BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama program magang di PT Global Loyalty Indonesia, peran yang dijalankan adalah sebagai *Backend Web Developer* di divisi *IT Corporate*, yang beroperasi langsung di bawah departemen *IT Development*.

Dalam peran ini, arahan khusus diberikan oleh *Supervisor* sekaligus *Junior Manager*, dengan pendampingan oleh seorang mentor pribadi yang merupakan salah satu karyawan pada divisi *IT Corporate*.

Proses kerja dan komunikasi tim sangat terstruktur. Diskusi dan koordinasi rutin dilakukan melalui berbagai metode, mulai dari pertemuan tatap muka di kantor, *update* progres dalam *weekly meeting*, hingga komunikasi intensif via grup atau *chat* personal *WhatsApp*.

Untuk koordinasi proyek, arahan (*briefing*) diberikan langsung oleh *Project Manager* dan *Supervisor* kepada tim *Developer*, *Front-End*, dan *Back-End*, dengan mengacu pada desain *UI/UX* yang sudah disiapkan sebelumnya menggunakan Figma.

3.2 Tugas dan Uraian Kerja Magang

Selama masa pelaksanaan magang, tugas yang dikerjakan adalah proyek pengembangan sistem berbasis web yang bertujuan untuk mendigitalisasi proses yang sebelumnya dilakukan secara manual oleh divisi *Human Resource* (HR).

Proyek ini menjadi bagian dari upaya perusahaan dalam meningkatkan efisiensi kerja, akurasi data, serta transparansi proses internal melalui penerapan teknologi informasi. Melalui proyek digitalisasi ini, sistem diharapkan mampu mengotomatisasi sebagian besar kegiatan administratif agar menjadi lebih cepat, akurat, dan mudah diakses oleh pihak yang berkepentingan.

3.3 Uraian Pelaksanaan Magang

3.3.1 Pengenalan Lingkungan Kerja

Kegiatan magang dimulai dengan adanya fase pengenalan lingkungan kerja yang diselenggarakan oleh karyawan tetap dan karyawan magang *Human Resource* (HR). Pada fase awal ini, selain adanya pengenalan dan sambutan resmi, dilakukan penyelesaian administrasi dan penandatanganan surat perjanjian magang.

Human Resource memberikan penjelasan mendalam mengenai Global Loyalty Indonesia, mencakup pemaparan materi mengenai latar belakang perusahaan. Dijelaskan bahwa GLI merupakan perusahaan yang bekerja berbasis *knowledge* (data). Selain itu, disampaikan ekspektasi agar implementasi *Artificial Intelligence* (AI) dapat dijadikan sebagai *tool* yang menunjang keefektifan bekerja.

3.3.2 Briefing Proyek oleh Project Manager

Setelah orientasi selesai, kegiatan dilanjutkan ke sesi *briefing* proyek oleh *Project Manager* (PM). Sesi ini mencakup penjelasan *timeline*, alur bisnis proyek, serta visualisasi *mockup* aplikasi menggunakan Figma. Penjelasan *timeline* pengerjaan diuraikan pada Table 3.1.

Table 3.1. Rencana Jadwal Pengerjaan *Website Talent Hive*

Fase	Tugas	Durasi Pengerjaan	Tanggal Mulai	Tanggal Selesai
Development		**39 Hari**	**25-08-2025**	**17-10-2025**
	Profile	5 Hari	25-08-2025	29-08-2025
	Job Description	6 Hari	01-09-2025	09-09-2025
	Competencies	6 Hari	10-09-2025	17-09-2025
	Skill Set	6 Hari	18-09-2025	25-09-2025
	Work History	6 Hari	26-09-2025	03-10-2025
	Recognition	5 Hari	06-10-2025	10-10-2025
	Training	5 Hari	13-10-2025	17-10-2025
QA & UAT		**16 Hari**	**20-10-2025**	**10-11-2025**

3.3.3 Tampilan User Interface (UI) dan Format *Backend*

Visualisasi antarmuka (*UI*) Figma berfungsi sebagai referensi utama dalam pengembangan. Desain ini menjadi landasan diskusi tim, estimasi kebutuhan

data, dan perancangan ERD. Gambar 3.1 menunjukkan struktur direktori dari *template backend* Python FastAPI yang digunakan untuk menjamin standarisasi kode perusahaan.

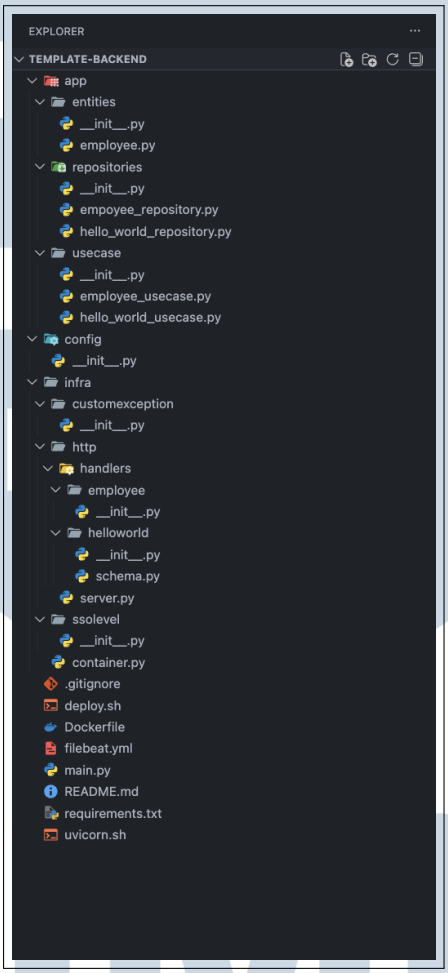

Figure 3.1. Struktur Direktori (*Template Backend*)

3.3.4 Pembuatan Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) dirancang menggunakan *mermaidchart.com* untuk memetakan tabel utama, atribut, dan relasi antar entitas. Rancangan ini divisualisasikan pada Gambar 3.2.

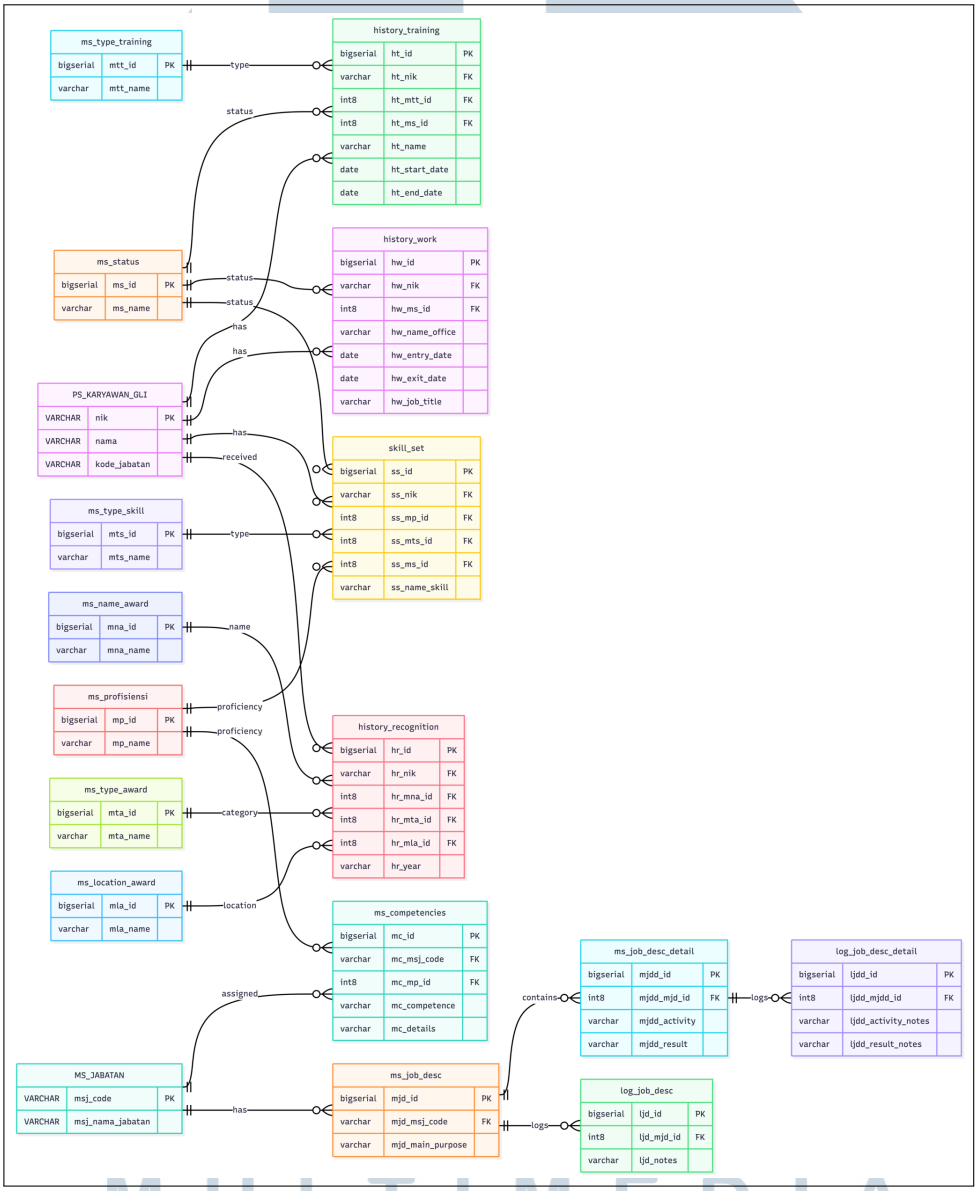

Figure 3.2. *Entity Relationship Diagram*

3.3.5 Tahapan Pengembangan Aplikasi

Pengembangan dilakukan menggunakan Python dan FastAPI, meliputi pembuatan *endpoint*, *query repository*, mekanisme RBAC, *use case*, hingga integrasi SSO. Proyek ini menerapkan tiga level akses utama, yaitu *Employee* (Gambar 3.3), *Manager* (Gambar 3.4), dan *Admin* (Gambar 3.5).

