

**PENGEMBANGAN BACKEND SISTEM INFORMASI  
KEPEGAWAIAN CHRIS PADA PT GANDA VISI  
JAYATAMA**



**LAPORAN MBKM MAGANG**

**JOSE ANDREAS LIE  
00000067097**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK DAN INFORMATIKA  
UNIVERSITAS MULTIMEDIA NUSANTARA  
TANGERANG  
2025**

**PENGEMBANGAN BACKEND SISTEM INFORMASI  
KEPEGAWAIAN CHRIS PADA PT GANDA VISI  
JAYATAMA**



LAPORAN MBKM MAGANG

**JOSE ANDREAS LIE  
00000067097**

**UMN**  
**UNIVERSITAS**  
**MULTIMEDIA**  
**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNIK DAN INFORMATIKA**  
**UNIVERSITAS MULTIMEDIA NUSANTARA**  
**TANGERANG**  
**2025**

## HALAMAN PERNYATAAN ORISINALITAS TIDAK PLAGIAT

Dengan ini saya,

Nama : Jose Andreas Lie  
NIM : 00000067097  
Program Studi : Informatika

Menyatakan dengan sesungguhnya bahwa Laporan MBKM Magang saya yang berjudul:

**Pengembangan Backend Sistem Informasi Kepegawaian CHRIS pada PT Ganda Visi Jayatama**

merupakan hasil karya saya sendiri, bukan merupakan hasil plagiat, dan tidak pula dituliskan oleh orang lain; Semua sumber, baik yang dikutip maupun dirujuk, telah saya cantumkan dan nyatakan dengan benar pada bagian Daftar Pustaka.

Jika di kemudian hari terbukti ditemukan kecurangan/penyimpangan, baik dalam pelaksanaan skripsi maupun dalam penulisan laporan karya ilmiah, saya bersedia menerima konsekuensi untuk dinyatakan TIDAK LULUS. Saya juga bersedia menanggung segala konsekuensi hukum yang berkaitan dengan tindak plagiarisme ini sebagai kesalahan saya pribadi dan bukan tanggung jawab Universitas Multimedia Nusantara.

Tangerang, Tgl. Pengumpulan

[meterai Rp10.000,-]

(Jose Andreas Lie)

## **HALAMAN PERSETUJUAN**

Laporan MBKM Magang dengan judul

### **PENGEMBANGAN BACKEND SISTEM INFORMASI KEPEGAWAIAN CHRIS PADA PT GANDA VISI JAYATAMA**

oleh

Nama : Jose Andreas Lie  
NIM : 00000067097  
Program Studi : Informatika  
Fakultas : Fakultas Teknik dan Informatika

Telah disetujui untuk diajukan pada

Sidang Ujian Laporan MBKM Magang Universitas Multimedia Nusantara

Tangerang, Tgl. Pengumpulan

Pembimbing

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA  
(Dr. Ivransa Zuhdi Pane, B.Eng., M.Eng.)  
NIDN: 0012345600  
Ketua Program Studi Informatika,

(Arya Wicaksana, S.Kom., M.Eng.Sc., OCA)  
NIDN: 0315109103

## HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH MAHASISWA

Yang bertanda tangan di bawah ini:

Nama : Jose Andreas Lie  
NIM : 00000067097  
Program Studi : Informatika  
Jenjang : S1  
Jenis Karya : Laporan MBKM Magang

Menyatakan dengan sesungguhnya bahwa:

- Saya bersedia memberikan izin sepenuhnya kepada Universitas Multimedia Nusantara untuk mempublikasikan hasil karya ilmiah saya di repositori Knowledge Center, sehingga dapat diakses oleh Civitas Akademika/Publik. Saya menyatakan bahwa karya ilmiah yang saya buat tidak mengandung data yang bersifat konfidensial dan saya juga tidak akan mencabut kembali izin yang telah saya berikan dengan alasan apapun.
- Saya tidak bersedia karena dalam proses pengajuan untuk diterbitkan ke jurnal/konferensi nasional/internasional (dibuktikan dengan *letter of acceptance*)\*\*.

Tangerang, Tgl. Pengumpulan

Yang menyatakan

Jose Andreas Lie

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

\*\* Jika tidak bisa membuktikan LoA jurnal/HKI selama enam bulan ke depan, saya bersedia mengizinkan penuh karya ilmiah saya untuk diunggah ke KC UMN dan menjadi hak institusi UMN.

## **Halaman Persembahan / Motto**

”A good name is to be more desired than great wealth, Favor is better than silver and gold.”

Proverbs 22:1 (NASB)



## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan kerja praktik ini dengan baik. Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan Program Studi S1 Informatika di Universitas Multimedia Nusantara.

Mengucapkan terima kasih

1. Bapak Dr. Ir. Andrey Andoko, M.Sc., selaku Rektor Universitas Multimedia Nusantara.
2. Bapak Dr. Eng. Niki Prastomo, S.T., M.Sc., selaku Dekan Fakultas Teknik dan Informatika Universitas Multimedia Nusantara.
3. Bapak Arya Wicaksana, S.Kom., M.Eng.Sc., OCA, selaku Ketua Program Studi Informatika Universitas Multimedia Nusantara.
4. Bapak Dr. Ivransa Zuhdi Pane, B.Eng., M.Eng., sebagai Pembimbing pertama yang telah banyak meluangkan waktu untuk memberikan bimbingan, arahan dan motivasi atas terselesainya tesis ini.
5. Kepada Bapak Edo Setiawan, selaku Head of Development, yang telah memberikan kesempatan kepada penulis untuk melaksanakan kerja praktik di perusahaan ini.
6. Keluarga dan teman-teman saya yang telah memberikan bantuan dukungan material dan moral, sehingga penulis dapat menyelesaikan tesis ini. (kalau ada).

Semoga laporan ini dapat bermanfaat bagi pembaca, khususnya bagi penulis sendiri. Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh karena itu penulis mengharapkan kritik dan saran yang membangun untuk perbaikan di masa yang akan datang.

Tangerang, Tgl. Pengumpulan

Jose Andreas Lie

# PENGEMBANGAN BACKEND SISTEM INFORMASI KEPEGAWAIAN CHRIS PADA PT GANDA VISI JAYATAMA

Jose Andreas Lie

## ABSTRAK

Laporan ini membahas proses pengembangan lanjutan *backend* dari sistem informasi kepegawaian CHRIS (Concise Human Resource Information System) yang digunakan di PT Ganda Visi Jayatama. Fokus utama pengembangan mencakup penambahan modul baru dan penyempurnaan modul yang telah ada. Beberapa fitur utama yang dikembangkan meliputi: integrasi otentikasi biometrik pada aplikasi *mobile* CHRISM, penerapan sistem hierarki berbasis pohon pada modul cuti, serta perancangan dan implementasi modul *Payroll* yang baru. Selain itu, perbaikan juga dilakukan pada modul *User Management* dengan peningkatan validasi form input.

Pengembangan dilakukan menggunakan teknologi PERN Stack (*PostgreSQL*, *Express.js*, *React.js*, dan *Node.js*) dengan pendekatan *Agile* sebagai metodologi kerja tim. Selama pelaksanaan proyek, telah dibuat 17 *endpoint API* baru dan pengembangan terhadap 7 *endpoint* eksisting, sehingga total terdapat 24 *endpoint* yang mendukung berbagai fitur sistem. Seluruh pengujian dilakukan secara manual. Dampak dari pengembangan ini antara lain peningkatan efisiensi dan akurasi dalam proses penggajian melalui modul *Payroll*, peningkatan keamanan akses melalui autentikasi biometrik, serta sistem persetujuan cuti yang lebih fleksibel melalui implementasi struktur hierarki dinamis. Adapun kendala yang dihadapi mencakup kode yang tidak sesuai *best practice*, struktur data yang belum optimal, dan kompleksitas dalam pengolahan data historis. Seluruh kendala tersebut diatasi melalui *refactoring code*, penambahan struktur data baru, dan optimalisasi proses migrasi.

**Kata kunci:** Backend development, Biometrik, Payroll, Sistem informasi kepegawaian, Struktur hierarki.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

**BACKEND ENHANCEMENT OF THE CHRIS HR INFORMATION SYSTEM  
AT PT GANDA VISI JAYATAMA**

Jose Andreas Lie

**ABSTRACT**

This report presents the backend development and enhancement of the CHRIS (Concise Human Resource Information System) used by PT Ganda Visi Jayatama. The development focused on extending existing modules and integrating new features, such as biometric authentication for CHRISM (CHRIS Mobile), a tree-based hierarchy system for the leave permit module, and the implementation of a fully new Payroll module. Additionally, improvements were made to the User Management module by strengthening input validation.

The system was developed using the PERN Stack (PostgreSQL, Express.js, React.js, and Node.js) under an Agile development methodology. In total, 17 new API endpoints were developed, along with the enhancement of 7 existing ones, resulting in 24 functional endpoints across the system. All testing was conducted manually.

The improvements provided significant impact in operational efficiency—automated salary calculation through the Payroll module, increased security via biometric authentication, and a more dynamic leave approval process through hierarchical supervision. Challenges encountered during the project included codebase that did not follow best practices, hardcoded values, and complex historical data handling. These were resolved through refactoring, additional transaction tables, and optimization of data migration processes.

**Keywords:** *Backend development, Biometric, Payroll, Human resource information system, Hierarchical structure.*

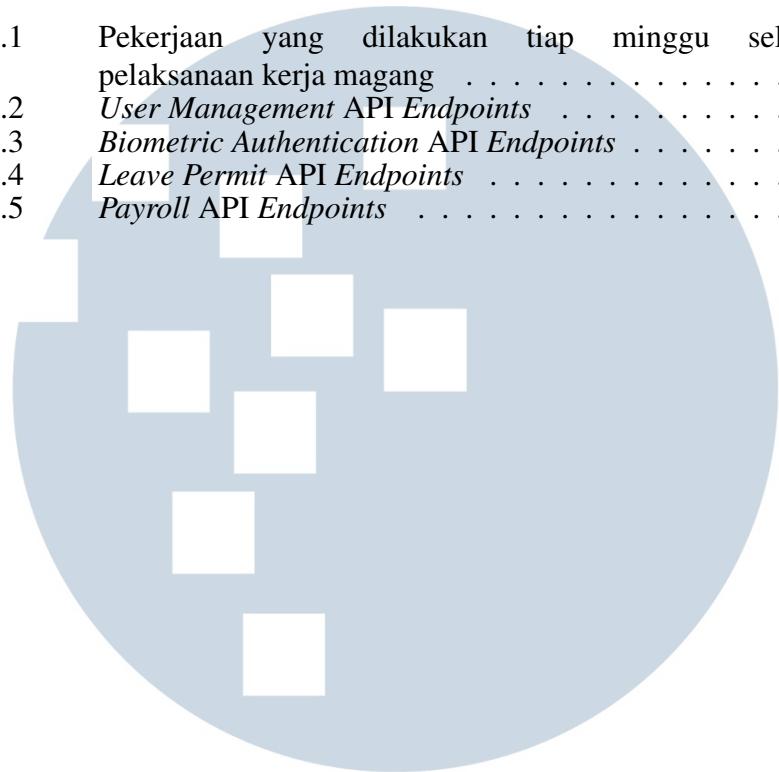
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## DAFTAR ISI

HALAMAN JUDUL . . . . .	i
HALAMAN PERNYATAAN ORISINALITAS . . . . .	ii
HALAMAN PERSETUJUAN . . . . .	iii
HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH . . . . .	iv
HALAMAN PERSEMBAHAN/MOTO . . . . .	v
KATA PENGANTAR . . . . .	vi
ABSTRAK . . . . .	vii
ABSTRACT . . . . .	viii
DAFTAR ISI . . . . .	ix
DAFTAR TABEL . . . . .	x
DAFTAR GAMBAR . . . . .	xi
DAFTAR LAMPIRAN . . . . .	xiii
BAB 1 PENDAHULUAN . . . . .	1
1.1 Latar Belakang Masalah . . . . .	1
1.2 Maksud dan Tujuan Kerja Magang . . . . .	3
1.3 Waktu dan Prosedur Pelaksanaan Kerja Magang . . . . .	4
BAB 2 GAMBARAN UMUM PERUSAHAAN . . . . .	6
2.1 Sejarah Singkat Perusahaan . . . . .	6
2.2 Visi dan Misi Perusahaan . . . . .	6
2.3 Struktur Organisasi Perusahaan . . . . .	7
BAB 3 PELAKSANAAN KERJA MAGANG . . . . .	8
3.1 Kedudukan dan Koordinasi . . . . .	8
3.2 Tugas yang Dilakukan . . . . .	9
3.3 Uraian Pelaksanaan Magang . . . . .	9
3.4 Pengumpulan dan Analisis Kebutuhan . . . . .	11
3.5 Perancangan dan Pengembangan Sistem . . . . .	13
3.5.1 <i>User Management</i> dan Validasi Data . . . . .	13
3.5.2 Leave Permit . . . . .	16
3.5.3 Fitur Autentikasi Biometrik pada CHRIS Mobile . . . . .	21
3.5.4 Sistem Hierarki Supervisi . . . . .	25
3.5.5 Payroll . . . . .	27
3.6 Implementasi Sistem . . . . .	42
3.6.1 Implementasi <i>User Management</i> dan Validasi Data . . . . .	43
3.6.2 Implementasi Fitur Autentikasi Biometrik pada CHRIS Mobile . . . . .	50
3.6.3 Implementasi Leave Permit . . . . .	53
3.6.4 Implementasi Payroll . . . . .	59
3.7 Kendala dan Solusi yang Ditemukan . . . . .	77
3.7.1 Kendala . . . . .	77
3.7.2 Solusi . . . . .	77
BAB 4 SIMPULAN DAN SARAN . . . . .	79
4.1 Simpulan . . . . .	79
4.2 Saran . . . . .	79
DAFTAR PUSTAKA . . . . .	81

## DAFTAR TABEL

Tabel 3.1	Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang . . . . .	10
Tabel 3.2	<i>User Management API Endpoints</i> . . . . .	43
Tabel 3.3	<i>Biometric Authentication API Endpoints</i> . . . . .	50
Tabel 3.4	<i>Leave Permit API Endpoints</i> . . . . .	53
Tabel 3.5	<i>Payroll API Endpoints</i> . . . . .	60



