi HRIS.

Dengan struktur tabel ini, sistem dapat mengelola informasi pengguna dengan efisien dan memberikan kontrol akses yang sesuai berdasarkan peran masing-masing pengguna. Pengguna dapat melakukan login dengan menggunakan kombinasi `username` dan `password`, dan peran (`role`) akan menentukan hak akses serta fungsionalitas yang dapat diakses oleh pengguna. Selain itu, penggunaan *hash* pada kolom `password` meningkatkan tingkat keamanan data, menghindari penyimpanan kata sandi dalam bentuk teks biasa.

```

1 function register(request $req) // menambahkan user ke tabel user
  pada basis data
2   {
3     $user = new User;
4     $user->employeeid= $req->input('employeeid ');
5     $user->username= $req->input('username ');
6     $user->role= $req->input('role ');
7     $user->password= Hash::make($req->input('password '));
8     $user->save();
9     return response()->json(['message' => 'Registration
  successful', 'user' => $user], 201);

```



```

10 }
11
12 public function login(Request $req) // mengecek input user
    apakah data ada di basis data
13 {
14     $user = User::where('employeeid', $req->username)->first();
15     if (!$user || !Hash::check($req->password, $user->password)){
16         return ["Error" => "Sorry, email or password invalid"];
17     }
18     return $user;
19 }

```

Kode 3.2: Potongan kode fungsi Laravel untuk fitur login dan register

Potongan kode 3.2 merupakan implementasi registrasi (*register*) dan autentikasi (*login*) dalam aplikasi menggunakan Laravel. Registrasi memproses data pengguna baru, validasi, dan penyimpanan ke basis data. Autentikasi membandingkan username dan password dengan data di basis data untuk memberikan akses sesuai peran pengguna. Laravel mempermudah pengembangan dan menawarkan keamanan, termasuk *hashing* untuk kata sandi.

Pada baris 16 potongan kode 3.2, menunjukkan pesan yang ditampilkan jika terdapat username atau password yang salah atau tidak sesuai dengan hasil query pada variabel \$user. Pesan yang ditampilkan adalah *"Sorry, email or password invalid"*.

```

1 // contoh penggunaan API dalam login form
2
3 async function login() {
4     let item = { username, password };
5     let result = await fetch("http://localhost:8000/api/login", {
6         method: "POST",
7         headers: {
8             "Content-Type": "application/json",
9             Accept: "application/json",
10        },
11        body: JSON.stringify(item),
12    });
13    result = await result.json();
14    if (JSON.stringify(result) !== JSON.stringify({ Error: "Sorry,
15        email or password invalid" }))
16        // mengecek jika password tidak sama dengan error, maka step
        dalam kurung if dijalankan
        {

```

```

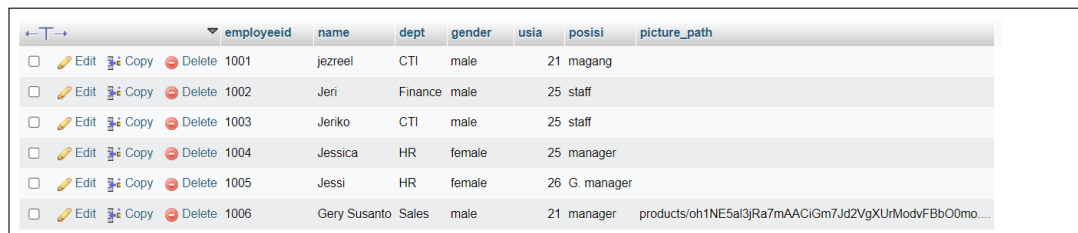
17     localStorage.setItem("user-info", JSON.stringify(result));
18     // menyimpan user-info di dalam local storage
19   }else{
20     setShowAlert(!showAlert);
21   }
22 }

```

Kode 3.3: Potongan kode penggunaan API pada fitur login

Fungsi login pada 3.3 merupakan bagian dari implementasi otentikasi pengguna pada aplikasi. Fungsi ini menggunakan metode asinkron untuk menangani operasi yang memerlukan waktu, seperti pengiriman permintaan HTTP POST ke endpoint "http://localhost:8000/api/login". Dalam prosesnya, fungsi ini membuat objek item yang berisi informasi pengguna seperti username dan password. Setelahnya, permintaan tersebut dikirimkan menggunakan fungsi fetch, dan hasilnya diuraikan sebagai respons JSON.