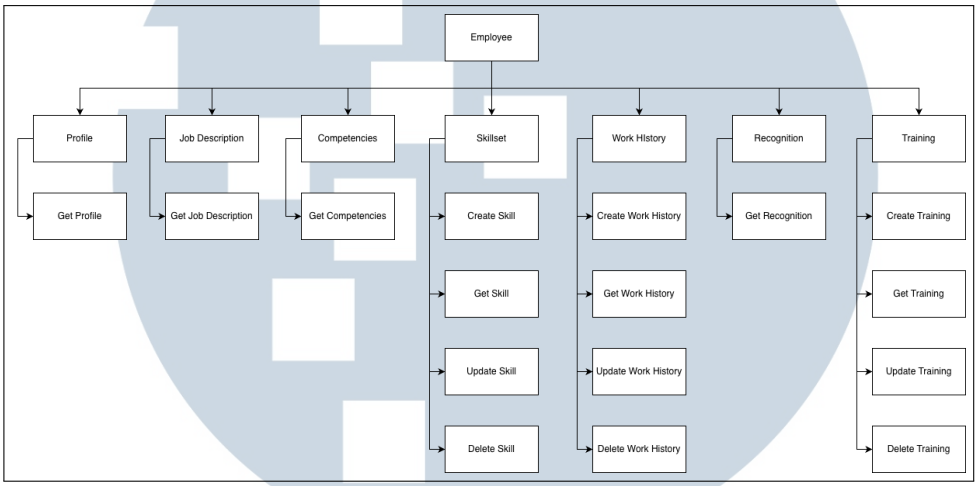

Figure 3.3. Struktur *Endpoint API* pada Level *Employee*

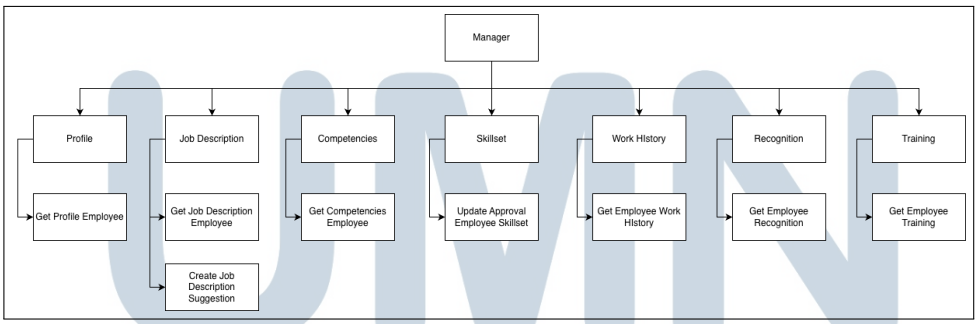

Figure 3.4. Struktur *Endpoint API* pada Level *Manager*

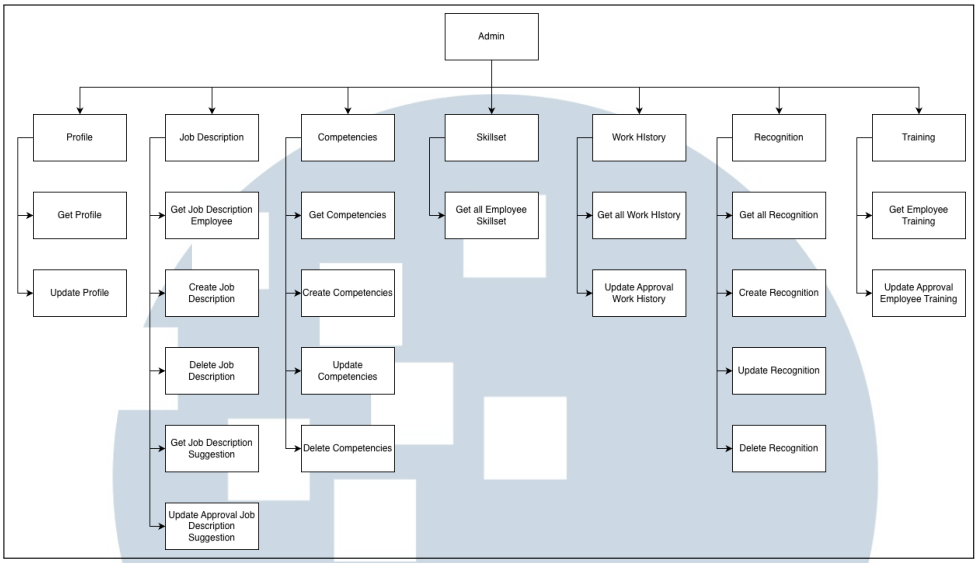

Figure 3.5. Struktur *Endpoint API* pada Level *Admin*

3.3.6 Fungsionalitas Sistem

Sistem terdiri dari tujuh modul utama. Alur kerja utama sistem digambarkan pada Gambar 3.6. Navigasi sistem dimulai dari login SSO hingga pemilihan fitur pada *dashboard*, yang secara prosedural dituangkan dalam Algoritma 1.

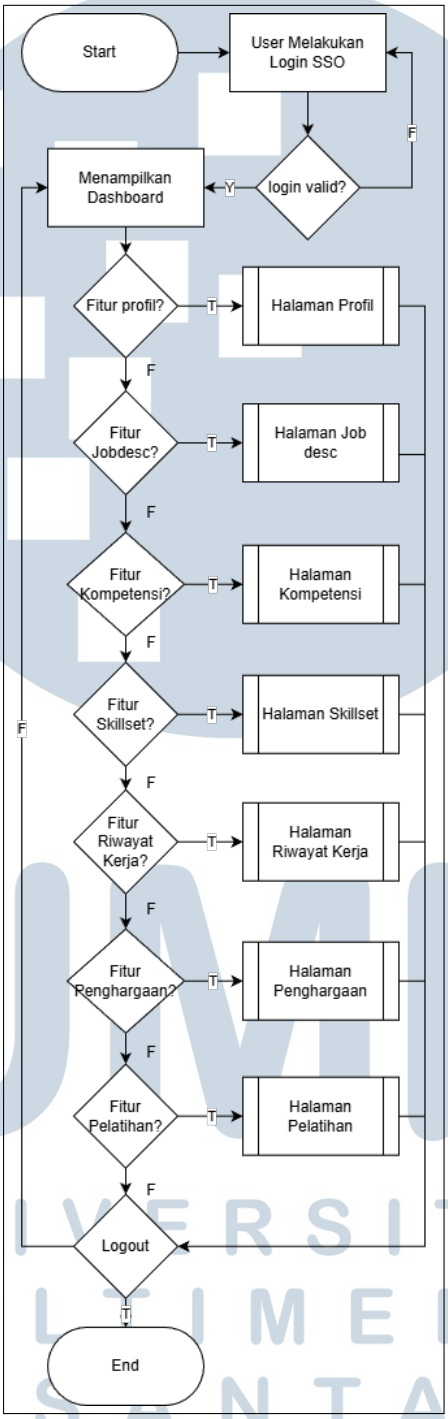

Figure 3.6. *Flowchart* Utama Sistem *Talent Hive*

Algorithm 1 Alur Navigasi Utama Sistem

```
1:  procedure ALURUTAMA
2:      Start
3:      User melakukan login SSO
4:      if login valid then
5:          Menampilkan dashboard
6:          while user belum logout do
7:              User memilih fitur pada dashboard
8:              if fitur = Profil then
9:                  Buka Halaman Profil
10:             else if fitur = Jobdesc then
11:                 Buka Halaman Jobdesc
12:             else if fitur = Kompetensi then
13:                 Buka Halaman Kompetensi
14:             else if fitur = Skillset then
15:                 Buka Halaman Skillset
16:             else if fitur = Riwayat Kerja then
17:                 Buka Halaman Riwayat Kerja
18:             else if fitur = Penghargaan then
19:                 Buka Halaman Penghargaan
20:             else if fitur = Pelatihan then
21:                 Buka Halaman Pelatihan
22:             else if fitur = Logout then
23:                 User logout; End; return
24:             end if
25:         end while
26:     else
27:         Login tidak valid; End
28:     end if
29: end procedure
```

3.3.6.1 Profil (*Profile*)

Fitur Profil berfungsi sebagai pusat data identitas karyawan. Alur kerja modul ini divisualisasikan pada Gambar 3.7 dan logika pengelolaannya dijelaskan dalam Algoritma 2.