**UMN**  
**UNIVERSITAS**  
**MULTIMEDIA**  
**NUSANTARA**

## DAFTAR GAMBAR

Gambar 2.1	Logo perusahaan PT Ganda Visi Jayatama . . . . .	6
Gambar 2.2	Struktur organisasi perusahaan PT Visi Ganda Jayatama . . . . .	7
Gambar 3.1	Struktur Tim CHRIS . . . . .	8
Gambar 3.2	Diagram ERD untuk modul User Management . . . . .	14
Gambar 3.3	Diagram ERD untuk modul Leave Permit . . . . .	17
Gambar 3.4	Flowchart alur sistem Leave Permit . . . . .	18
Gambar 3.5	Flowchart alur sistem Cancel Leave Permit . . . . .	20
Gambar 3.6	Flowchart alur sistem registrasi biometrik pada CHRIS Mobile . . . . .	22
Gambar 3.7	Flowchart alur sistem autentikasi biometrik pada CHRIS Mobile . . . . .	24
Gambar 3.8	Logika Fungsi Hierarki untuk Mencari Semua Bawahan . . . . .	26
Gambar 3.9	Diagram ERD untuk modul Payroll . . . . .	27
Gambar 3.10	Flowchart <i>create payroll configuration</i> . . . . .	30
Gambar 3.11	Flowchart <i>update payroll configuration</i> . . . . .	31
Gambar 3.12	Flowchart <i>delete payroll configuration</i> . . . . .	32
Gambar 3.13	Flowchart implementasi payroll untuk setiap user . . . . .	34
Gambar 3.14	Flowchart menambahkan data untuk payroll information . . . . .	35
Gambar 3.15	Flowchart mendaftarkan data untuk salary . . . . .	36
Gambar 3.16	Flowchart mendaftarkan data user allowances secara <i>general</i> . . . . .	37
Gambar 3.17	Flowchart mendaftarkan data <i>user payroll application</i> . . . . .	38
Gambar 3.18	Flowchart mendaftarkan data <i>user allowances</i> . . . . .	39
Gambar 3.19	Flowchart mendapatkan salary slip dari setiap user . . . . .	41
Gambar 3.20	Contoh struktur <i>request</i> untuk /auth/register . . . . .	45
Gambar 3.21	Contoh struktur <i>response</i> untuk /auth/register . . . . .	46
Gambar 3.22	Contoh struktur <i>parameters</i> untuk /users/update . . . . .	46
Gambar 3.23	Contoh struktur <i>request</i> untuk /users/update . . . . .	47
Gambar 3.24	Contoh struktur <i>response</i> untuk /users/update . . . . .	47
Gambar 3.25	Contoh struktur <i>request</i> untuk /employee-statuses . . . . .	48
Gambar 3.26	Contoh struktur <i>request</i> untuk /banks . . . . .	49
Gambar 3.27	Contoh struktur <i>request</i> untuk /auth/biometric/register . . . . .	50
Gambar 3.28	Contoh struktur <i>response</i> untuk /auth/biometric/register . . . . .	51
Gambar 3.29	Contoh struktur <i>request</i> untuk /auth/biometric/login . . . . .	51
Gambar 3.30	Contoh struktur <i>response</i> untuk /auth/biometric/login . . . . .	52
Gambar 3.31	Contoh struktur <i>response</i> untuk GET /leave-permit . . . . .	54
Gambar 3.32	Contoh struktur <i>request</i> untuk GET /leave-permit/:id . . . . .	55
Gambar 3.33	Contoh struktur <i>response</i> untuk GET /leave-permit/:id . . . . .	55
Gambar 3.34	Contoh struktur <i>request</i> untuk POST /leave-permit . . . . .	56
Gambar 3.35	Contoh struktur <i>response</i> untuk POST /leave-permit . . . . .	56
Gambar 3.36	Contoh struktur <i>request</i> untuk DELETE /leave-permit . . . . .	57
Gambar 3.37	Contoh struktur <i>response</i> untuk DELETE /leave-permit/:id . . . . .	57
Gambar 3.38	Contoh struktur <i>response</i> untuk GET /leave-permit/dashboard . . . . .	58
Gambar 3.39	Contoh struktur <i>response</i> untuk GET /leave-permit/all . . . . .	59

Gambar 3.40	Contoh struktur <i>request</i> untuk GET /payroll-configuration/all . . . . .	62
Gambar 3.41	Contoh struktur <i>request</i> untuk GET /payroll-configuration/:id . . . . .	63
Gambar 3.42	Contoh struktur <i>response</i> untuk GET /payroll-configuration/:id . . . . .	63
Gambar 3.43	Contoh struktur <i>request</i> untuk POST /payroll-configuration . . . . .	64
Gambar 3.44	Contoh struktur <i>response</i> untuk POST /payroll-configuration . . . . .	64
Gambar 3.45	Contoh struktur <i>request</i> untuk DELETE /payroll-configuration/:id . . . . .	64
Gambar 3.46	Contoh struktur <i>response</i> untuk DELETE /payroll-configuration/:id . . . . .	65
Gambar 3.47	Contoh struktur <i>params</i> untuk PATCH /payroll-configuration/:id . . . . .	65
Gambar 3.48	Contoh struktur <i>request</i> untuk PATCH /payroll-configuration/:id . . . . .	65
Gambar 3.49	Contoh struktur <i>response</i> untuk PATCH /payroll-configuration/:id . . . . .	66
Gambar 3.50	Contoh struktur <i>request</i> untuk GET /users/:user_id/payroll-configuration . . . . .	66
Gambar 3.51	Contoh struktur <i>response</i> untuk GET /users/:user_id/payroll-configuration . . . . .	67
Gambar 3.52	Contoh struktur <i>request</i> untuk GET /users/:user_id/salary . . . . .	67
Gambar 3.53	Contoh struktur <i>response</i> untuk GET /users/:user_id/salary . . . . .	67
Gambar 3.54	Contoh struktur <i>request</i> untuk GET /user-allowance/:user_id/:payroll_configuration_id . . . . .	68
Gambar 3.55	Contoh struktur <i>response</i> untuk GET /user-allowance/:user_id/:payroll_configuration_id . . . . .	68
Gambar 3.56	Contoh struktur <i>params</i> untuk PATCH /users/:user_id/with-details . . . . .	69
Gambar 3.57	Contoh struktur <i>request</i> untuk PATCH /users/:user_id/with-details . . . . .	70
Gambar 3.58	Contoh struktur <i>response</i> untuk PATCH /users/:user_id/with-details . . . . .	71
Gambar 3.59	Contoh struktur <i>request</i> untuk GET /salary-slip/all . . . . .	71
Gambar 3.60	Contoh struktur <i>response</i> untuk GET /salary-slip/all . . . . .	72
Gambar 3.61	Contoh struktur <i>request</i> untuk GET /salary-slip/:user_id . . . . .	73
Gambar 3.62	Contoh struktur <i>response</i> untuk GET /salary-slip/:user_id . . . . .	74
Gambar 3.63	Contoh struktur <i>request</i> untuk GET /salary-slip?date=YYYY-MM . . . . .	75
Gambar 3.64	Contoh struktur <i>response</i> untuk GET /salary-slip?date=YYYY-MM . . . . .	76

## **DAFTAR LAMPIRAN**

Lampiran 1	MBKM-01 Cover Letter MBKM Internship Track 1 . . . . .	83
Lampiran 2	MBKM-02 MBKM Internship Track 1 Card . . . . .	84
Lampiran 3	MBKM-03 Daily Task - Internship Track 1 . . . . .	85
Lampiran 4	MBKM-04 Verification Form of Internship Report MBKM Internship Track 1 . . . . .	96
Lampiran 5	Form Bimbingan . . . . .	97
Lampiran 6	Transkrip Wawancara . . . . .	98



## BAB 1

### PENDAHULUAN

#### 1.1 Latar Belakang Masalah

Transformasi digital dalam pengelolaan sumber daya manusia (SDM) telah menjadi kebutuhan penting di era industri 4.0. Penerapan sistem informasi kepegawaian atau Human Resource Information System (HRIS) memungkinkan organisasi untuk mengotomatisasi proses administratif, meningkatkan akurasi data, dan mempercepat pengambilan keputusan strategis [1]. Dalam konteks ini, PT Ganda Visi Jayatama, sebuah perusahaan yang bergerak di bidang teknologi informasi, mengembangkan sebuah sistem HRIS internal bernama CHRIS (Concise Human Resource Information System) sebagai solusi terintegrasi untuk mendukung pengelolaan pegawai.

Sistem CHRIS dirancang sebagai platform yang menangani data kepegawaian, proses absensi, hingga pengajuan cuti. Seiring berjalannya waktu dan meningkatnya kompleksitas kebutuhan perusahaan, sistem ini memerlukan pengembangan lebih lanjut untuk memastikan performa dan skalabilitasnya tetap optimal [2]. Sistem sebelumnya dinilai belum mampu mengakomodasi seluruh proses operasional SDM secara menyeluruh, khususnya pada aspek *payroll*, manajemen struktur organisasi, dan otentikasi pengguna.

Salah satu permasalahan yang muncul adalah proses penggajian yang masih dilakukan secara manual atau menggunakan aplikasi terpisah, yang rentan menimbulkan kesalahan perhitungan dan redundansi data. Menurut Ahmed dan Ali (2023), digitalisasi modul *payroll* dapat mengurangi kesalahan penghitungan gaji hingga 85% dan mempercepat proses pembayaran [3]. Oleh karena itu, integrasi modul *Payroll* yang mampu menghitung gaji secara otomatis berdasarkan data tunjangan dan absensi menjadi sangat krusial.

Selain itu, sistem pengajuan cuti atau Leave Permit yang digunakan sebelumnya memiliki keterbatasan dalam mendukung alur persetujuan yang merefleksikan struktur organisasi perusahaan. Proses perizinan masih bersifat statis, hanya mengizinkan atasan tertentu untuk menyetujui permohonan cuti, tanpa mempertimbangkan dinamika hubungan struktural antar pegawai. Kelemahan ini berdampak pada kurangnya fleksibilitas serta potensi keterlambatan dalam proses persetujuan. Penyesuaian terhadap logika hierarchy-based approval menjadi

penting guna memperkuat kontrol, meningkatkan akuntabilitas, dan memastikan bahwa setiap permohonan ditinjau oleh pihak yang tepat sesuai dengan struktur jabatan.

Penelitian oleh Ujianto et al. (2024) menunjukkan bahwa ketiadaan sistem yang terkomputerisasi dalam proses cuti dan lembur menyebabkan karyawan tidak mengetahui status pengajuan yang telah dilakukan, dan pihak HRD mengalami kesulitan dalam mengontrol proses kepegawaian. Sistem yang masih mengandalkan formulir manual memperlambat proses dan meningkatkan risiko miskomunikasi antar bagian. Oleh karena itu, pengembangan HRIS dengan alur pengajuan cuti berbasis digital dan terstruktur secara hierarkis sangat diperlukan demi mendukung efisiensi dan efektivitas proses operasional perusahaan secara menyeluruh [4].

Aspek kenyamanan dan keamanan sistem juga menjadi fokus utama dalam pengembangan CHRIS, terutama dengan diperkenalkannya aplikasi mobile CHRISS. Untuk memperkuat keamanan dan kenyamanan akses, autentikasi biometrik berbasis sidik jari diimplementasikan. Teknologi ini dinilai lebih aman dan praktis dibandingkan metode tradisional berbasis kata sandi.

Masalah lain yang ditemukan adalah kurangnya normalisasi data pada tabel-tabel utama, seperti status kepegawaian dan data bank. Sebelumnya, data seperti status kepegawaian ditulis secara *hardcoded* dalam bentuk *string*, yang berpotensi menimbulkan inkonsistensi data. Dengan diterapkannya referensi tabel untuk *employment\_status* dan *bank*, sistem kini lebih fleksibel dan mendukung pengelolaan data yang lebih terstruktur.

Lebih lanjut, kebutuhan akan struktur organisasi yang dinamis mendorong implementasi sistem hierarki berbasis pohon atau tree hierarchy. Sistem ini digunakan untuk mendefinisikan hubungan antara pegawai dan atasan secara fleksibel, yang tidak hanya berguna dalam pengelolaan alur cuti, tetapi juga dapat diperluas ke fitur-fitur lain seperti evaluasi kinerja, distribusi tugas, dan pelaporan. Struktur organisasi semacam ini memungkinkan sistem untuk menelusuri hubungan antar-entitas dengan efisien dan menetapkan tanggung jawab berdasarkan posisi dalam hierarki. Menurut Evrendilek et al. (2017), struktur pohon dalam organisasi menciptakan tantangan dan peluang tersendiri dalam hal penugasan kerja karena setiap entitas yang diberi tugas akan mempengaruhi keseluruhan sub-struktur di bawahnya [5]. Pendekatan ini memberikan fondasi logis untuk pengembangan sistem yang mempertimbangkan keterhubungan antar pegawai secara hierarkis.

Pengembangan sistem CHRIS selama masa kerja praktik ini bertujuan untuk mengatasi berbagai permasalahan di atas dengan mengimplementasikan

modul-modul baru yang terintegrasi, memperkuat validasi input, serta menyusun ulang struktur data yang ada. Proyek ini merupakan kelanjutan dari sistem yang telah dikembangkan sebelumnya oleh tim internal PT Ganda Visi Jayatama. Pengembangan dilakukan secara iteratif menggunakan prinsip Agile dan dilakukan bersama dengan tim backend, frontend, dan desainer, untuk memastikan integrasi sistem yang solid dan relevan dengan kebutuhan pengguna akhir [6, 7].

## 1.2 Maksud dan Tujuan Kerja Magang

Pelaksanaan kerja magang di PT Ganda Visi Jayatama bertujuan untuk memberikan pengalaman langsung kepada mahasiswa dalam menghadapi permasalahan riil di industri, serta berkontribusi terhadap pengembangan sistem informasi yang sedang berjalan. Melalui kegiatan magang ini, peserta memperoleh pemahaman yang lebih mendalam mengenai praktik terbaik dalam pengembangan perangkat lunak, khususnya dalam konteks pengelolaan sistem kepegawaian berbasis web.

Secara khusus, maksud dari kerja magang ini adalah:

- Mengimplementasikan pengetahuan akademik yang telah diperoleh selama masa perkuliahan dalam lingkungan kerja profesional.
- Mengasah keterampilan teknis dan kolaboratif melalui kerja tim lintas divisi dalam proyek pengembangan perangkat lunak yang kompleks.
- Mengamati, mempelajari, dan memahami alur kerja profesional dalam pengembangan sistem backend yang terstruktur dan terdokumentasi.

Adapun tujuan utama dari pelaksanaan magang ini, yang difokuskan pada pengembangan sistem CHRIS, meliputi:

1. Mengembangkan dan menyempurnakan modul **Payroll**, agar proses penggajian dapat dilakukan secara otomatis, terstandarisasi, dan efisien.
2. Mengoptimalkan modul **Leave Permit** agar mendukung alur persetujuan berdasarkan struktur organisasi dan menyediakan fitur pembatalan cuti yang fleksibel.
3. Meningkatkan **keamanan akses sistem** melalui penerapan autentikasi biometrik pada aplikasi mobile CHRISM.

4. Menyusun dan menerapkan struktur **hierarki supervisi** berbasis pohon (*tree hierarchy*) untuk mendukung proses persetujuan yang lebih fleksibel.
5. Melakukan **refaktor dan validasi form input** pada modul **User Management** untuk meningkatkan akurasi dan integritas data pegawai.
6. Mengembangkan **RESTful API** yang mendukung integrasi antara frontend dan backend secara optimal.

Dengan pencapaian tujuan-tujuan tersebut, diharapkan sistem CHRIS dapat mendukung operasional perusahaan secara lebih efisien, aman, dan terukur, serta memberikan kontribusi nyata dalam peningkatan kualitas pengelolaan sumber daya manusia di PT Ganda Visi Jayatama.