Selanjutnya, fungsi ini melakukan pengecekan pada respons dari server. Jika respons tidak mengandung pesan error yang menandakan kegagalan login, informasi pengguna disimpan dalam local storage dengan kunci "user-info". Sebaliknya, jika respons mengandung pesan error, fungsi setShowAlert dipanggil untuk menampilkan pemberitahuan ke pengguna.



	employeeid	name	dept	gender	usia	posisi	picture_path
<input type="checkbox"/>	1001	jezreel	CTI	male	21	magang	
<input type="checkbox"/>	1002	Jeri	Finance	male	25	staff	
<input type="checkbox"/>	1003	Jeriko	CTI	male	25	staff	
<input type="checkbox"/>	1004	Jessica	HR	female	25	manager	
<input type="checkbox"/>	1005	Jessi	HR	female	26	G. manager	
<input type="checkbox"/>	1006	Gery Susanto	Sales	male	21	manager	products/oh1NE5al3jRa7mAACiGm7Jd2VgXUrModvFBb00mo...

Gambar 3.19. Back-end basis data *employee*

Basis data *employee* pada aplikasi HRIS PT. Zeelandia Indonesia berfungsi untuk menyimpan informasi karyawan. Tabel *employee* terdiri dari kolom-kolom seperti *employeeid*, *name*, *position*, *department*, dan sebagainya. Data ini mencakup detail pribadi karyawan, posisi, departemen, dan informasi relevan lainnya. Basis data *employee* digunakan sebagai sumber data utama untuk melakukan penilaian karyawan dan pendataan.

	id	questionid	employeeid	supervisor1id	supervisor2id	nilai1	nilai2	timestamp1	timestamp2	status
<input type="checkbox"/>	195	25	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	194	24	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	193	23	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	192	22	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	191	21	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	190	20	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	189	19	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	188	18	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	187	17	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	186	16	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	185	15	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	184	14	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	183	13	1002			1	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	182	12	1002			2	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	181	11	1002			3	1	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	180	10	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	
<input type="checkbox"/>	179	9	1002			3	3	2023-10-31 13:17:30	2023-10-31 13:17:30	

Gambar 3.20. *Back-end* basis data respon dari formulir *Performance Appraisal*

Tabel basis data yang menampung respons dari formulir penilaian pada aplikasi HRIS PT. Zeelandia Indonesia berfungsi untuk menyimpan hasil penilaian karyawan. Tabel ini mencakup kolom-kolom seperti id, employeeid, questionid, nilai, dan lainnya. Setiap respons dari karyawan terhadap pertanyaan penilaian akan disimpan dalam tabel ini. Informasi tersebut nantinya digunakan untuk menganalisis dan menghasilkan laporan evaluasi kinerja karyawan.



```
Route::post('register', [SomeController::class,'register']);
Route::post('login', [SomeController::class, 'login']);
Route::post('addEmployee', [EmployeeController::class,'addEmployee']);
Route::get('list/{employeeid}', [EmployeeController::class,'list']);
Route::get('listAll/', [EmployeeController::class,'allEmployee']);
Route::delete('delete/{employeeid}', [EmployeeController::class,'delete']);
Route::get('employee/{employeeid}', [EmployeeController::class, 'getEmployee']);
Route::put('updateemployee/{employeeid}', [EmployeeController::class, 'updateEmployee']);
Route::post('addResponse', [PaController::class,'addResponse']);
Route::get('/pa/questions', [PaController::class,'getQuestions']);
Route::post('/behavior/final', [PaController::class,'updateStatusFinal']);
Route::post('/kpi/final', [KpiController::class,'updateStatusFinal']);
Route::post('/pinalti/final', [PinaltiController::class,'updateStatusFinal']);
Route::get('/response_pa/{employeeid}', [PaController::class, 'getResponseByEmployee']);
Route::post('/kpi/addResponse', [KpiController::class,'addResponseKPI']);
Route::get('/response_kpi/{employeeid}', [KPIController::class, 'getKPIResponseByEmployee']);
Route::post('/pinalti/addResponse', [PinaltiController::class,'addResponsePinalti']);
Route::get('/response_pinalti/{employeeid}', [PinaltiController::class, 'getPinaltiResponseByE
Route::get('pdp', [PdpController::class,'pdp_indicators']);
Route::post('/pdp/addResponse', [PdpController::class,'addResponse']);
Route::get('/response_pdp/{employeeid}', [PdpController::class, 'getResponseByEmployee']);
Route::post('/pdp/kpi/addResponse', [PdpController::class,'addResponseKpi']);
Route::get('/pdp/response_kpi/{employeeid}', [PdpController::class, 'getKPIResponseByEmployee'
Route::get('/cga/questions', [CgaController::class,'getQuestions']);
Route::post('/cga/addResponse', [CgaController::class,'addResponse']);
Route::get('/response_cga/{employeeid}', [CgaController::class, 'getResponseByEmployee']);
Route::get('/user/{username}', [SomeController::class, 'getUserData']);
```