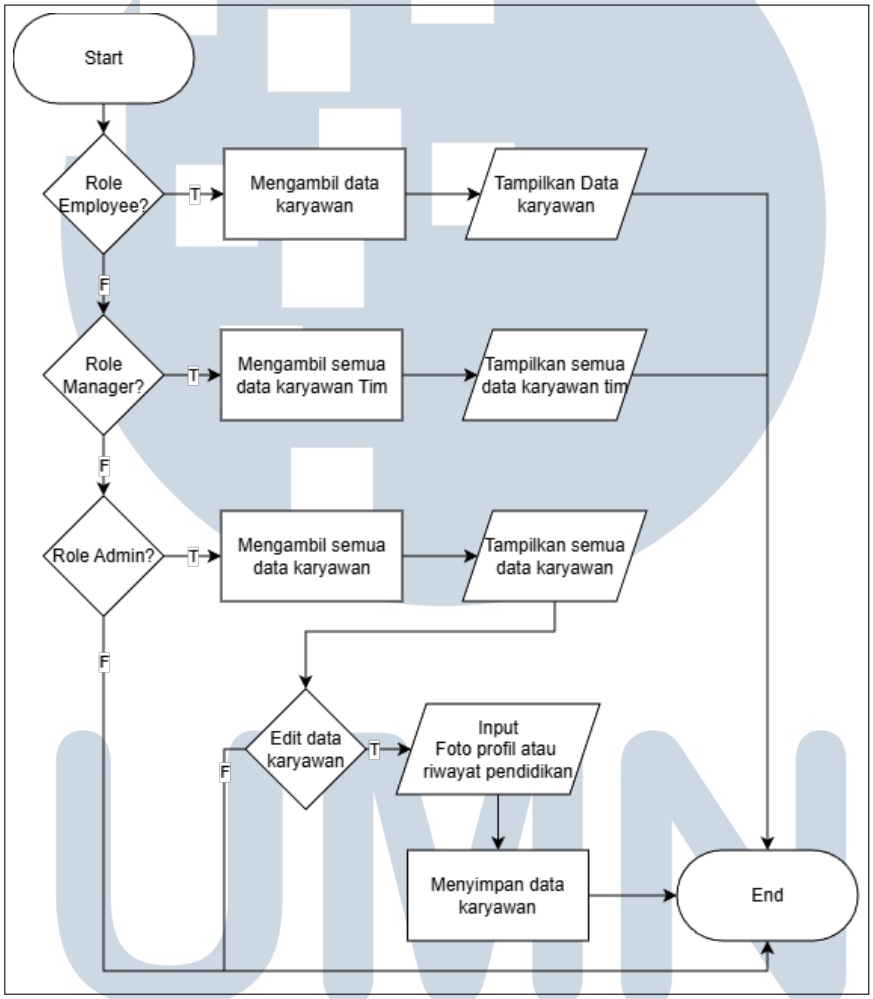

Figure 3.7. *Flowchart* Modul Profil

Pada implementasinya, *role employee* dapat mengambil data pribadi guna memantau status kepegawaian. *Role manager* memiliki fungsi serupa namun terbatas pada data bawahannya. *Role admin* memiliki akses menyeluruh serta berwenang memperbarui foto dan informasi edukasi. Contoh *endpoint* profil oleh *Manager* ditunjukkan pada Gambar 3.8.

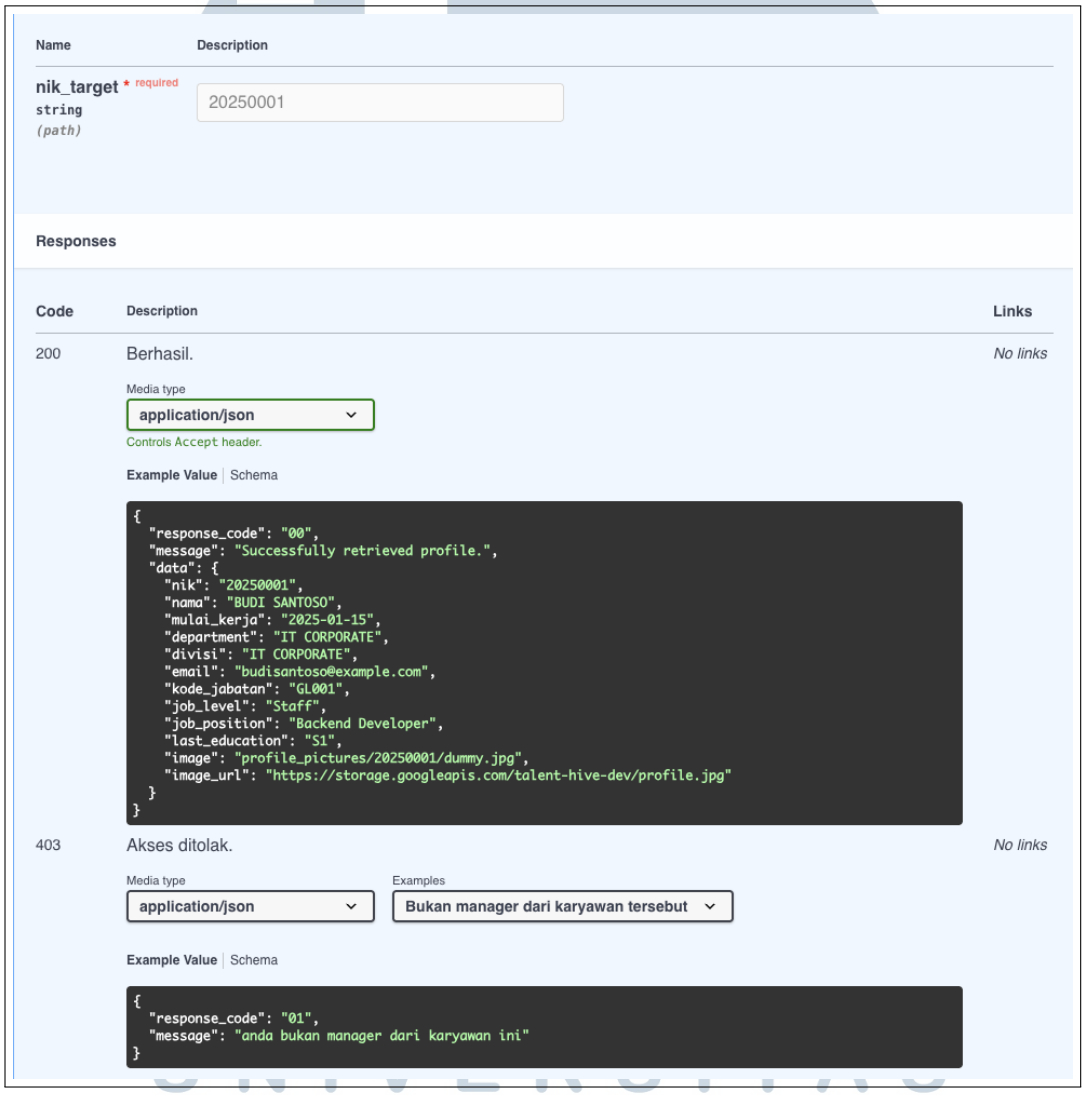

Figure 3.8. Contoh *Endpoint GET Profile* oleh *Manager*

Algorithm 2 Alur Fitur Profil

```
1:  procedure KELOLAPROFIL(role)
2:      Start
3:      if role = Employee then
4:          Mengambil data pribadi
5:      else if role = Manager then
6:          Mengambil data tim
7:      else if role = Admin then
8:          Mengambil semua data
9:      end if
10:     if user memilih edit then
11:         Input data; Simpan data
12:     end if
13:     End
14: end procedure
```

3.3.6.2 Deskripsi Pekerjaan (*Job Description*)

Fitur ini menyediakan informasi tanggung jawab jabatan. *Flowchart* modul ini dapat dilihat pada Gambar 3.9 dan operasional pengerjaannya dituangkan dalam Algoritma 3.