### 1.3 Waktu dan Prosedur Pelaksanaan Kerja Magang

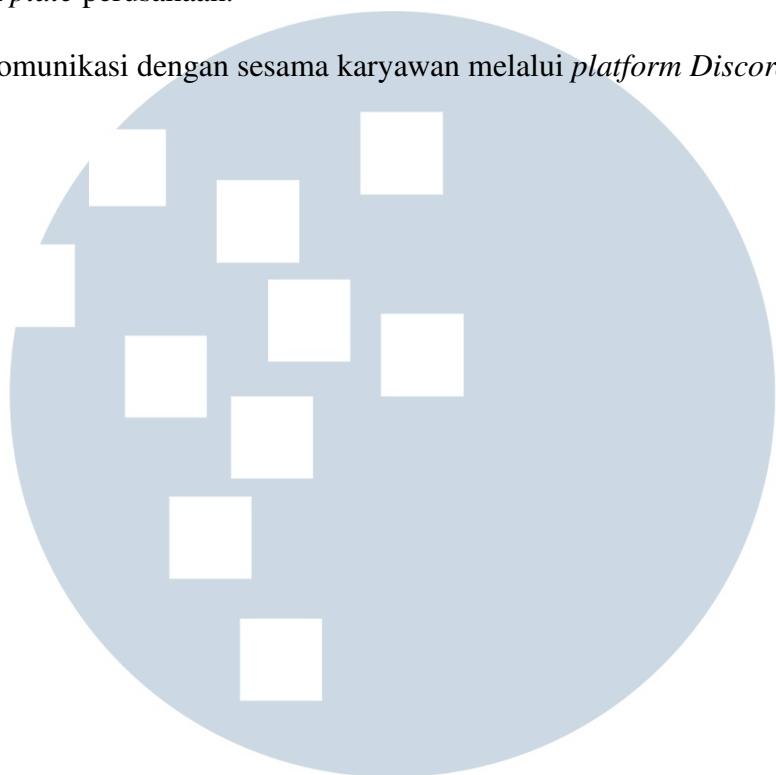
Pelaksanaan kerja magang berlangsung dari tanggal 13 Januari 2025 sampai dengan 13 Juli 2025 berdasarkan kontrak kerja yang telah disepakati dengan perusahaan. Selama periode magang ini dibimbing oleh seorang pembimbing lapangan yaitu Bapak Edo Setiawan yang menjabat sebagai Head Of Development di PT Ganda Visi Jayatama. Jadwal kerja magang di PT Ganda Visi Jayatama diatur sebagai berikut:

1. Aktivitas kerja magang dilaksanakan setiap hari Senin hingga Jumat, dengan jam kerja mulai pukul 09.00 WIB sampai dengan 18.00 WIB.
2. Pelaksanaan kerja magang dilakukan secara *Work From Office* (WFO).

Selama menjalani program kerja magang, terdapat sejumlah prosedur yang telah ditetapkan, antara lain:

1. Mengikuti sesi orientasi (*onboarding*) pada minggu pertama kerja magang.
2. Melakukan presensi harian dengan mencatat tugas yang telah diselesaikan pada hari sebelumnya (*yesterday tasks*), rencana aktivitas untuk hari ini (*today tasks*), serta kendala yang dihadapi dalam pengerjaan sebelumnya (*blocking*).
3. Berpartisipasi dalam rapat mingguan yang diadakan setiap hari Jumat untuk membahas perkembangan proyek HRIS yang sedang dikerjakan.

4. Menghadiri pertemuan bulanan untuk mendiskusikan pengembangan *boilerplate* perusahaan.
5. Berkommunikasi dengan sesama karyawan melalui *platform Discord*.



UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## BAB 2

### GAMBARAN UMUM PERUSAHAAN

#### 2.1 Sejarah Singkat Perusahaan



Gambar 2.1. Logo perusahaan PT Ganda Visi Jayatama

Gambar 2.1 merupakan logo dari perusahaan PT Ganda Visi Jayatama. Berdasarkan dokumen internal dari Universitas Multimedia Nusantara [8], PT Ganda Visi Jayatama merupakan sebuah perusahaan yang bergerak di bidang IT Consultant yang sebelumnya berlokasi di Ruko Golden 8 Blok K No. 25. Hingga saat ini, perusahaan telah beroperasi selama kurang lebih 3 tahun dan kini bertempat di Ruko Crystal No. 19 Gading Serpong. Selama periode tersebut, PT Ganda Visi Jayatama mengalami pertumbuhan pesat dengan memiliki sekitar 20 karyawan dari berbagai divisi.

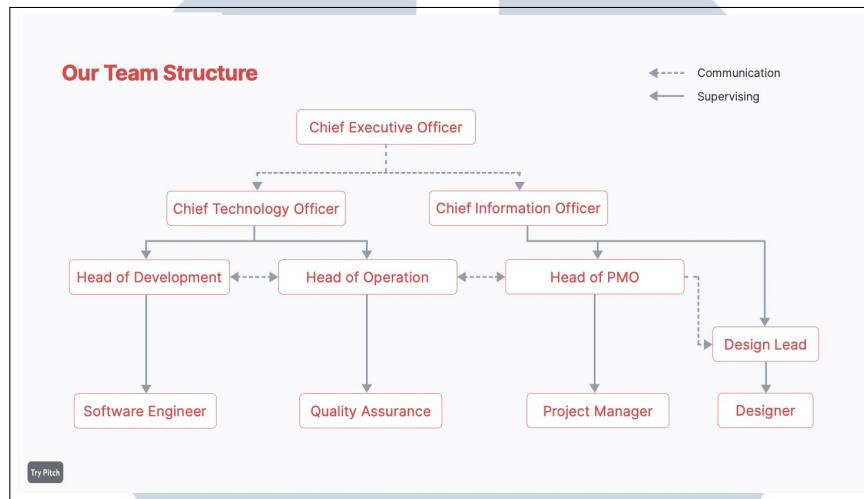
Perusahaan ini telah menyelesaikan berbagai proyek hingga tahun 2024, seperti Spectacle, Kecap Inggeris, Home Mart, PMI Company Profile, dan Habco Company Profile. Saat ini, PT Ganda Visi Jayatama tengah mengerjakan beberapa proyek, di antaranya Habco MRO Apps, PMI Autopos, Enigma, dan proyek-proyek lainnya.

#### 2.2 Visi dan Misi Perusahaan

Visi dari PT Ganda Visi Jayatama adalah "*to be the go-to IT consultancy firm for businesses looking to navigate the digital landscape with ease and confidence.*" Sedangkan misi dari perusahaan ini adalah "*to empower our clients by providing personalized IT solutions that enable them to streamline operations, enhance customer experiences, and drive growth. We accomplish this by staying at the forefront of technology trends, adhering to industry best practices, and leveraging a team of skilled and dedicated IT professionals. Ultimately, our goal is to help our clients succeed in an ever-changing digital world.*"

## 2.3 Struktur Organisasi Perusahaan

Struktur organisasi perusahaan PT Ganda Visi Jayatama dapat dilihat pada Gambar 2.2.



Gambar 2.2. Struktur organisasi perusahaan PT Visi Ganda Jayatama

Struktur organisasi PT Ganda Visi Jayatama dipimpin oleh seorang Chief Executive Officer (CEO) yang memiliki tanggung jawab utama dalam menetapkan arah dan strategi perusahaan secara keseluruhan. Di bawah CEO, terdapat dua posisi eksekutif penting, yaitu Chief Technology Officer (CTO) dan Chief Information Officer (CIO), yang masing-masing berfokus pada pengembangan teknologi dan pengelolaan sistem informasi perusahaan.

CTO membawahi dua divisi utama, yakni Head of Development dan Head of Operation. Head of Development memimpin tim Software Engineer yang bertugas dalam pengembangan perangkat lunak dan inisiatif teknologi. Sementara itu, Head of Operation bertanggung jawab atas pengelolaan tim Quality Assurance yang memastikan standar kualitas produk terpenuhi.

Di sisi lain, CIO membawahi Head of Project Management Office (PMO) yang memiliki peran penting dalam pengelolaan dan koordinasi seluruh proyek yang berjalan di perusahaan. CIO juga berkoordinasi langsung dengan Design Lead, yang memimpin tim Designer dalam mendukung kebutuhan visual dan antarmuka pengguna produk.

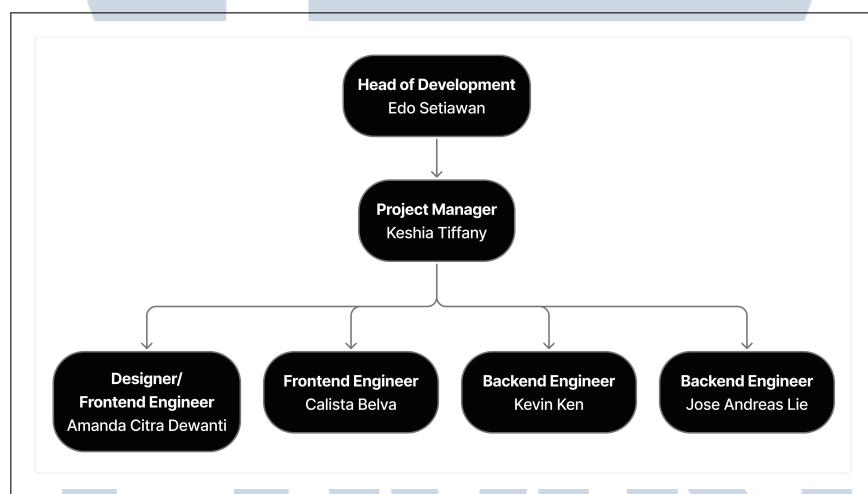
Struktur ini menggambarkan alur koordinasi dan supervisi yang sistematis antar posisi dan divisi, yang bertujuan untuk memperkuat kolaborasi lintas fungsi dalam mewujudkan tujuan strategis perusahaan.

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Selama pelaksanaan kerja magang di PT Ganda Visi Jayatama, peran yang dijalankan berada dalam tim pengembangan sistem CHRIS (Concise Human Resources Information System) sebagai *Backend Engineer Intern*. Tanggung jawab utamanya mencakup pengembangan dan pengujian *Application Programming Interface* (API) yang digunakan oleh sistem, serta kolaborasi dengan anggota tim lainnya dalam penyusunan dan penyempurnaan fitur-fitur sistem kepegawaian berbasis *web*.



Gambar 3.1. Struktur Tim CHRIS

Gambar 3.1 merupakan struktur dari tim CHRIS yang terdiri dari sejumlah anggota dengan peran yang saling terintegrasi. Bimbingan diberikan oleh Bapak Edo Setiawan selaku *Supervisor* sekaligus Head of Development, yang secara rutin melaksanakan evaluasi mingguan terhadap progres dan melakukan *code review* atas hasil pengembangan *backend*. Koordinasi teknis lebih lanjut dilaksanakan bersama Bapak Muhammad Alwin Alamsyah Handoko Putra selaku *Backend Lead*, yang memimpin diskusi internal tim *backend* setiap hari Jumat melalui *Backend Internal Meeting*. Perencanaan serta distribusi tugas dikoordinasikan oleh Project Manager, Ibu Keshia Tiffany, yang bertanggung jawab dalam pembagian *backlog* kepada anggota tim, serta mengadakan sesi evaluasi pribadi (*one-on-one*) dengan masing-masing anggota tim magang.

Dalam pengembangan tampilan antarmuka sistem, kolaborasi dilakukan bersama *Designer* Amanda Citra Dewanti yang merancang desain akhir dari *web*, serta dua *Frontend Engineer Intern*, Amanda Citra Dewanti dan Calista Belva, yang membangun antarmuka *web* menggunakan React. Sementara itu, pengembangan API menggunakan Express.js dan Node.js, serta pengelolaan basis data dengan PostgreSQL, dijalankan oleh dua Backend Engineer, yaitu Kevin Ken dan satu rekan lainnya dalam tim.

Seluruh kegiatan kerja magang dilakukan secara langsung di kantor (*Work From Office*). Koordinasi dilakukan melalui *daily standup* setiap pagi untuk melaporkan progres harian, menyampaikan rencana kerja, serta mendiskusikan kendala yang dihadapi. Setiap dua minggu sekali, tim juga melaksanakan *sprint retrospective* untuk mengevaluasi hasil kerja dalam satu *sprint* dan menentukan perbaikan serta target *sprint* berikutnya. Penugasan proyek dikelola menggunakan *platform* Jira (Atlassian) dalam bentuk *backlog sprint* yang dibagikan kepada setiap anggota tim secara terstruktur dan terukur.

### 3.2 Tugas yang Dilakukan

Selama pelaksanaan kerja magang di PT Ganda Visi Jayatama, terdapat tanggung jawab utama dalam satu proyek utama, yaitu pengembangan aplikasi *Internal System*. Tugas-tugas yang dijalankan selama magang terbagi ke dalam beberapa aktivitas utama sebagai berikut:

1. Mengembangkan API untuk kebutuhan aplikasi *Internal System*, yang mencakup pembuatan fitur-fitur backend sesuai dengan spesifikasi fungsional.
2. Melakukan dokumentasi terhadap API yang telah dikembangkan menggunakan *platform* dokumentasi API, Apidog.
3. Melakukan pengujian secara mandiri terhadap API yang dibuat untuk memastikan bahwa seluruh *endpoint* berjalan sesuai dengan fungsinya, serta menangani *error handling* dan validasi data.

### 3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

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

<b>Minggu Ke -</b>	<b>Pekerjaan yang dilakukan</b>
1	Mempelajari boilerplate backend dan mulai mengembangkan API untuk Employee Status pada sistem CHRIS.
2	Melanjutkan pengembangan dan penyempurnaan API Employee Status serta melakukan validasi ulang pada form Employee di sistem CHRIS.
3	Melakukan revisi minor pada API employee form, menyelesaikan tabel User dan Employee Status, serta berpartisipasi dalam Sprint Retro.
4	Mengembangkan fitur pagination untuk berbagai modul (User, Leave, Attendance), membuat API form pengajuan cuti, serta melakukan code review dan diskusi dalam monthly meeting.
5	Fokus pada revisi dan pengembangan API perizinan cuti, integrasi dengan frontend, serta showcase sistem CHRIS dan implementasi pagination untuk Leave Types.
6	Melakukan revisi dan filtering pada Leave Permit Dashboard, menambahkan fitur cancel, serta aktif dalam code review dan weekly meeting tim backend.
7	Melakukan berbagai pengujian dan UAT untuk Leave Management, membangun sistem tree berbasis jabatan untuk izin, serta menangani revisi migrasi dan API CHRISM (CHRIS Mobile).
8	Mengembangkan sistem hierarki supervisi berbasis tree, menerapkan biometrik pada login API, dan mulai membangun user report summary API serta mempersiapkan People Report.
9	Fokus pada penyempurnaan fitur User Report, termasuk penambahan filter tanggal dan perbaikan minor, serta melakukan hashing biometrik dan refactor pada dashboard Leave Permit.
10	Memulai riset intensif terkait sistem Payroll dan skema tabelnya, membuat dokumentasi di Apidog.
Lanjut pada halaman berikutnya	

Tabel 3.1 – Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
11	Melanjutkan pengembangan API Payroll berdasarkan hasil riset skema tabel, memperbarui dokumentasi di Apidog, serta mengikuti kegiatan Backlog Planning dan Sprint Closing.
12	Fokus pada pengembangan lanjutan API Payroll termasuk fitur Create, Get, Update, dan Delete, serta mulai menangani logika data untuk User Allowances.
13	Melanjutkan secara intensif pengembangan API Payroll khusus untuk pengelolaan dan perhitungan Each User Allowances secara berkelanjutan sepanjang minggu.
14	Mulai mengembangkan dan menyempurnakan Salary Slip APIs serta melakukan bugfix dan refactor pada User Allowance dan konfigurasi Payroll untuk integrasi dengan frontend.
15	Menambahkan fitur penghapusan User Allowance, memperbaiki konfigurasi endpoint Payroll, dan membuat API gabungan untuk manajemen detail user, payroll, serta tunjangan.
16	Melanjutkan integrasi Salary Slip dengan frontend serta melakukan pengujian menyeluruh terhadap modul Payroll, Allowance, dan Salary Slip.
17	Melakukan perbaikan pada logika dan pagination Salary Slip serta Payroll Config, merevisi sistem, dan menyiapkan internal report serta showcase Payroll.