Gambar 3.21. routing API yang dibangun menggunakan Laravel

Gambar 3.21 menunjukkan kode pengatur API *routing*. API routing adalah mekanisme dalam pengembangan suatu aplikasi yang memungkinkan pengembang menentukan bagaimana permintaan HTTP harus ditangani oleh aplikasi web atau layanan backend. Ini melibatkan penentuan rute atau jalur tertentu yang sesuai dengan permintaan pengguna dan menentukan tindakan atau respons yang akan diambil sebagai tanggapan sesuai dengan fungsi yang digunakan dalam *controller*. Jika route yang digunakan pada *front-end* tidak sesuai dengan route pada gambar 3.21 maka error "404 NOT FOUND" ditampilkan ke layar atau *console*, menunjukkan bahwa API yang dituju tidak tersedia.

3.4.2 Kendala

1. Tidak Adanya Tim dalam Proyek Magang

Tidak adanya tim untuk bekerja sama dalam proyek magang, sehingga penanggung jawab mengemban peran sebagai full-stack developer, system analyst, dan tester secara bersamaan.

2. Deadline Proyek yang Singkat

Batas waktu proyek yang relatif singkat menuntut penanganan yang cermat

untuk menyelesaikan pekerjaan dengan tepat waktu.

3. Kebutuhan yang Banyak dari Pihak HR

Adanya banyak kebutuhan dan permintaan fitur dari pihak HR menimbulkan tantangan dalam pengelolaan prioritas dan waktu.

4. Perubahan Requirements dan Fitur yang Mendadak

Adanya perubahan requirements atau penambahan fitur secara mendadak menuntut respons yang cepat untuk menjaga kelancaran proyek.

5. Tipe Data yang Berbeda-beda dalam Tabel Penilaian

Perbedaan tipe data antar tabel penilaian, termasuk penggunaan master list kompetensi dan input langsung dari pengguna, memerlukan pendekatan yang teliti.

6. SOP yang Berbeda-beda dari Setiap Departemen

Departemen-departemen yang berbeda memiliki SOP penilaian yang berbeda, dan beberapa karyawan sudah menjadi supervisor, menimbulkan kompleksitas dalam penyesuaian sistem.

7. Miskomunikasi terhadap SOP dari Pihak HR

Miskomunikasi terkait beberapa Standard Operating Procedures (SOP) dari pihak HR menjadi kendala yang membutuhkan solusi cepat.

3.4.3 Solusi

1. Intensifikasi Upaya Belajar dan Peningkatan Kemampuan

Respon terhadap ketiadaan tim dengan mengintensifkan upaya belajar, meningkatkan kemampuan, dan memastikan keakuratan setiap tahapan pengembangan fitur serta logika pengiriman data.

2. Penyusunan Jadwal Pengerjaan yang Terstruktur

Dalam menghadapi deadline proyek yang singkat, penyusunan jadwal pengerjaan yang terstruktur menjadi solusi untuk memastikan setiap tahapan proyek dapat diselesaikan sesuai waktu yang ditentukan.

3. Prioritisasi Pengerjaan Berdasarkan Urgensi

Dengan banyaknya kebutuhan dari pihak HR, solusi yang diambil adalah membagi kebutuhan ke dalam kategori urgensi dan memprioritaskan pengerjaan pada yang paling mendesak terlebih dahulu.

4. Respons Cepat terhadap Perubahan dan Penambahan Fitur

Respon cepat terhadap perubahan requirements atau penambahan fitur dilakukan dengan segera mengerjakan revisi atau implementasi fitur baru setelah menerima pemberitahuan.

5. Komunikasi Intensif dengan Tim HR untuk Penentuan Tipe Data

Komunikasi intensif dengan tim HR untuk menentukan field mana yang berasal dari input dan mana yang bersumber dari master list. Selanjutnya, bekerja lebih keras untuk menyesuaikan data yang dikirim ke database agar sesuai dengan kebutuhan unik setiap tabel penilaian.

6. Diskusi dan Uji Coba dengan Pihak HR

Terus melakukan diskusi dan uji coba dengan pihak HR untuk memastikan SOP dan panduan sistem dapat disesuaikan dengan variasi departemen dan posisi karyawan yang menjadi supervisor.

7. Perbaiki Kode Fitur yang Terkait dengan SOP

Miskomunikasi terkait SOP dari pihak HR dapat diatasi dengan memperbaiki kode fitur yang terkait, terutama dalam hal akses, agar sesuai dengan SOP yang berlaku.