Algorithm 3 Alur Fitur Deskripsi Pekerjaan

```
1:  procedure KELOLAJOBDESCRIPTION(role)
2:      Start
3:      if role = Employee then
4:          Mengambil data jobdesc sendiri
5:      else if role = Manager then
6:          Mengambil data tim
7:      else if role = Admin then
8:          Mengambil semua data
9:      end if
10:     if role = Admin then
11:         if tambah/ubah/hapus then
12:             Proses data; Simpan perubahan
13:         end if
14:     end if
15:     End
16: end procedure
```

Dalam modul ini, *role employee* hanya dapat melihat deskripsi pekerjaan sesuai posisi mereka. *Role manager* dapat melihat deskripsi bawahannya serta

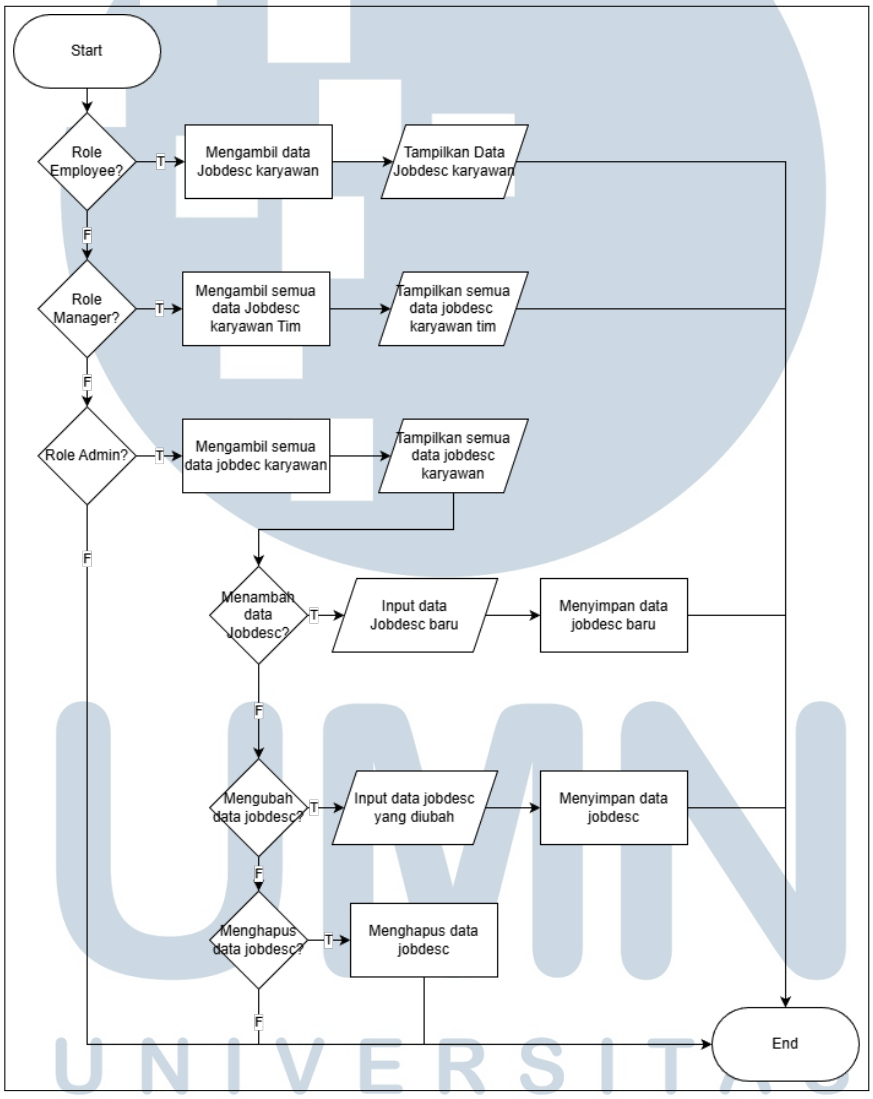

Figure 3.9. *Flowchart* Modul Deskripsi Pekerjaan

mengusulkan perubahan. *Role admin* memiliki kontrol penuh untuk operasional *CRUD* serta validasi usulan *manager*. Contoh pemanggilan API ditunjukkan pada Gambar 3.10.

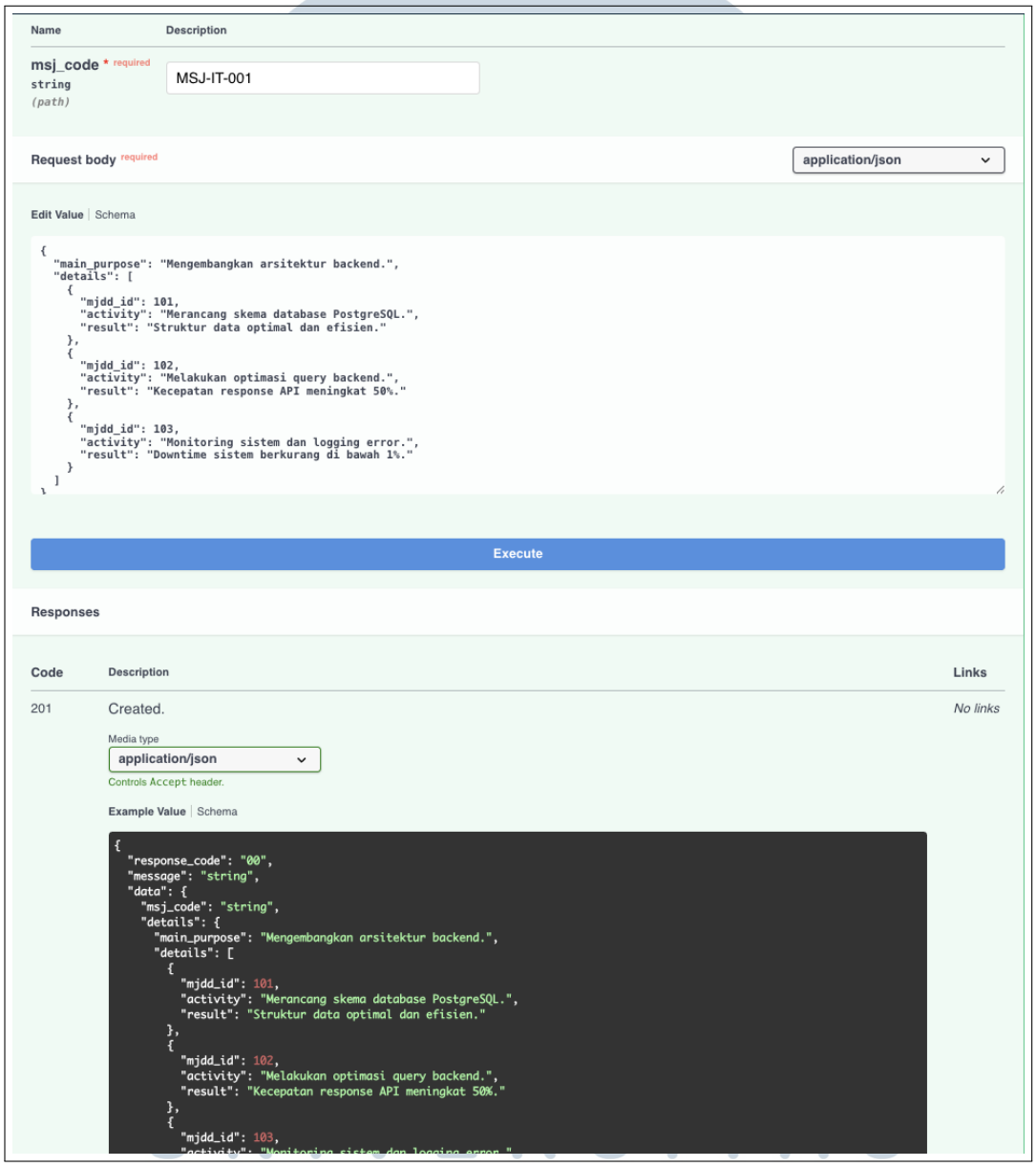

Figure 3.10. Contoh *Endpoint POST Job Description* oleh *Admin*

3.3.6.3 Kompetensi (*Competencies*)

Modul ini digunakan untuk menstandarisasi daftar kompetensi jabatan. Alur kerjanya terdapat pada Gambar 3.11 dan rincian logikanya didefinisikan dalam

Algoritma 4.