### 3.4 Pengumpulan dan Analisis Kebutuhan

Kebutuhan sistem dalam proyek ini diperoleh melalui koordinasi langsung dengan *supervisor* dan tim *backend internal*. Sebagian besar *requirement* ditentukan secara iteratif berdasarkan kebutuhan bisnis dan sprint mingguan yang telah direncanakan oleh tim. Proses pengumpulan *requirement* dilakukan melalui diskusi teknis, *retrospective meeting*, dan *task assignment* harian.

Berikut ini adalah uraian *requirement* utama yang berhasil diidentifikasi dan diimplementasikan dalam proyek selama masa kerja praktik.

### A Refaktor User Management dan Validasi Data

Pengembangan dimulai dengan perbaikan sistem **User Management**, termasuk validasi *form input* dan *refactor* struktur tabel seperti *user* dan *employment status*. Hal ini bertujuan untuk memastikan integritas data pengguna dan kemudahan pengelolaan melalui *backend* maupun *frontend*.

### B Optimalisasi Leave Permit

Modul **Leave Permit** dikembangkan agar lebih efisien dan intuitif. Perubahan meliputi *refactor* pada proses *form submission*, penambahan tombol pembatalan (*cancel*) pengajuan cuti, serta tampilan daftar cuti untuk atasan. Fitur-fitur ini dirancang agar mencerminkan alur persetujuan yang realistik dan terstruktur.

### C Implementasi Pagination

Untuk mendukung jumlah data yang besar, sistem pagination ditambahkan pada beberapa modul utama seperti **User**, **Leave**, dan **Attendance**. Hal ini dilakukan guna menjaga performa dan kenyamanan pengguna.

### D Penambahan Fitur Biometrik untuk CHRIS Mobile

Fitur biometrik ditambahkan untuk mendukung proses autentikasi pada sistem CHRIS Mobile (CHRISM). Pengguna dapat melakukan login menggunakan data biometrik seperti sidik jari yang telah di-hash dan disimpan dalam kolom khusus pada tabel *users*. Fitur ini ditujukan untuk meningkatkan keamanan serta kenyamanan akses pengguna terhadap sistem.

### E Pengembangan Sistem Hierarki

Dibuat fungsi *tree hierarchy* berdasarkan struktur jabatan untuk mendukung fitur-fitur seperti izin cuti (*accept/reject*) dan tampilan dashboard atasan. Fungsi ini menjadi dasar logika akses dan pengelolaan hubungan antar pegawai.

## F Modul Payroll

Modul **Payroll** dikembangkan untuk menghasilkan slip gaji setiap bulannya yang dihitung berdasarkan tunjangan. Termasuk di dalamnya pengembangan API untuk CRUD *data payroll*, penyusunan *salary slip*, dan integrasi dengan *frontend*.

### 3.5 Perancangan dan Pengembangan Sistem

Bagian ini menjelaskan perancangan dan pengembangan sistem yang mencakup struktur basis data dan alur sistem untuk fitur-fitur yang dikembangkan selama masa kerja praktik.

#### 3.5.1 *User Management* dan Validasi Data

Sistem CHRIS telah dilengkapi dengan modul *User Management* yang berfungsi untuk mengelola data pegawai secara efisien. Modul ini mencakup fitur untuk menambahkan, memperbarui, dan menghapus data pegawai, serta melakukan validasi terhadap input yang dimasukkan melalui formulir. Validasi ini bertujuan untuk memastikan bahwa data yang diterima sesuai dengan format dan ketentuan yang berlaku. Meskipun demikian, sejumlah aspek dari modul ini memerlukan penyempurnaan guna meningkatkan integritas data dan mempermudah pengelolaan sistem.

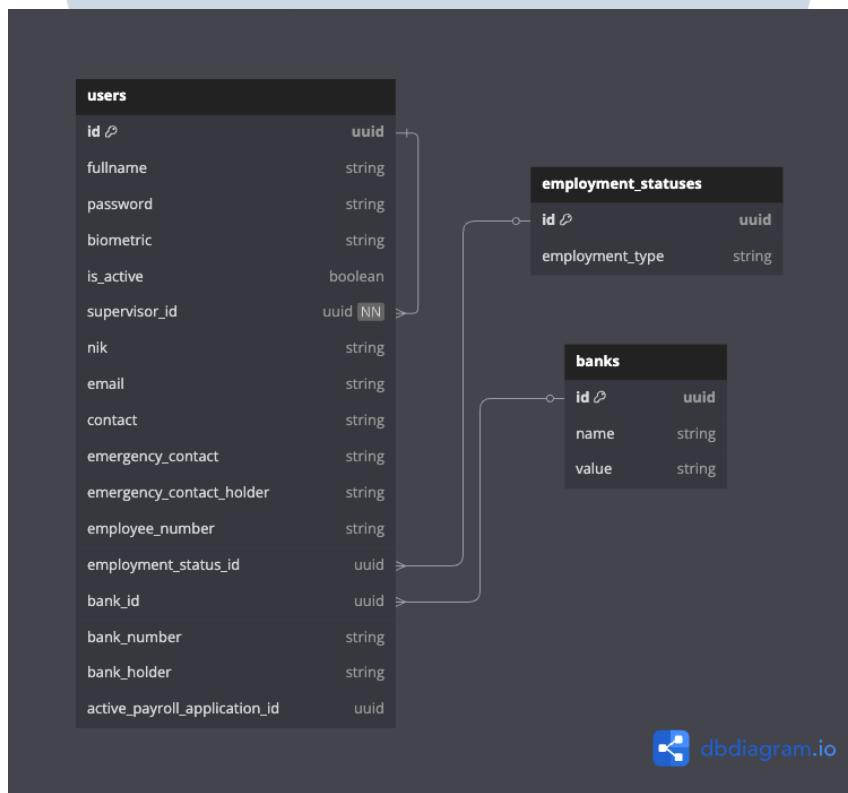
Adapun perbaikan dan pengembangan yang telah dilakukan antara lain:

- **Refactor User Management:** Alur pengelolaan data pegawai diperbarui agar setiap entri memiliki status kepegawaian yang terdefinisi dengan baik, sehingga struktur data menjadi lebih sistematis dan mudah diakses.
- **Validasi Form Input:** Validasi terhadap form input ditingkatkan, mencakup pengecekan format email, nomor telepon, serta memastikan bidang yang wajib diisi tidak terlewat, guna mencegah terjadinya inkonsistensi data.
- **Penyempurnaan Struktur Tabel:** Struktur tabel *users* diperbarui dengan menambahkan beberapa referensi eksternal untuk meningkatkan normalisasi data, antara lain:
  - Penambahan *employment\_status\_id* yang mereferensikan tabel *employment\_statuses*, menggantikan pendekatan enumerasi yang sebelumnya digunakan secara *hardcoded*.

- Penambahan *bank\_id* yang mereferensikan tabel *banks*, menggantikan kolom nama bank dalam bentuk *string* pada tabel *users* untuk menjamin konsistensi data dan memudahkan pengelompokan informasi perbankan.

Perubahan ini dilakukan sebagai bagian dari penerapan praktik terbaik dalam pengembangan sistem backend berbasis relasional. Selain itu, modifikasi ini juga memberikan fleksibilitas lebih tinggi dalam pengelolaan data serta meningkatkan skalabilitas modul User Management dalam jangka panjang.

## A Diagram ERD User Management



Gambar 3.2. Diagram ERD untuk modul User Management

Gambar 3.2 menunjukkan struktur basis data untuk modul User Management yang telah dimodifikasi. Terdapat beberapa tabel utama yang saling berhubungan, yaitu:

- **Users:** Tabel ini menyimpan data pegawai, termasuk informasi pribadi, status kepegawaian, dan referensi bank.

- **Employment Statuses:** Tabel ini menyimpan berbagai status kepegawaian yang dapat dimiliki oleh pegawai, seperti aktif, cuti, atau tidak aktif.
- **Banks:** Tabel ini menyimpan informasi mengenai bank yang digunakan oleh pegawai untuk penggajian.

Sebelumnya modul User Management menggunakan pendekatan *hardcoded* untuk status kepegawaian dan bank, namun kini telah diubah menjadi referensi tabel yang lebih fleksibel. Hal ini memungkinkan penambahan atau perubahan status kepegawaian dan bank tanpa perlu mengubah kode sumber, sehingga meningkatkan efisiensi pengelolaan data.

## B Validasi Data pada Formulir User Management

Validasi data pada formulir *User Management* dilakukan untuk menjamin integritas, konsistensi, dan keamanan data yang masuk ke dalam sistem. Validasi dilakukan baik di sisi *frontend* maupun di sisi *backend*, dengan ketentuan sebagai berikut:

- **Fullname:** Nama lengkap pegawai harus diisi dengan format yang benar, yaitu merupakan huruf *alphanumeric*. Validasi ini bertujuan untuk memastikan bahwa nama pegawai dapat dikenali dan diidentifikasi dengan jelas dalam sistem.
- **Email:** Hanya alamat email dengan domain `@concise.co.id` yang diperbolehkan. Validasi ini diterapkan untuk memastikan bahwa hanya pegawai internal yang terdaftar di sistem. Format email juga diverifikasi menggunakan ekspresi reguler untuk menghindari entri tidak valid.
- **NIK:** Nomor Induk Kependudukan (NIK) harus diisi dengan format yang benar, yaitu terdiri dari 16 digit angka. Validasi ini penting untuk memastikan bahwa NIK yang dimasukkan sesuai dengan standar yang berlaku di Indonesia.
- **Contact:** Hanya nomor telepon yang dimulai dengan *prefix* +62 atau 0 yang diterima. Validasi ini bertujuan untuk memastikan bahwa nomor telepon yang dimasukkan sesuai dengan format nomor telepon pada umumnya.

- **Emergency Contact:** Sama seperti nomor telepon, hanya nomor yang dimulai dengan *prefix* +62 atau 0 yang diterima. Hal ini untuk memastikan bahwa kontak darurat yang dimasukkan dapat dihubungi dengan mudah.
- **Bank Holder:** Nama pemegang rekening bank harus diisi dengan format yang benar, yaitu merupakan huruf *alphanumeric*. Validasi ini bertujuan untuk memastikan bahwa nama pemegang rekening sesuai dengan nama pegawai yang terdaftar dalam sistem.
- **Bank Account Number:** Validasi dilakukan untuk memastikan bahwa nomor rekening bank yang dimasukkan hanya terdiri dari angka. Hal ini penting untuk menghindari kesalahan dalam proses pengajian.

Validasi ini tidak hanya berfungsi untuk memperbaiki pengalaman pengguna, tetapi juga mencegah terjadinya kesalahan logika dan duplikasi data di tingkat basis data. Seluruh ketentuan ini dirancang berdasarkan standar praktik terbaik dalam pengelolaan data karyawan di lingkungan perusahaan.

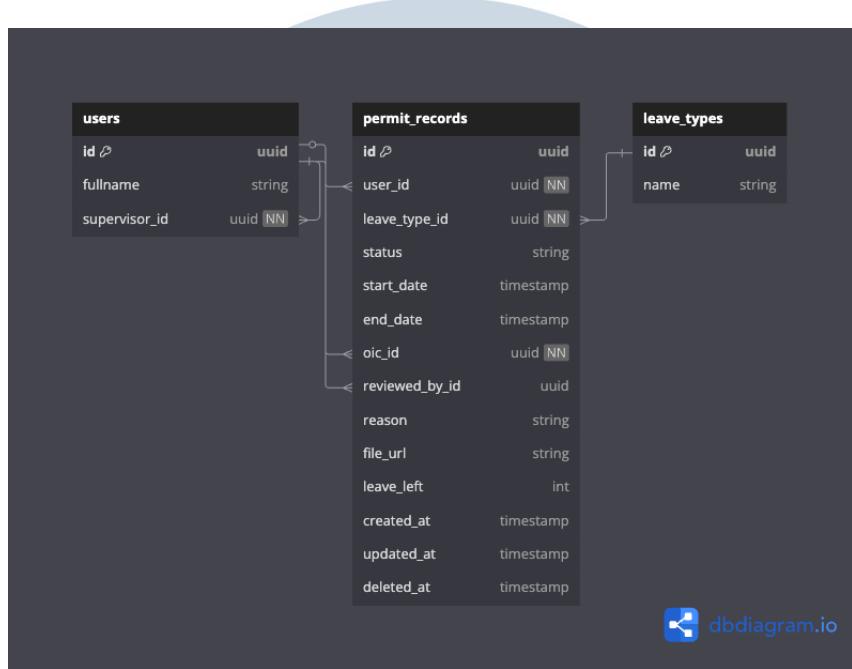
### 3.5.2 Leave Permit

Modul *Leave Permit* merupakan fitur yang memungkinkan pegawai mengajukan permohonan cuti, serta memberikan wewenang kepada atasan untuk menyetujui atau menolak permohonan tersebut. Setiap jenis cuti memiliki jatah tersendiri, dan modul ini juga berfungsi untuk menghitung secara otomatis sisa cuti yang dimiliki oleh masing-masing pegawai. Sistem ini dirancang untuk mencerminkan alur persetujuan yang terstruktur dan realistik, dengan memperhatikan hierarki jabatan di dalam perusahaan.

Sistem ini sudah pernah digunakan sebelumnya, namun mengalami beberapa kendala yang perlu diperbaiki. Beberapa perbaikan yang dilakukan antara lain adalah:

- **Refactor Form Submission:** Proses pengajuan cuti ditambahkan *officer in charge (oic)* dengan tujuan sebagai pengganti pegawai saat ia cuti.
- **Cancel Button:** Ditambahkan fitur pembatalan (*cancel*) pengajuan cuti, sehingga pegawai dapat membatalkan permohonan yang belum disetujui.
- **Leave Permit Dashboard:** Tampilan daftar cuti ditampilkan di *home page* supaya semua pegawai dapat melihat siapa saja yang mengajukan cuti di minggu itu.