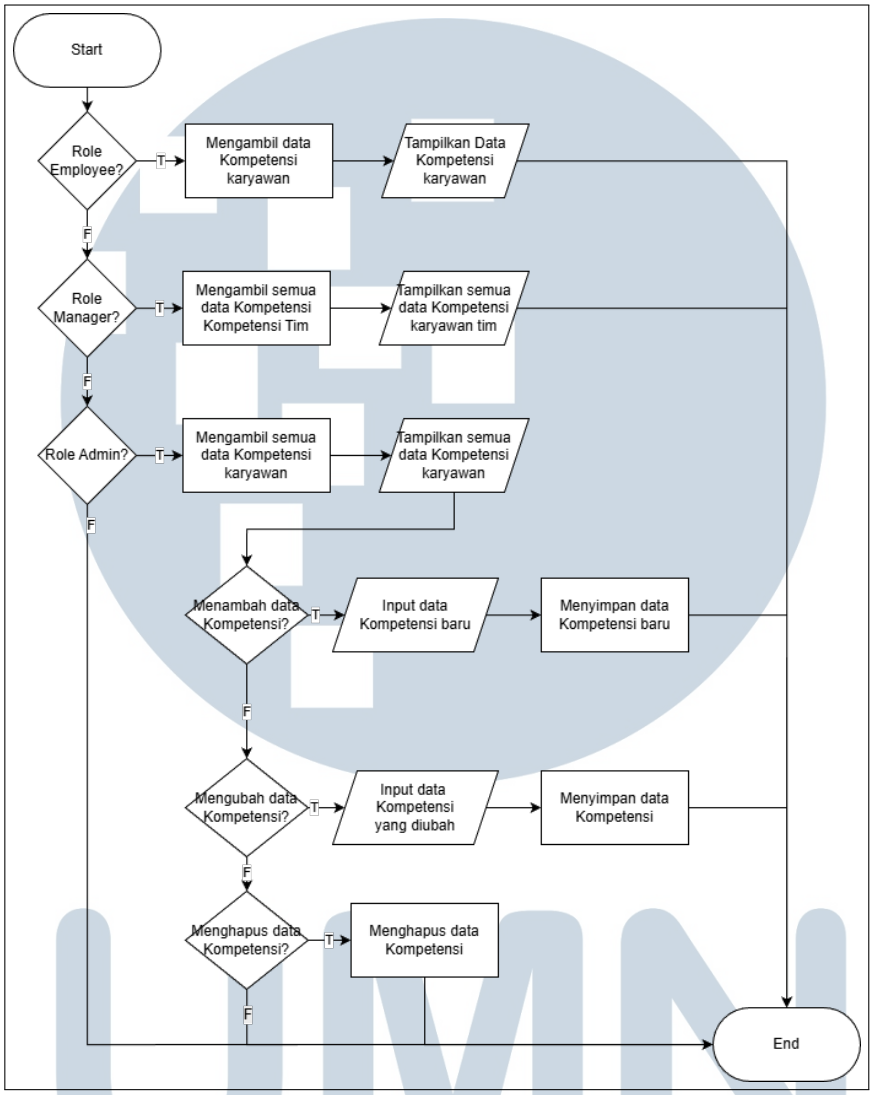

Figure 3.11. *Flowchart* Modul Kompetensi

Role employee dapat menarik data kompetensi sesuai jabatannya. *Role manager* diberikan akses terbatas pada lingkup bawahannya dengan validasi posisi yang ketat. *Role admin* memegang kendali penuh dalam pengelolaan data master kompetensi. Contoh pemanggilan API terlihat pada Gambar 3.12.

Algorithm 4 Alur Fitur Kompetensi

```
procedure KELOLAKOMPETENSI(role)
    Start
    if role = Employee then
        Mengambil kompetensi sendiri
    else if role = Manager then
        Mengambil kompetensi tim
    else if role = Admin then
        Mengambil semua kompetensi
    end if
    if role = Admin and manipulasi data then
        Simpan perubahan master data
    end if
    End
end procedure
```

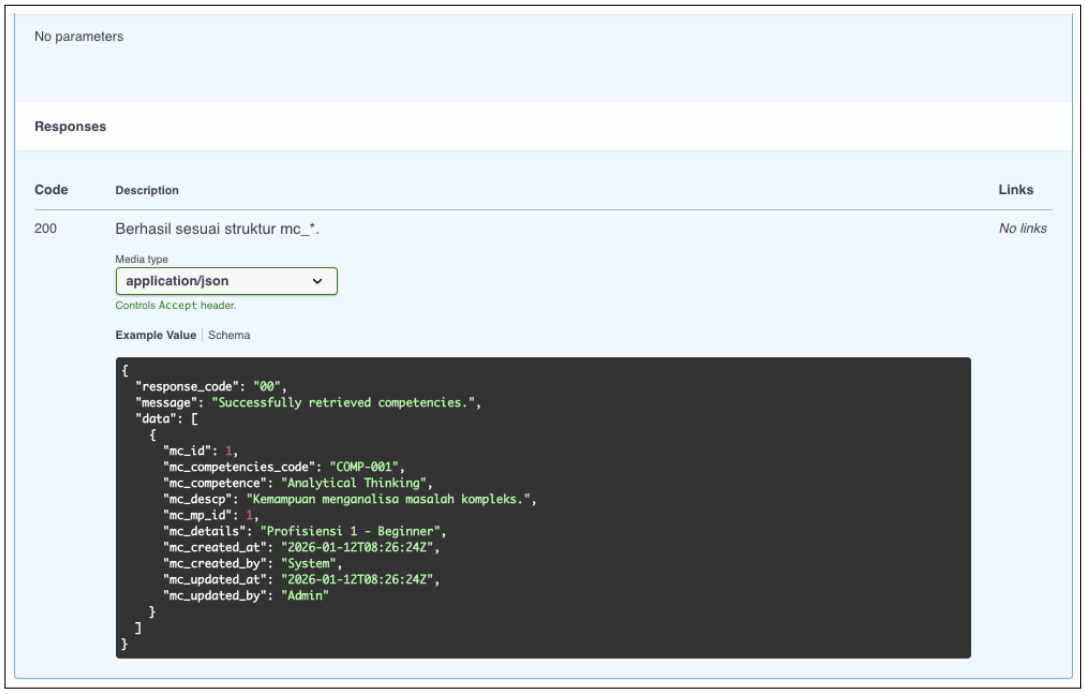

Figure 3.12. Contoh *Endpoint GET Competencies* oleh *Employee*

3.3.6.4 Rangkaian Keahlian (*Skillset*)

Modul ini berfungsi sebagai manajemen keahlian mandiri. Alur kerja modul ditunjukkan pada Gambar 3.13 dan diatur melalui logika pada Algoritma 5.

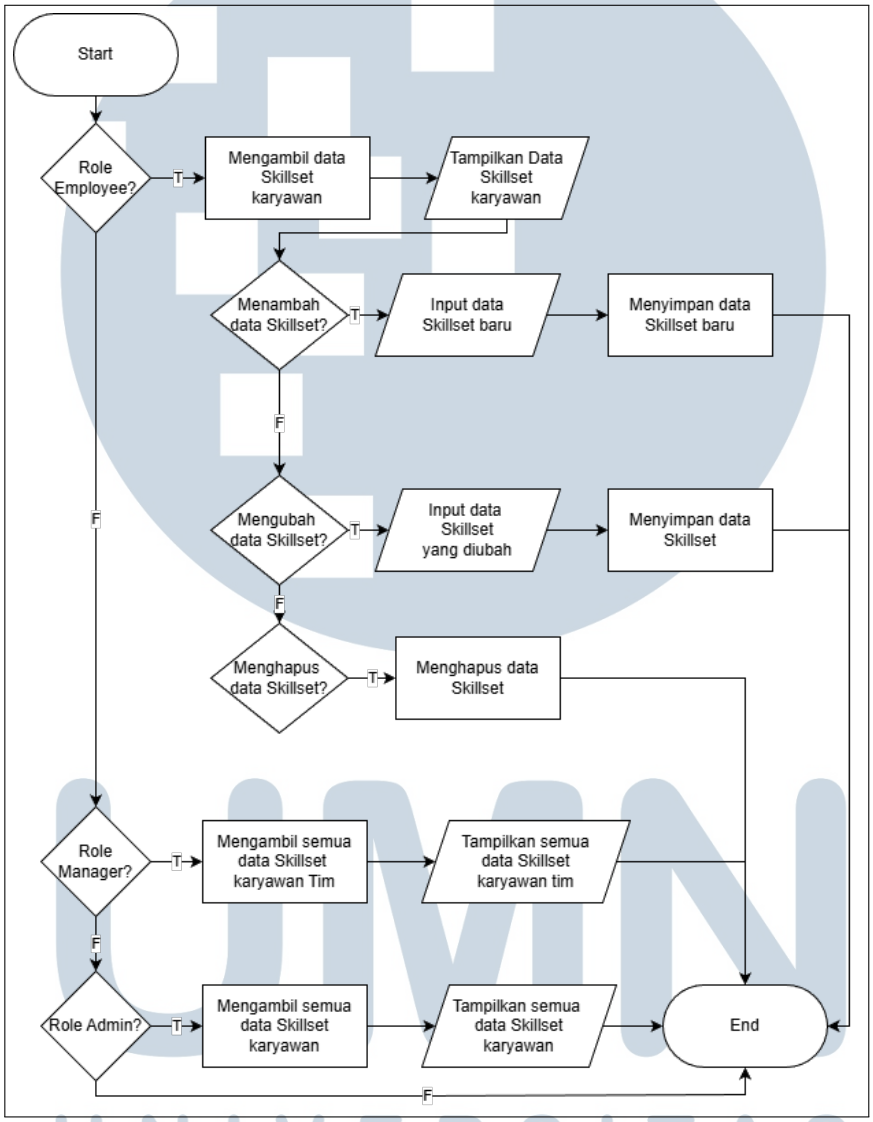

Figure 3.13. *Flowchart* Modul *Skillset*

Pada fitur ini, *role employee* dapat melakukan operasional *CRUD* terhadap keahlian pribadi. Untuk *role manager* dan *admin*, fungsionalitas dibatasi hanya pada pengambilan data guna peninjauan kompetensi tim. Contoh pemanggilan API ditunjukkan pada Gambar 3.14.