## A Diagram ERD Leave Permit

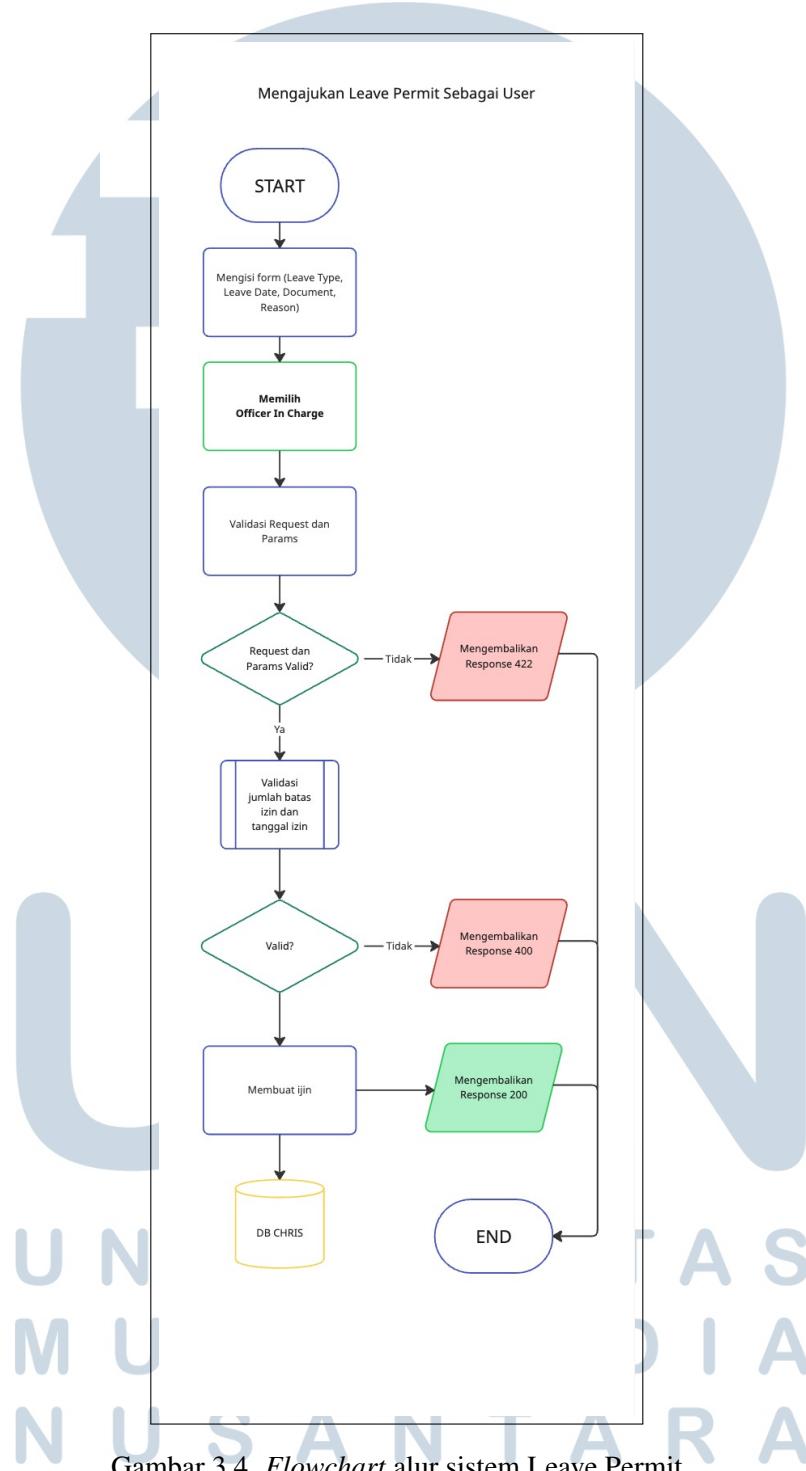


Gambar 3.3. Diagram ERD untuk modul Leave Permit

Gambar 3.3 menunjukkan struktur basis data untuk modul Leave Permit yang dimodifikasi. Terdapat beberapa tabel utama yang saling berhubungan, yaitu:

- **User**: Tabel ini menyimpan data pegawai yang mengajukan cuti, termasuk informasi pribadi dan status kepegawaian.
- **Leave Types**: Tabel ini menyimpan jenis-jenis cuti yang tersedia, termasuk nama, deskripsi, dan jatah cuti yang diberikan kepada pegawai.
- **Permit Records**: Tabel ini menyimpan data permohonan cuti yang diajukan oleh pegawai, termasuk tanggal pengajuan, tanggal mulai dan selesai cuti, status persetujuan, dan pegawai lain yang mengantikannya sebagai *office in charge*.

## B Alur Sistem Leave Permit



Gambar 3.4. Flowchart alur sistem Leave Permit

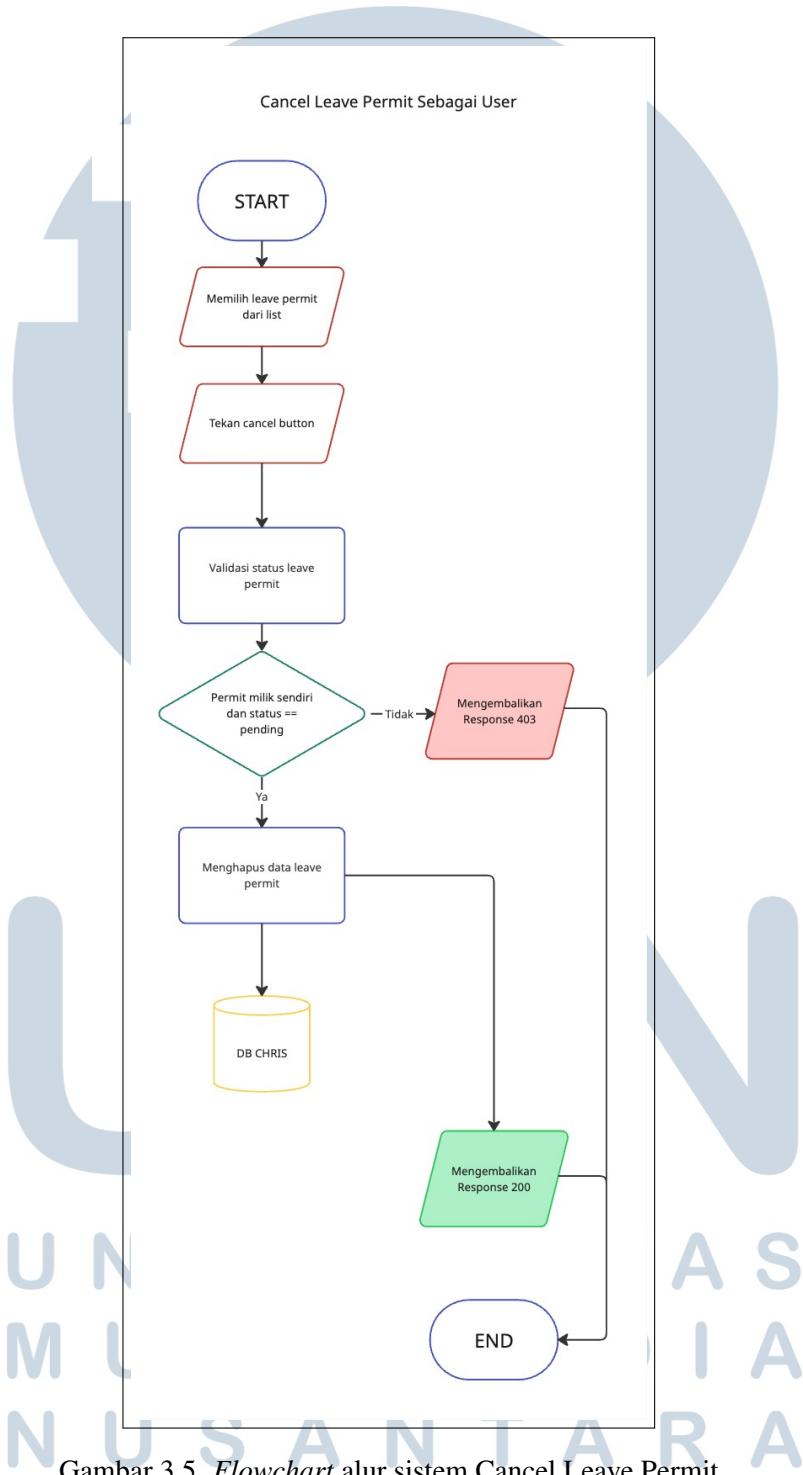
Gambar 3.4 menunjukkan alur sistem *Leave Permit* yang diawali oleh pegawai yang mengajukan cuti melalui formulir yang tersedia. Informasi yang

diisi mencakup jenis cuti, rentang tanggal, alasan pengajuan, serta penunjukan *officer in charge* sebagai pengganti selama periode cuti. Setelah permohonan dikirimkan, sistem akan menyimpan data ke dalam tabel *Permit Records*, dan atasan dapat memberikan persetujuan atau penolakan. Selama permohonan belum diproses, pegawai memiliki opsi untuk membatalkannya. Jika disetujui, sistem secara otomatis akan memperbarui sisa jatah cuti sesuai jenis cuti yang diajukan. Status pengajuan dapat dipantau melalui dashboard yang menampilkan daftar cuti yang aktif dalam minggu berjalan.

Sebagai bagian dari pengembangan lanjutan modul ini, dilakukan penyesuaian struktur data dengan menambahkan kolom *oic\_id* pada tabel *Permit Records*. Penambahan ini ditujukan untuk memenuhi kebutuhan bisnis dalam menjamin kesinambungan operasional saat pegawai cuti, dengan menunjuk rekan kerja yang bertanggung jawab selama periode tersebut. Perubahan ini turut memperkuat logika bisnis sistem dan memastikan distribusi tugas tetap berjalan secara efisien.



### C Alur Sistem Cancel Leave Permit



Gambar 3.5. Flowchart alur sistem Cancel Leave Permit

Gambar 3.5 menggambarkan alur proses pembatalan permohonan cuti oleh pegawai. Setelah pengajuan dilakukan, pegawai dapat membatalkan permohonan

selama statusnya belum disetujui oleh atasan. Permintaan pembatalan dikirim melalui formulir yang tersedia, kemudian sistem akan memverifikasi status permohonan. Jika permohonan belum disetujui, sistem akan melakukan *soft delete* pada data di tabel *Permit Records*. Namun, apabila permohonan telah disetujui, sistem akan menolak proses pembatalan dan menampilkan notifikasi kesalahan bahwa pengajuan tidak dapat dibatalkan.

## D Alur Sistem Leave Permit Dashboard

Alur pengambilan dan penampilan data cuti pada halaman *dashboard* digunakan untuk menampilkan pegawai-pegawai yang sedang cuti di minggu berjalan. Proses dimulai saat pegawai mengakses dashboard dan sistem kemudian melakukan query terhadap data *permit records* yang memiliki rentang tanggal cuti berada dalam minggu berjalan dan status pengajuan sudah aktif. Data yang diambil mencakup nama pegawai, jenis cuti, tanggal mulai dan selesai, serta *officer in charge* yang ditunjuk.

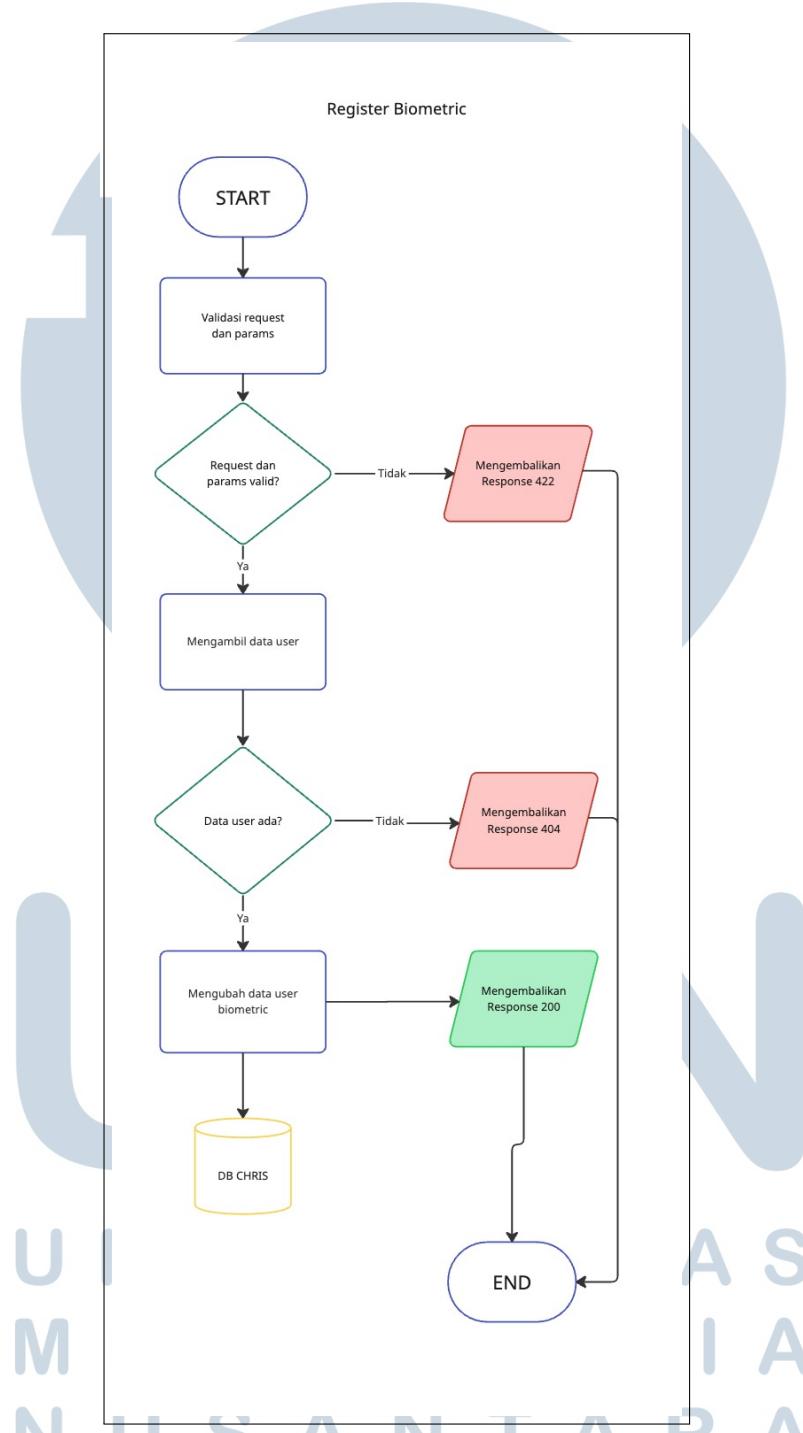
Informasi tersebut disajikan dalam bentuk tabel agar mudah dipahami dan dapat digunakan oleh pegawai untuk mengetahui siapa saja yang sedang atau akan cuti, serta mengetahui siapa rekan pengganti yang dapat dihubungi untuk keperluan operasional. Fitur ini ditujukan untuk meningkatkan transparansi dan mendukung koordinasi lintas tim selama periode cuti berlangsung.