Algorithm 5 Alur Fitur Skillset

```
procedure KELOLASKILLSET(role)
    Start
    if role = Employee then
        Mengambil data;
        if manipulasi then
            Simpan data skillset
        end if
    else
        Mengambil data (Read-only)
    end if
    End
end procedure
```

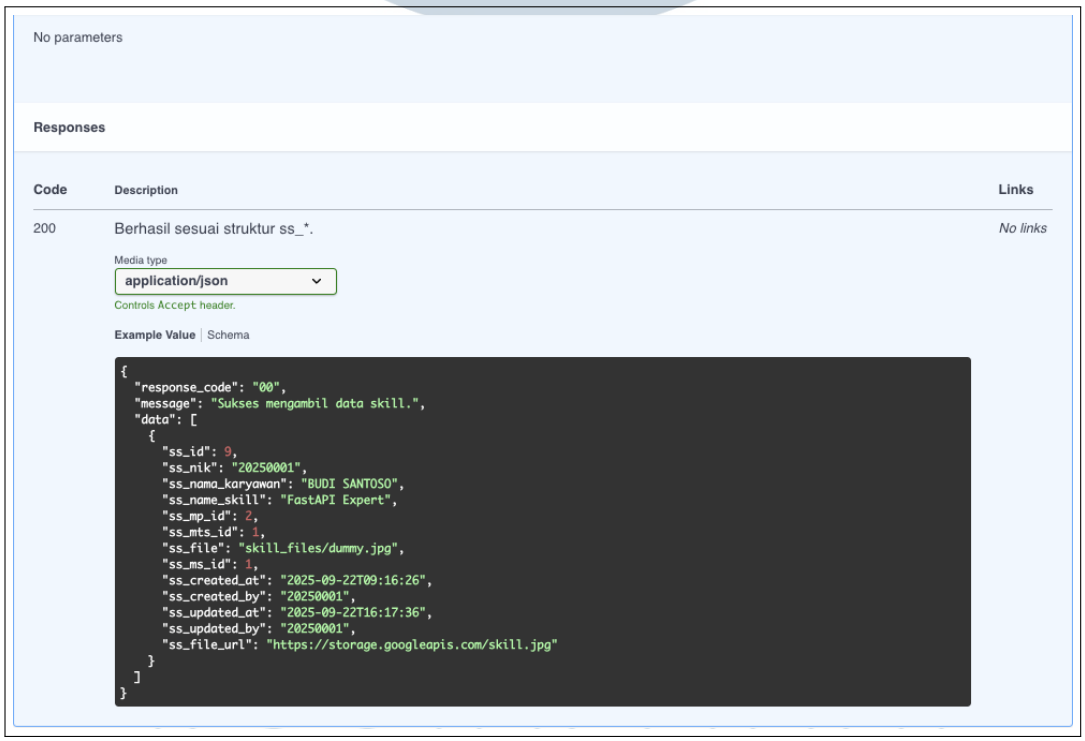

Figure 3.14. Contoh *Endpoint GET Skillset* oleh *Employee*

3.3.6.5 Riwayat Kerja (*Work History*)

Mekanisme ini mencatat rekam jejak profesional karyawan. *Flowchart* modul diperlihatkan pada Gambar 3.15 dan proses verifikasinya didefinisikan dalam Algoritma 6.

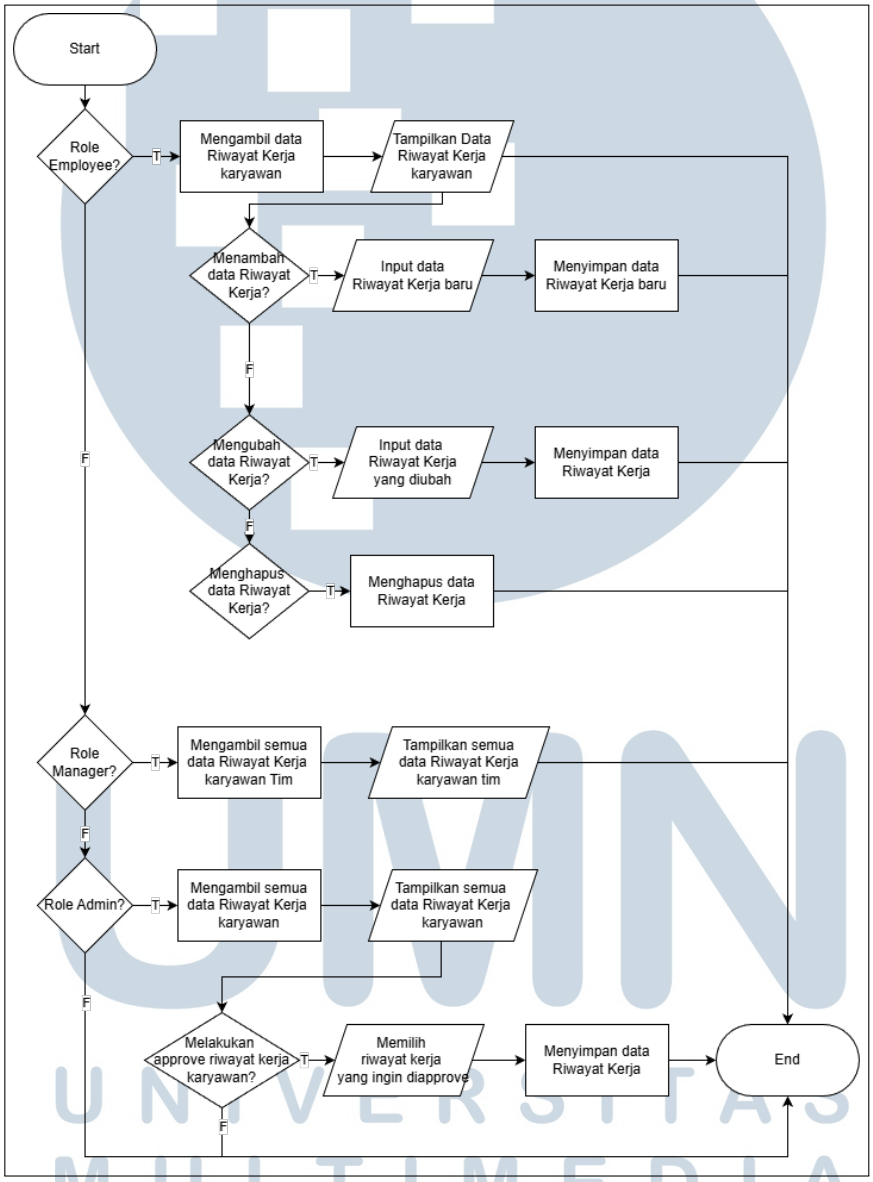

Figure 3.15. *Flowchart* Modul Riwayat Kerja

Role employee melakukan pengelolaan secara mandiri. *Role manager* memiliki hak akses menampilkan data bawahan sebagai referensi manajerial. *Role admin* bertanggung jawab sebagai verifikator untuk memberikan persetujuan terhadap data yang diinput karyawan. Contoh pemanggilan API ditunjukkan pada

Algorithm 6 Alur Fitur Riwayat Kerja

```
1:  procedure KELOLARIWAYATKERJA(role)
2:      Start
3:      if role = Employee then
4:          Kelola data (CRUD)
5:      else
6:          Tampilkan data tim/semua
7:      end if
8:      if role = Admin and pilih approve then
9:          Simpan status verifikasi
10:     end if
11:     End
12: end procedure
```

Gambar 3.16.