### 3.5.3 Fitur Autentikasi Biometrik pada CHRIS Mobile

Sebagai bagian dari pengembangan sistem CHRIS Mobile (CHRISM), ditambahkan fitur autentikasi berbasis biometrik untuk meningkatkan kenyamanan, kecepatan dan keamanan akses pengguna. Implementasi ini dilakukan dengan menambahkan kolom baru biometric pada tabel *users*. Kolom ini menyimpan hasil *hash* sepanjang maksimal 255 karakter dari data biometrik pengguna seperti sidik jari.

Fitur ini memberikan alternatif *login* selain kata sandi serta mendukung praktik keamanan modern, termasuk *multi-factor authentication*. Karena data biometrik telah melalui proses hashing, informasi yang disimpan tetap aman dan tidak dapat digunakan kembali secara langsung. Fitur ini hanya tersedia pada aplikasi CHRIS Mobile dan tidak memengaruhi sistem versi web atau *desktop*.

## A Alur Register Biometrik



Gambar 3.6. Flowchart alur sistem registrasi biometrik pada CHRIS Mobile

Gambar 3.6 menunjukkan alur sistem registrasi dan autentikasi biometrik pada aplikasi CHRIS Mobile (CHRISM). Proses dimulai ketika pengguna

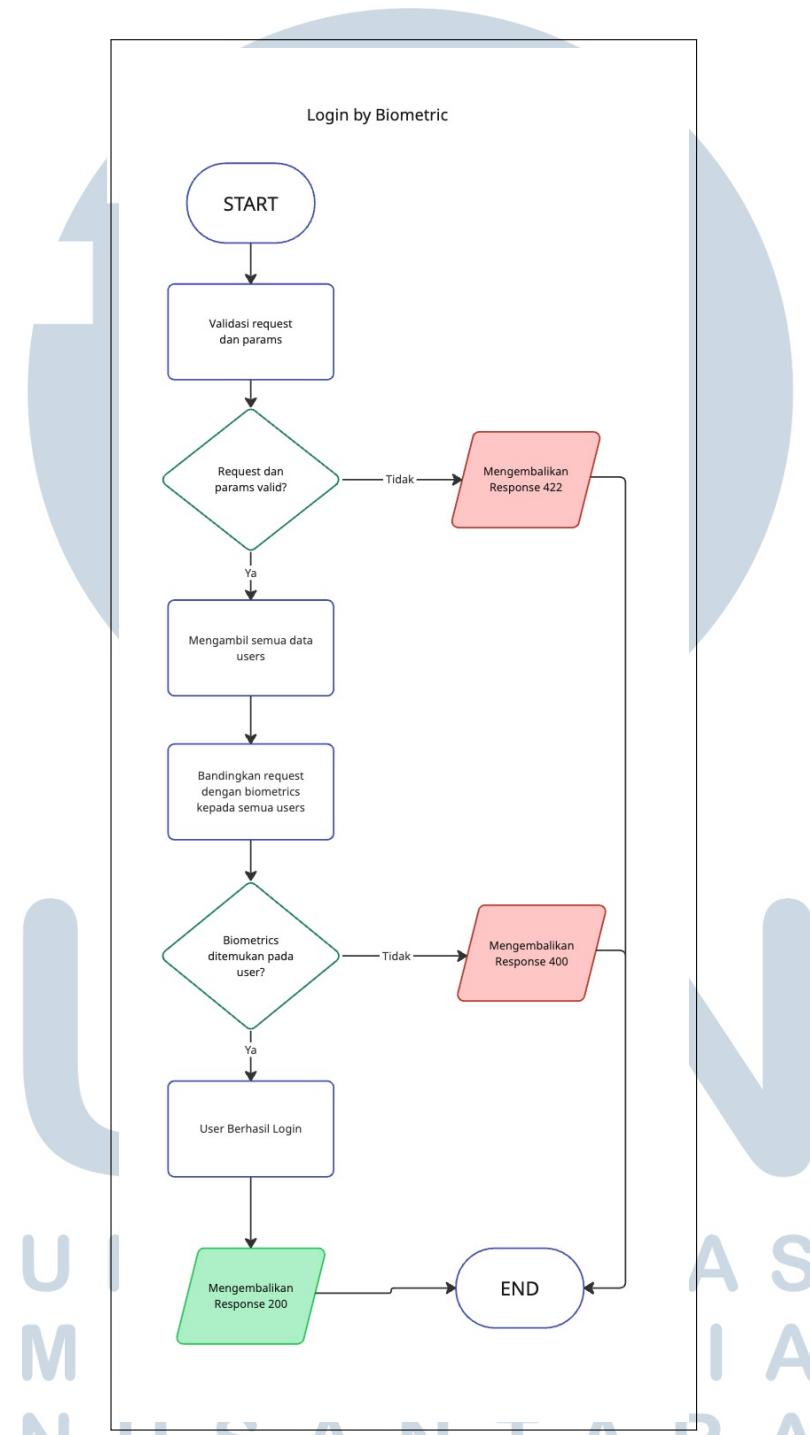
mengakses halaman profil dan memilih opsi “*Activate Biometric*”. Setelah itu, aplikasi akan memicu pemindaian biometrik menggunakan sensor sidik jari pada perangkat. Jika proses pemindaian berhasil, sistem akan secara otomatis menghasilkan *random string* sepanjang 255 karakter yang mewakili identitas biometrik pengguna. Nilai ini kemudian dikirimkan ke *backend* melalui API khusus untuk proses pendaftaran biometrik.

Di sisi *backend*, data tersebut akan di-hash dan disimpan pada kolom *biometric* di tabel *users* untuk keperluan autentikasi selanjutnya.

Dalam implementasi autentikasi, saat aplikasi dibuka, CHRISM akan kembali meminta verifikasi biometrik dari perangkat. Jika sidik jari cocok, sistem akan menembakkan *payload* berupa *random string* 255 karakter yang identik dengan yang telah didaftarkan sebelumnya, lalu mengirimkannya ke *endpoint login biometrik*. *Backend* akan mencocokkan hasil *hash* dari *string* tersebut dengan data yang tersimpan di basis data. Jika sesuai, maka autentikasi dinyatakan berhasil dan pengguna dapat langsung masuk ke sistem tanpa perlu menggunakan kata sandi. Proses ini dirancang untuk meningkatkan keamanan serta memberikan pengalaman masuk aplikasi yang lebih praktis dan efisien bagi pengguna CHRISM.



## B Alur Autentikasi Login Menggunakan Biometrik



Gambar 3.7. Flowchart alur sistem autentikasi biometrik pada CHRIS Mobile

Gambar 3.7 menunjukkan alur sistem autentikasi biometrik pada CHRIS Mobile. Proses dimulai ketika pengguna memilih opsi *login* menggunakan

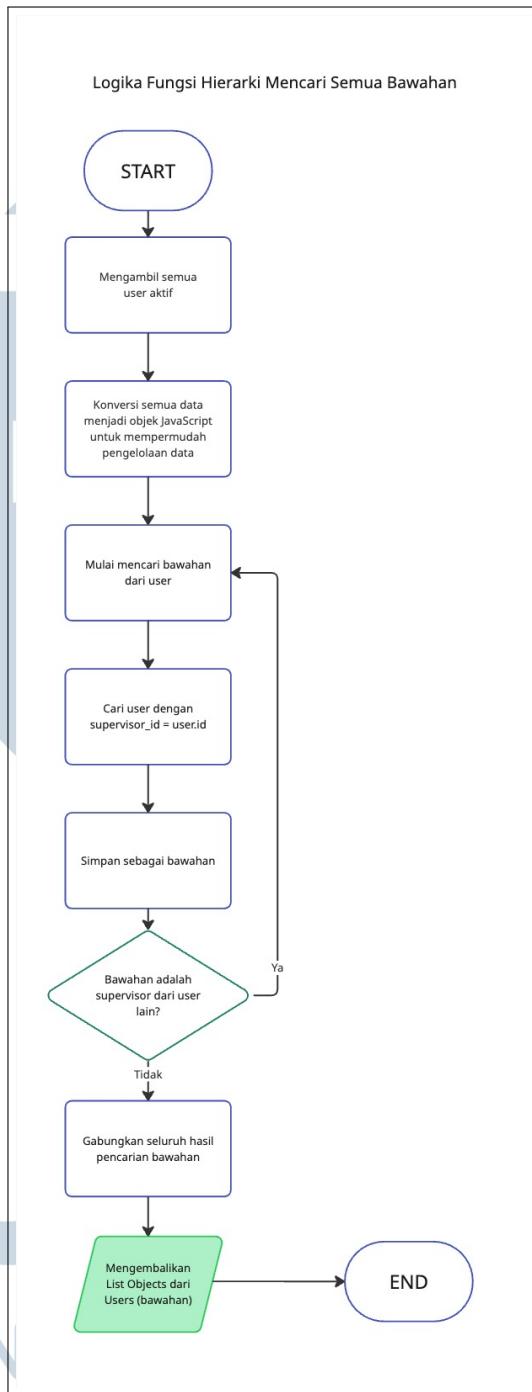
biometrik. Sistem kemudian akan meminta data biometrik dari perangkat, yang selanjutnya di-hash dan dibandingkan dengan data yang tersimpan di basis data. Jika cocok, pengguna akan berhasil masuk ke dalam aplikasi. Jika tidak, sistem akan menampilkan pesan kesalahan dan meminta pengguna untuk mencoba kembali. Fitur ini dirancang untuk memberikan pengalaman pengguna yang lebih cepat dan aman, serta mengurangi ketergantungan pada kata sandi yang dapat dilupakan atau dicuri.

#### 3.5.4 Sistem Hierarki Supervisi

Sistem CHRIS menerapkan struktur hierarki berbasis pohon (*tree hierarchy*) untuk mengelola hubungan antara pegawai dan atasan. Modul-modul dalam sistem ini, seperti pengajuan cuti, bergantung pada struktur tersebut, di mana permohonan cuti hanya dapat disetujui oleh atasan langsung dari pegawai yang bersangkutan.

Diagram pada Gambar 3.8 menggambarkan alur logika sistem dalam mencari seluruh bawahan dari seorang pegawai. Fungsi ini dimulai dengan mengambil seluruh data pengguna dari basis data, kemudian melakukan pencarian rekursif terhadap pegawai yang memiliki *supervisor\_id* yang sesuai dengan *id* pegawai tersebut. Pencarian dilakukan secara berlapis hingga seluruh struktur bawahan ditemukan.





Gambar 3.8. Logika Fungsi Hierarki untuk Mencari Semua Bawahan

Dengan pendekatan ini, sistem mampu menentukan siapa saja yang berada dalam rantai struktur supervisi, baik secara langsung maupun tidak langsung. Hal ini memungkinkan sistem untuk secara efisien menetapkan pihak yang berwenang dalam proses seperti persetujuan cuti, pelacakan struktur organisasi, maupun

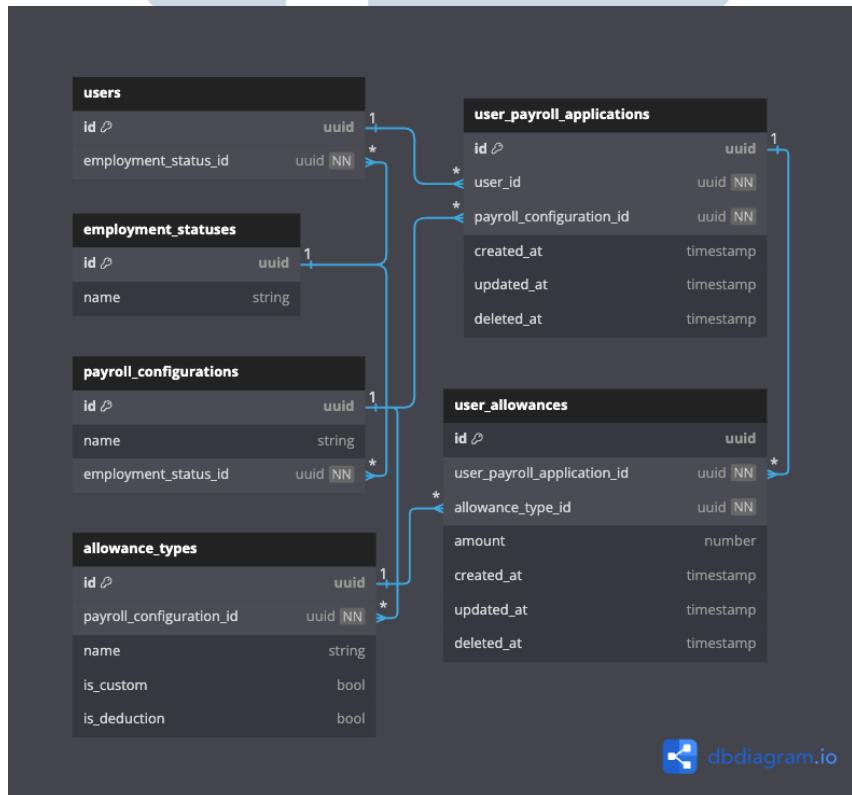
pengelolaan akses modul internal.

Fungsi ini telah digunakan secara langsung dalam modul *Leave Permit* untuk memastikan bahwa pengajuan cuti hanya dapat ditinjau dan disetujui oleh atasan yang sesuai. Selain itu, logika ini juga dapat diterapkan pada modul lain yang membutuhkan pemetaan hubungan antarpegawai secara hierarkis, seperti monitoring kinerja, delegasi tugas, atau manajemen tim lintas divisi.

### 3.5.5 Payroll

Modul *Payroll* merupakan salah satu fitur utama dalam sistem CHRIS. Modul ini bertujuan untuk mengelola data penggajian pegawai, termasuk perhitungan gaji berdasarkan tunjangan yang telah ditentukan.

#### A Diagram ERD Payroll



Gambar 3.9. Diagram ERD untuk modul Payroll

Struktur basis data untuk modul Payroll terdiri dari beberapa tabel utama yang saling berhubungan. Berikut adalah penjelasan singkat mengenai tabel-tabel

tersebut:

- **User:** Tabel ini menyimpan data pegawai yang mencakup informasi pribadi, status kepegawaian, dan referensi ke konfigurasi penggajian yang digunakan.
- **Employment Status:** Tabel ini menyimpan data status kepegawaian yang digunakan sebagai referensi pada berbagai modul dalam sistem, salah satunya adalah modul *Payroll*.
- **Payroll Configuration:** Tabel ini menyimpan konfigurasi penggajian yang mencakup nama, status kepegawaian, dan tunjangan yang berlaku. Setiap konfigurasi dapat memiliki beberapa tunjangan yang terkait.
- **Allowances Types:** Tabel ini menyimpan jenis-jenis tunjangan yang tersedia pada payroll configuration yang telah dibuat, dan juga tunjangan tambahan untuk pegawai tertentu. Setiap jenis tunjangan memiliki nama, dan tipe (tunjangan atau potongan).
- **User Payroll Application:** Tabel ini menyimpan data Payroll yang telah diisi oleh *Superadmin* untuk setiap pegawai.
- **User Allowances:** Tabel ini menyimpan data tunjangan spesifik untuk setiap pegawai. Tabel ini berisi informasi mengenai jenis tunjangan, jumlah, dan referensi ke pegawai yang bersangkutan.