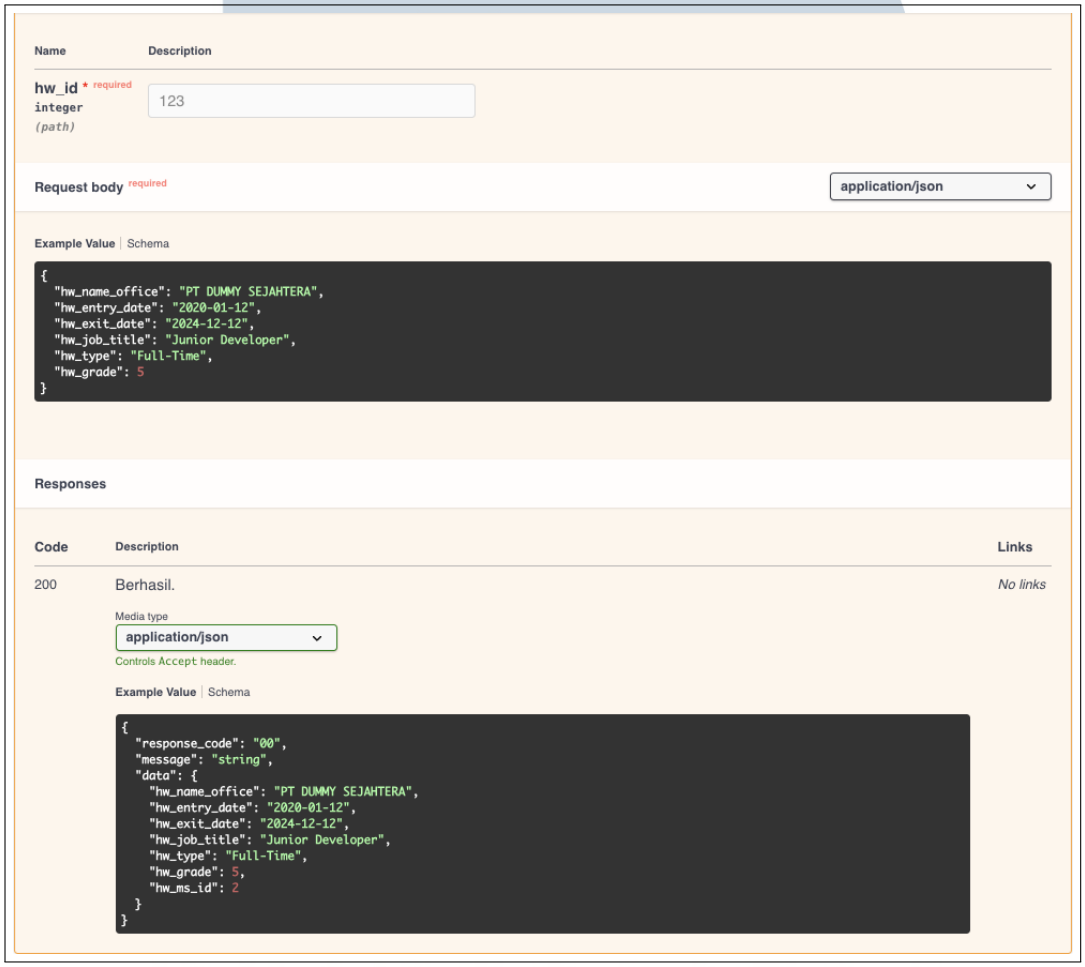

Figure 3.16. Contoh *Endpoint PUT Approval Work History* oleh *Admin*

3.3.6.6 Penghargaan (*Recognition*)

Modul ini mendokumentasikan apresiasi perusahaan. Alur kerjanya divisualisasikan pada Gambar 3.17 dan pengelolaannya dijelaskan dalam Algoritma 7.

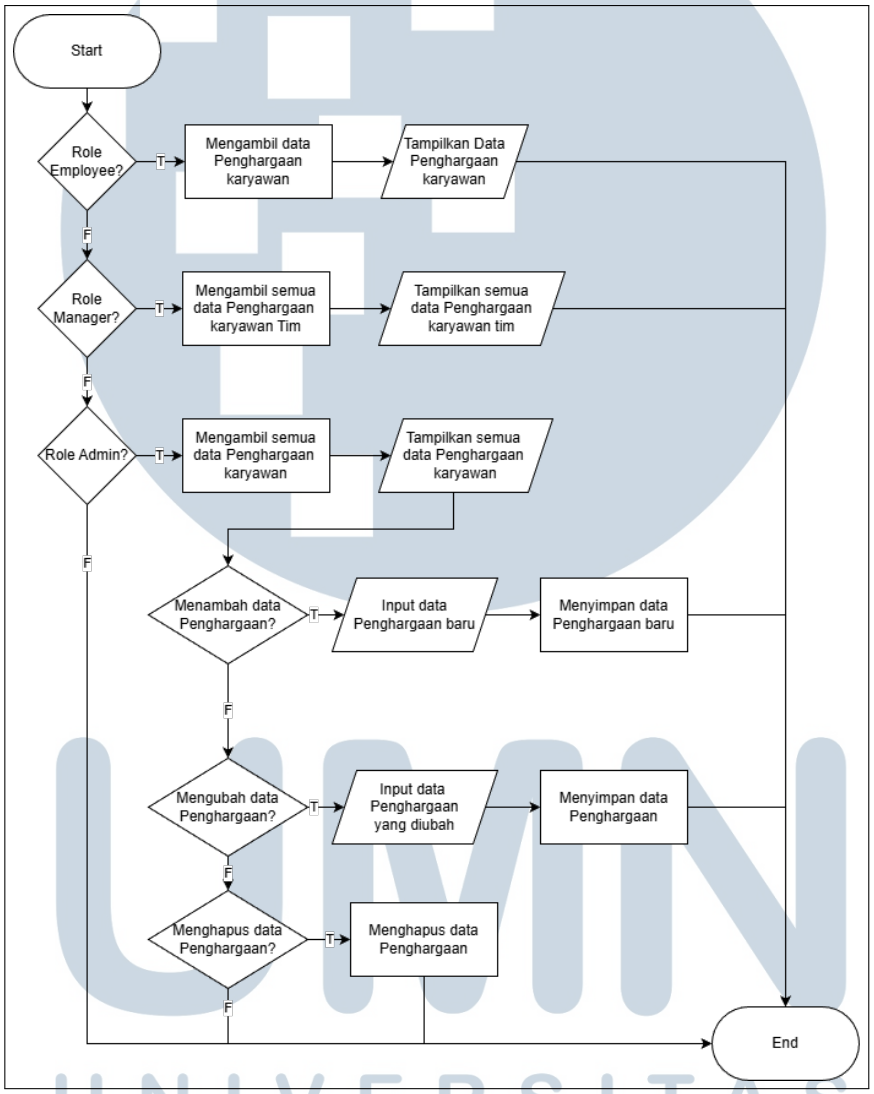

Figure 3.17. *Flowchart* Modul Penghargaan

Role employee hanya dapat melihat penghargaan mereka. *Role manager* dapat melihat daftar tersebut untuk evaluasi performa tim. Otoritas operasional *CRUD* berada di tangan *role admin* untuk menjaga validitas penghargaan. Contoh pemanggilan API ditunjukkan pada Gambar 3.18.

Algorithm 7 Alur Fitur Penghargaan

```
1:  procedure KELOLAPENGHARGAAN(role)
2:      Start
3:      if role = Employee then
4:          Tampilkan data pribadi
5:      else if role = Manager then
6:          Tampilkan data tim
7:      else if role = Admin then
8:          Kelola data (CRUD)
9:      end if
10:     End
11: end procedure
```

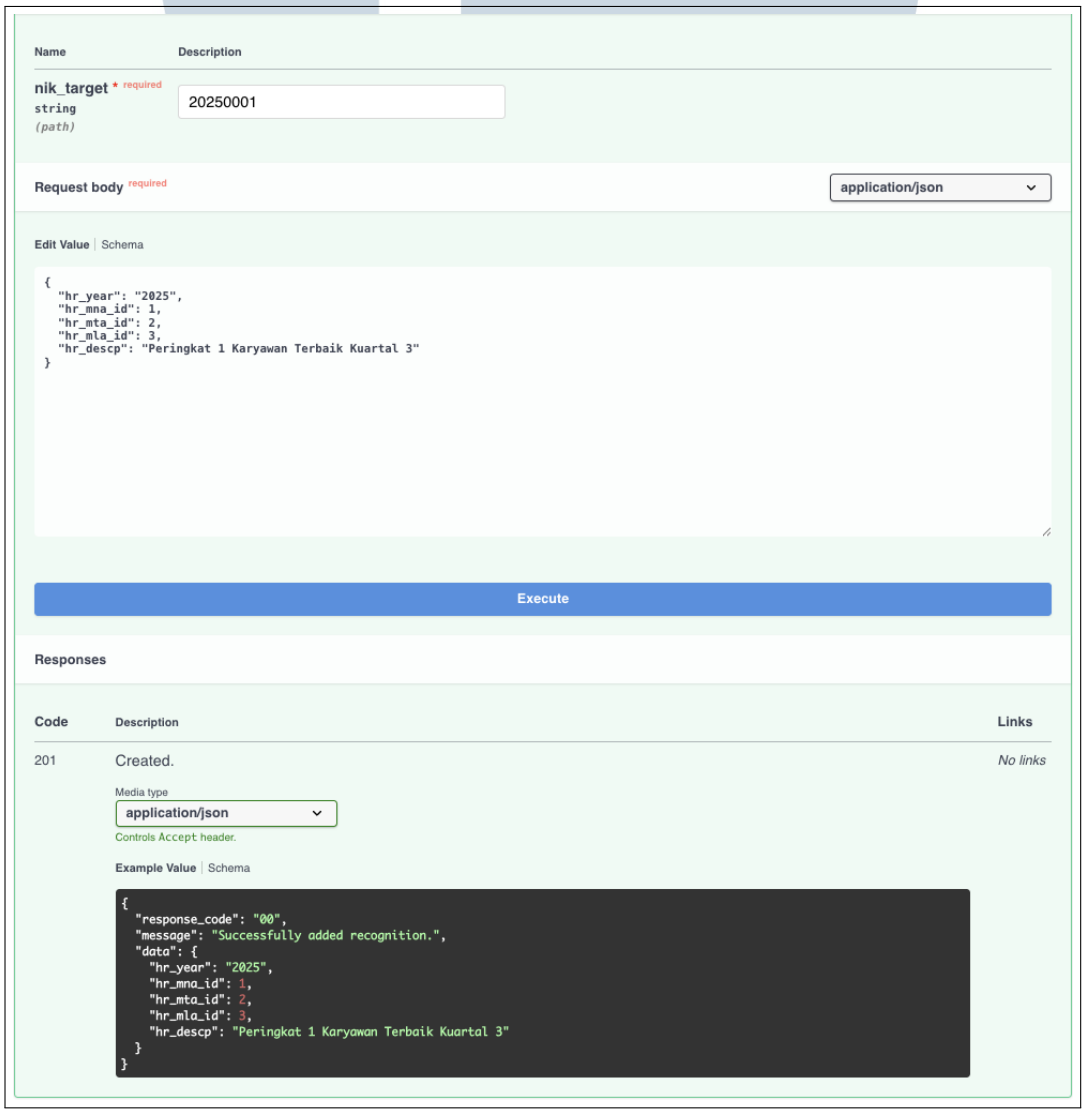

Figure 3.18. Contoh *Endpoint POST Recognition* oleh *Admin*

3.3.6.7 Riwayat Pelatihan (*Training History*)

Modul terakhir ini mencatat riwayat pengembangan diri. Alur kerjanya terdapat pada Gambar 3.19 dan fungsi persetujuannya dituangkan dalam Algoritma 8.

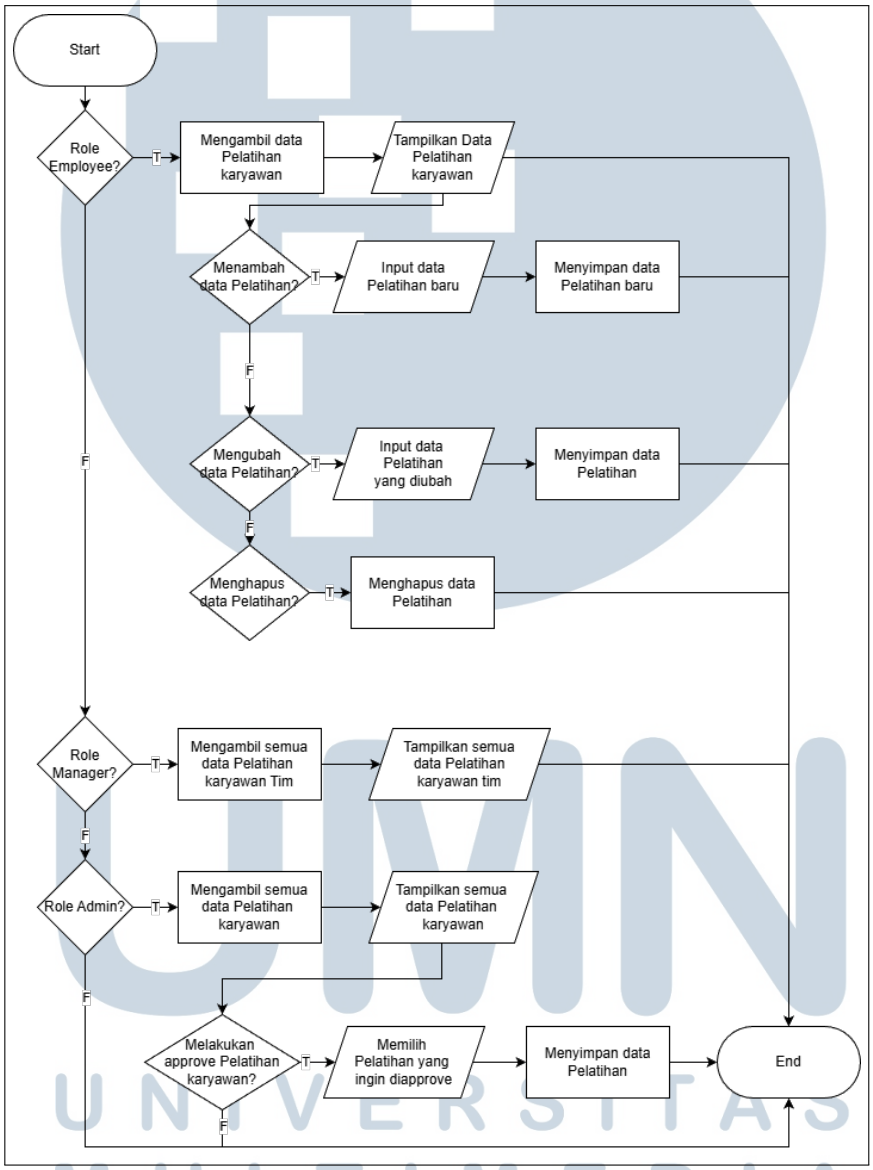

Figure 3.19. *Flowchart* Modul Riwayat Pelatihan

Role employee berhak mengelola data pelatihan yang diambil secara mandiri. *Role manager* memantau riwayat bawahan. *Role admin* berfungsi memberikan persetujuan terhadap sertifikasi yang diajukan. Contoh pemanggilan API ditunjukkan pada Gambar 3.20.

Algorithm 8 Alur Fitur Pelatihan

```
procedure KELOLAPELATIHAN(role)
    Start
    if role = Employee then
        Kelola (CRUD) data pelatihan
    else
        Tampilkan data (Read)
    end if
    if role = Admin and pilih approve then
        Simpan status verifikasi
    end if
    End
end procedure
```

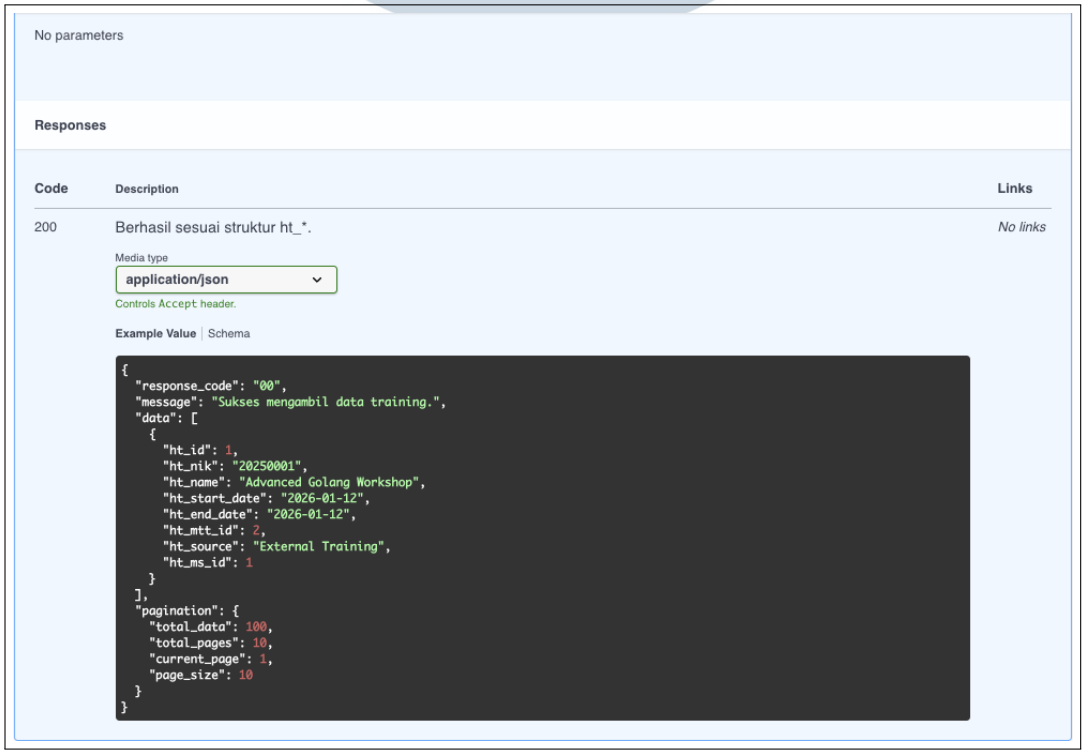

Figure 3.20. Contoh *Endpoint GET Training* oleh *Employee*

3.4 Kendala dan Solusi yang Ditemukan

3.4.1 Kendala

Dalam pengembangan aplikasi *Talent Hive*, ditemukan beberapa kendala yang memengaruhi linimasa proyek:

1. Dinamika Perubahan Kebutuhan: Perubahan spesifikasi dari pemangku kepentingan mengharuskan penyesuaian ulang logika bisnis.
2. Tingginya Volume Permintaan Fitur Baru: Permintaan tambahan mengakibatkan cakupan proyek meluas dari perencanaan awal.
3. Kompleksitas Refaktorisasi: Perubahan fitur fundamental menuntut modifikasi struktur basis data guna menjaga stabilitas sistem.
4. Keterbatasan Waktu: Perbedaan antara durasi magang dan volume permintaan fitur menyebabkan prioritas difokuskan pada fungsionalitas inti (*core features*).

3.4.2 Solusi

Guna mengatasi kendala tersebut, diambil langkah strategis melalui koordinasi intensif dengan tim proyek:

1. Koordinasi Prioritas: Dilakukan pemetaan urgensi permintaan fitur baru ke dalam kategori *Must-Have*.
2. Analisis Dampak Teknis: Kajian teknis dilakukan sebelum perubahan kode untuk meminimalisasi risiko kerusakan fitur stabil.
3. Sinkronisasi QA: Perubahan spesifikasi segera dikomunikasikan agar skenario pengujian tetap akurat.
4. Strategi MVP: Fokus dialihkan pada penyelesaian fitur krusial, sementara fitur tambahan didokumentasikan untuk rencana V2.