## B Alur Sistem Payroll

Alur sistem *Payroll* diawali dengan pembuatan data *Payroll Configuration*, yang mencakup nama konfigurasi, status kepegawaian (*Employment Status*), serta daftar tunjangan (*Allowances*) yang berlaku. Setelah konfigurasi dibuat, *Superadmin* melanjutkan ke modul *User Management* untuk mengatur data gaji setiap pegawai secara individual.

Dalam modul *User Management*, *Superadmin* memilih *Payroll Configuration* berdasarkan status kepegawaian pengguna, mengisi besaran gaji pokok, serta melengkapi jumlah masing-masing tunjangan yang ditetapkan. Selain itu, *Superadmin* juga dapat menambahkan tunjangan (*allowance*) atau potongan (*deduction*) khusus yang hanya berlaku bagi pengguna tersebut, guna menyesuaikan skema gaji secara fleksibel.

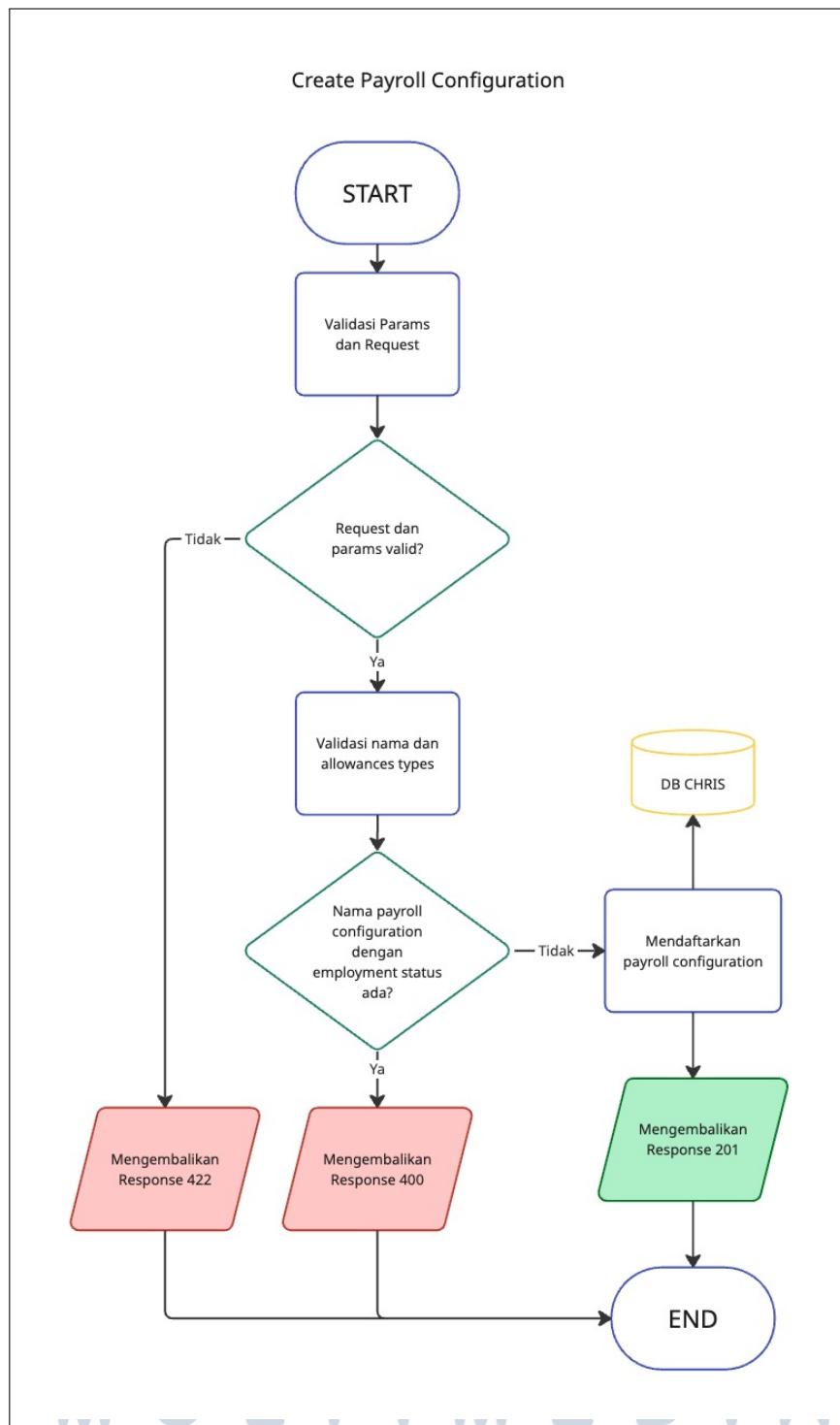
Setelah data selesai disimpan, *Superadmin* dapat mengakses modul *Salary Slip* untuk melakukan finalisasi gaji. Finalisasi ini memungkinkan pengecekan akhir terhadap rincian gaji sebelum tanggal gajian. Di PT Ganda Visi Jayatama, proses penggajian dilakukan setiap tanggal 25, sehingga proses finalisasi disarankan dilakukan pada tanggal 24 setiap bulannya. Setelah tanggal 25, data tidak dapat lagi diubah.

Pegawai yang telah memiliki data gaji terverifikasi dapat melihat slip gaji mereka masing-masing pada halaman *Salary Slip* dan mengunduhnya dalam format PDF.

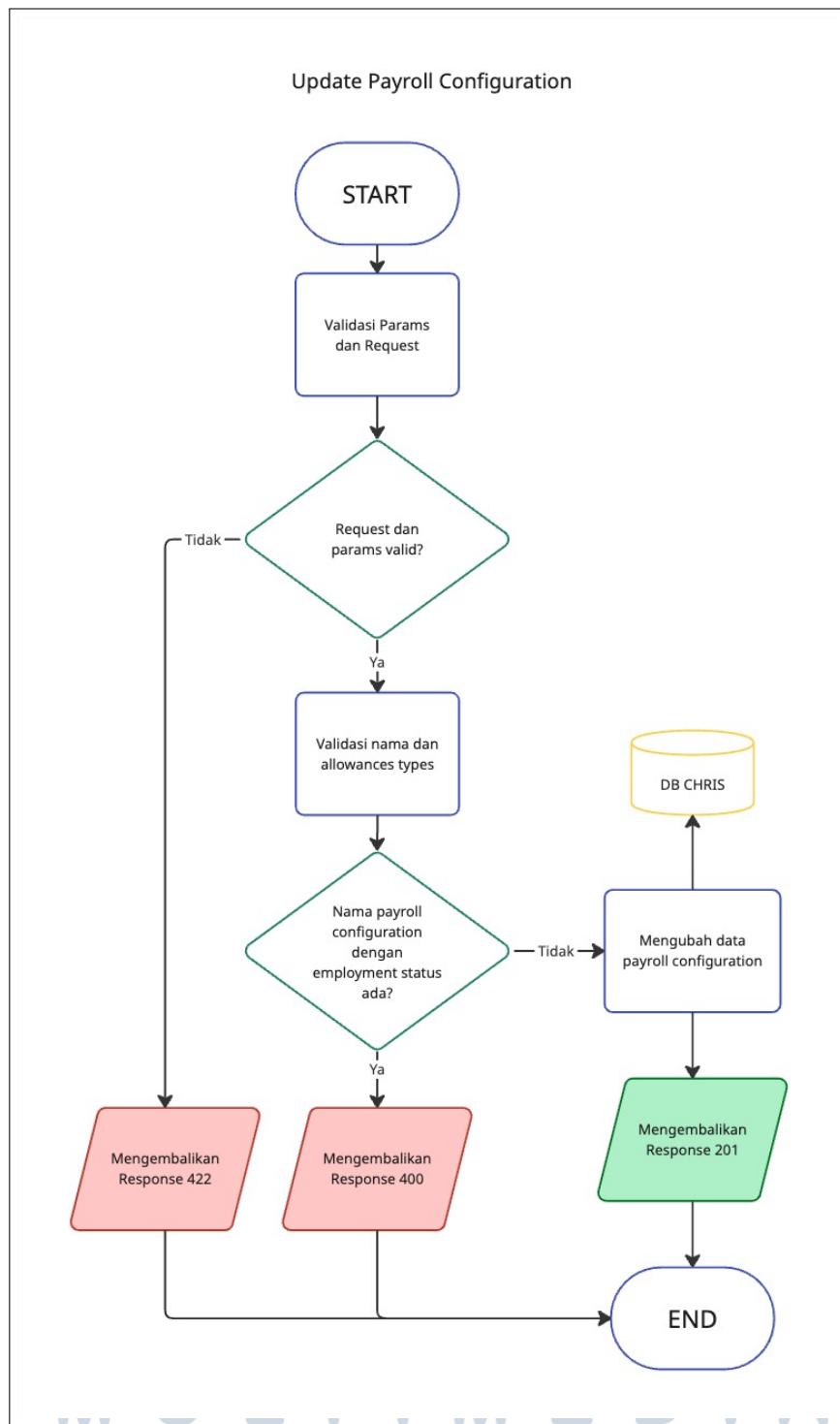
### B.1 Payroll Configuration

Pada Gambar ??, ??, dan ?? merupakan alur sistem untuk Payroll Configuration yang dimulai dari pembuatan konfigurasi penggajian, di mana *Superadmin* membuat konfigurasi baru dengan mengisi nama konfigurasi, status kepegawaian, dan daftar tunjangan (*Allowances*) yang berlaku. Setelah itu, *Superadmin* dapat mengakses modul *User Management* untuk mengatur data gaji setiap pegawai.

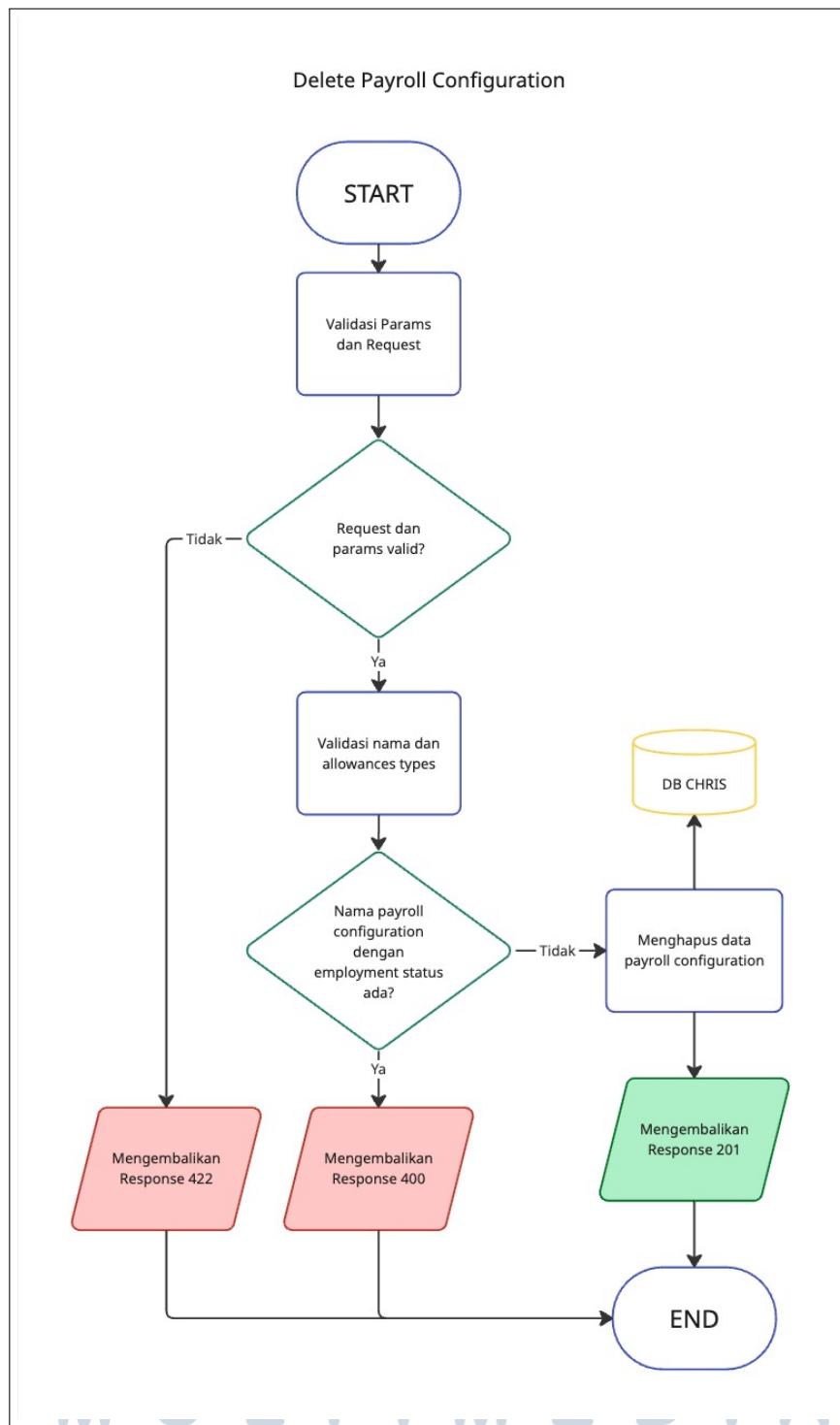




**NUSANTARA**  
Gambar 3.10. Flowchart create payroll configuration



**NUSANTARA**  
Gambar 3.11. Flowchart update payroll configuration



**NUSANTARA**

Gambar 3.10, 3.11, dan 3.12 menggambarkan alur proses pembuatan, pengubahan, dan penghapusan data *Payroll Configuration*. Ketiga proses tersebut menerapkan validasi yang sama, yaitu pengecekan *request* dan *params*, dan

pengecekan terhadap kombinasi nama Payroll Configuration dan Employment Status yang sudah terdaftar sebelumnya.

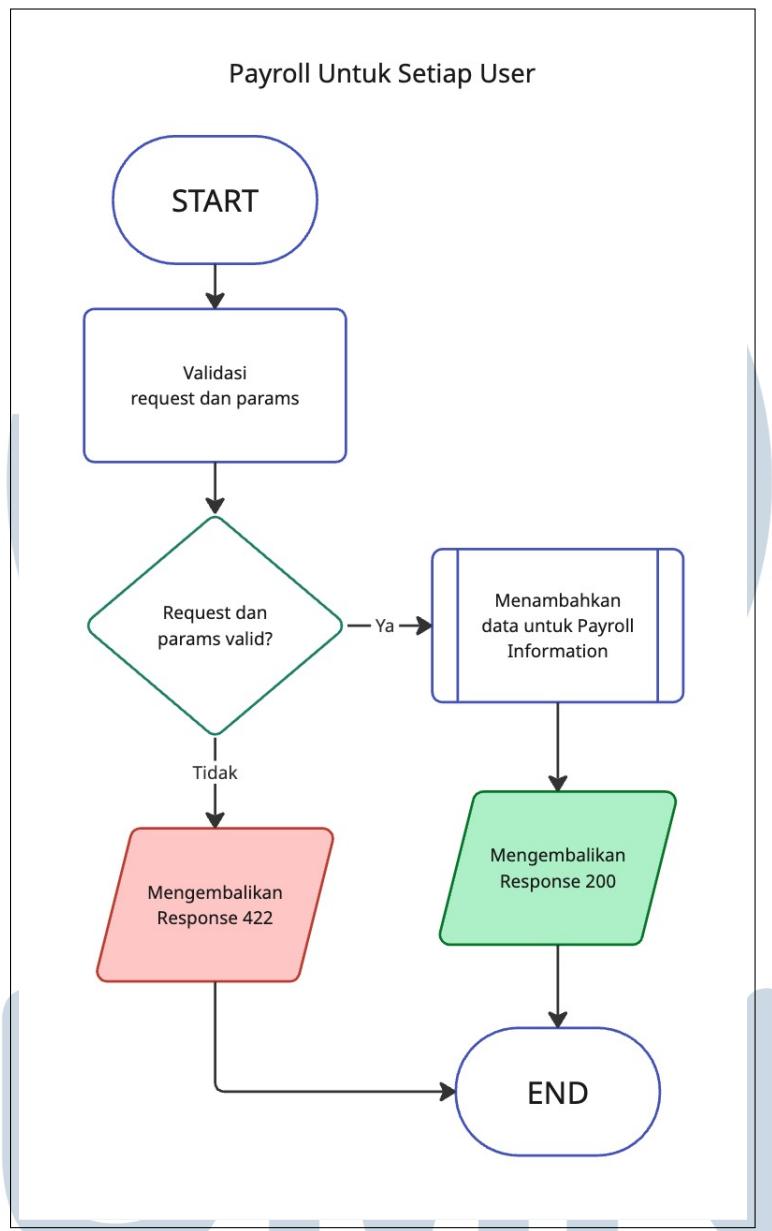
Apabila *request* atau *parameter* yang dikirimkan tidak sesuai dengan format atau aturan yang telah ditentukan, sistem akan merespons dengan kode 422 (*Invalid Format*) sebagai penolakan terhadap permintaan yang tidak valid. Selain itu, jika kombinasi nama *Payroll Configuration* dan *Employment Status* telah terdaftar sebelumnya, sistem akan mengembalikan respons kode 400 (*Bad Request*) untuk mencegah terjadinya duplikasi data.

Validasi ini bertujuan untuk menjaga konsistensi dan integritas data dalam sistem. Jika seluruh validasi berhasil dilewati, maka proses pembuatan, pengubahan, atau penghapusan akan dilanjutkan, dengan sistem memberikan respons berupa kode 201 (*Created*) untuk pembuatan, serta kode 200 (*OK*) untuk pengubahan dan penghapusan data.

## B.2 User Allowances

Setelah *Superadmin* membuat *payroll configuration*, langkah selanjutnya adalah mengatur data gaji setiap pegawai. Proses ini dilakukan melalui modul *User Management*, di mana *Superadmin* memilih *Payroll Configuration* berdasarkan status kepegawaian pengguna, mengisi besaran gaji pokok, serta melengkapi jumlah masing-masing tunjangan yang ditetapkan.

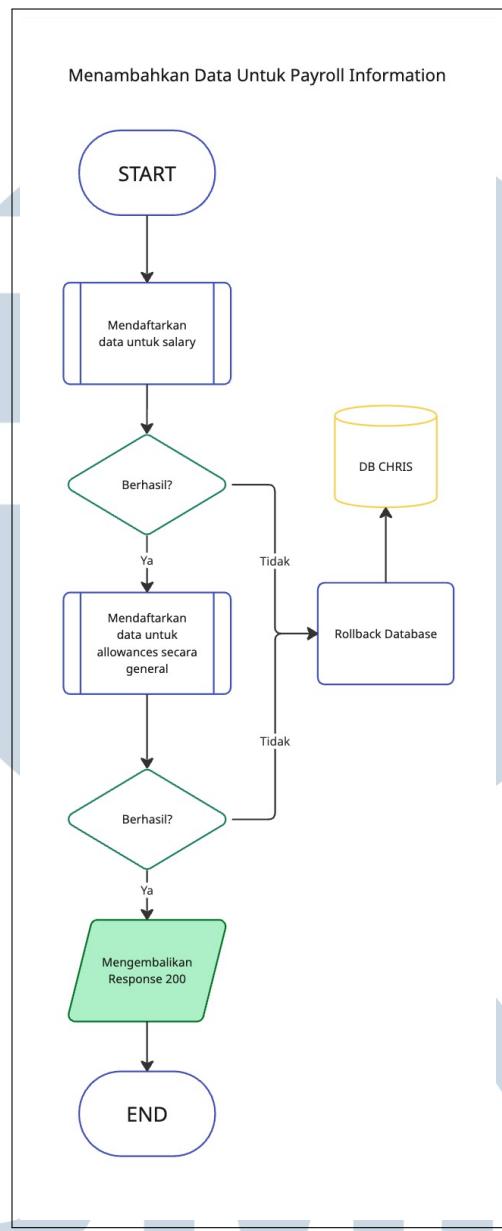




Gambar 3.13. Flowchart implementasi payroll untuk setiap user

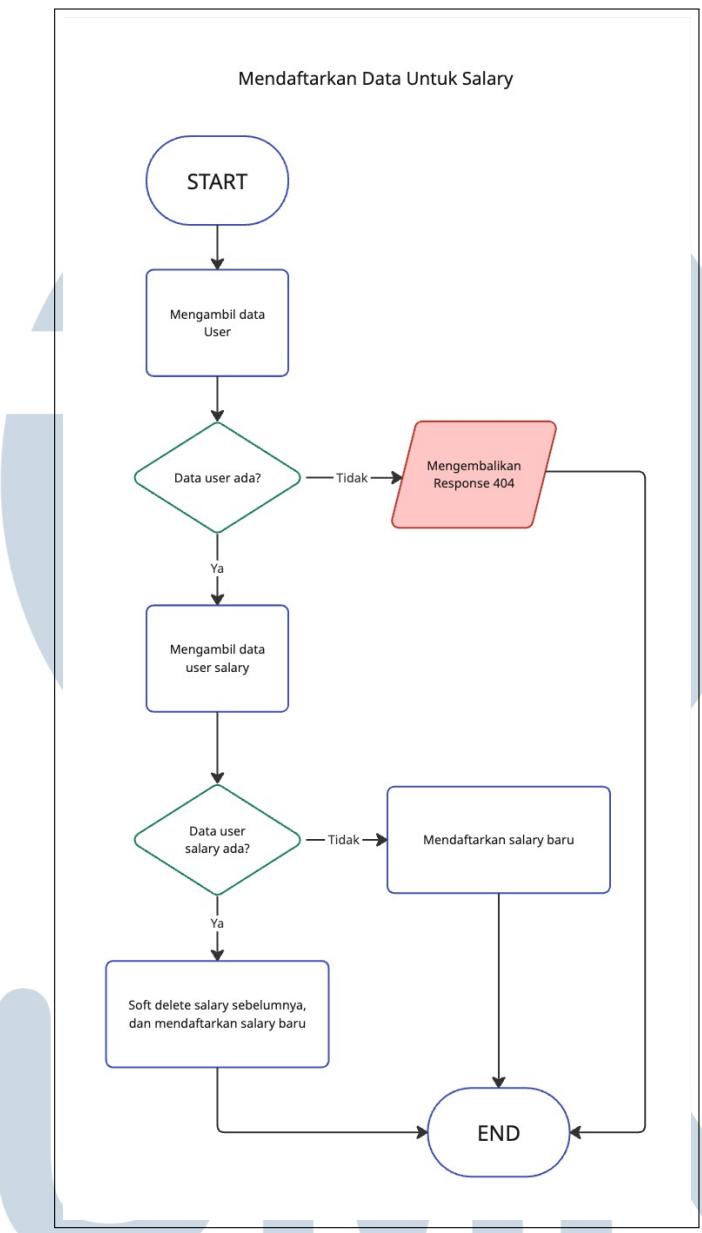
Gambar 3.13 menunjukkan proses untuk validasi *request* dan *parameter* akan dilanjuti dengan proses menambahkan data untuk *payroll information* secara *general*.

**UNIVERSITAS  
MULTIMEDIA  
NUSANTARA**



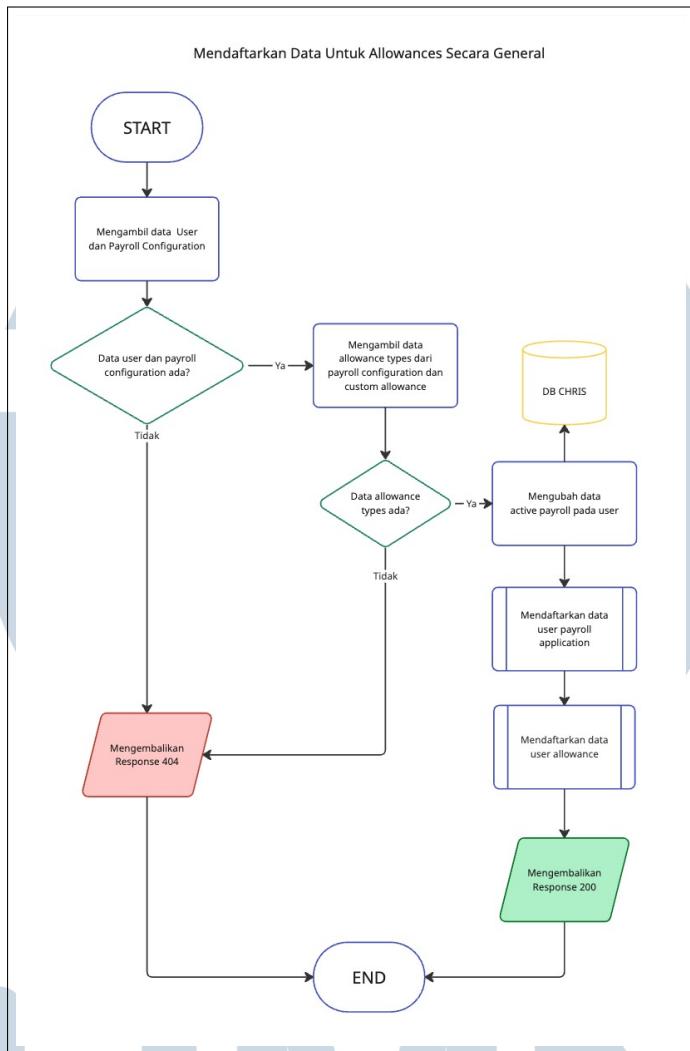
Gambar 3.14. Flowchart menambahkan data untuk payroll information

Pada *user management*, terdapat suatu field bernama *Payroll Information* yang berisi data gaji pegawai, dan field untuk memilih *Payroll Configuration* yang telah dibuat sebelumnya. Setelah memilih *Payroll Configuration*, *Superadmin* dapat mengisi data gaji pokok, tunjangan, dan potongan yang berlaku untuk pegawai tersebut. Gambar 3.13 menunjukkan alur sistem untuk pembuatan data gaji pegawai.



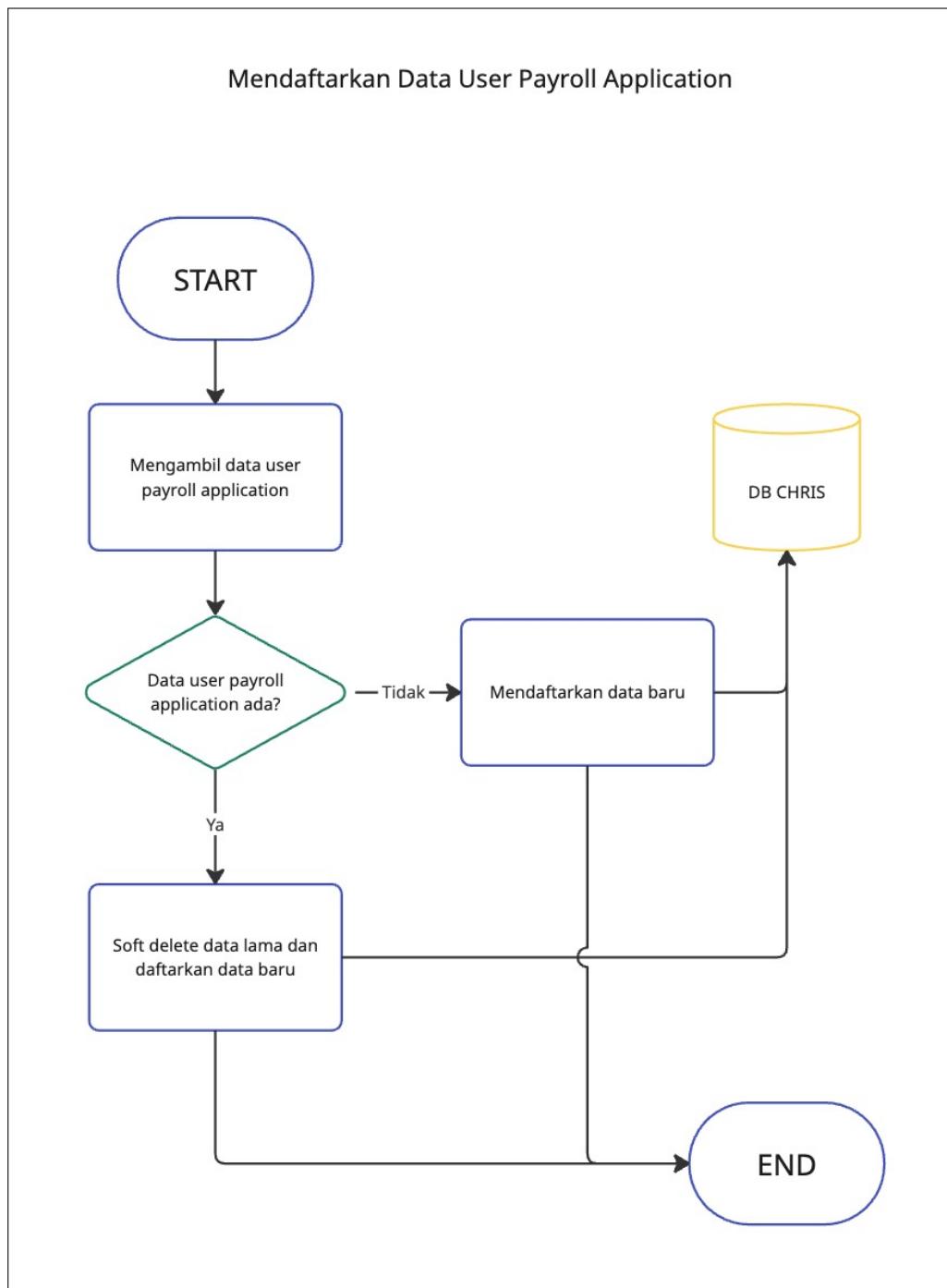
Gambar 3.15. Flowchart mendaftarkan data untuk salary

Gambar 3.15 menunjukkan proses pembuatan data gaji pegawai yang dimulai dengan mencari data pegawai dan mengambil data gaji pegawai tersebut. Setelah itu, jika data tersebut ditemukan, sistem akan menghapus data gaji pegawai yang lama secara *soft delete* dan membuat data gaji pegawai yang baru.



Gambar 3.16. *Flowchart* mendaftarkan data user allowances secara *general*

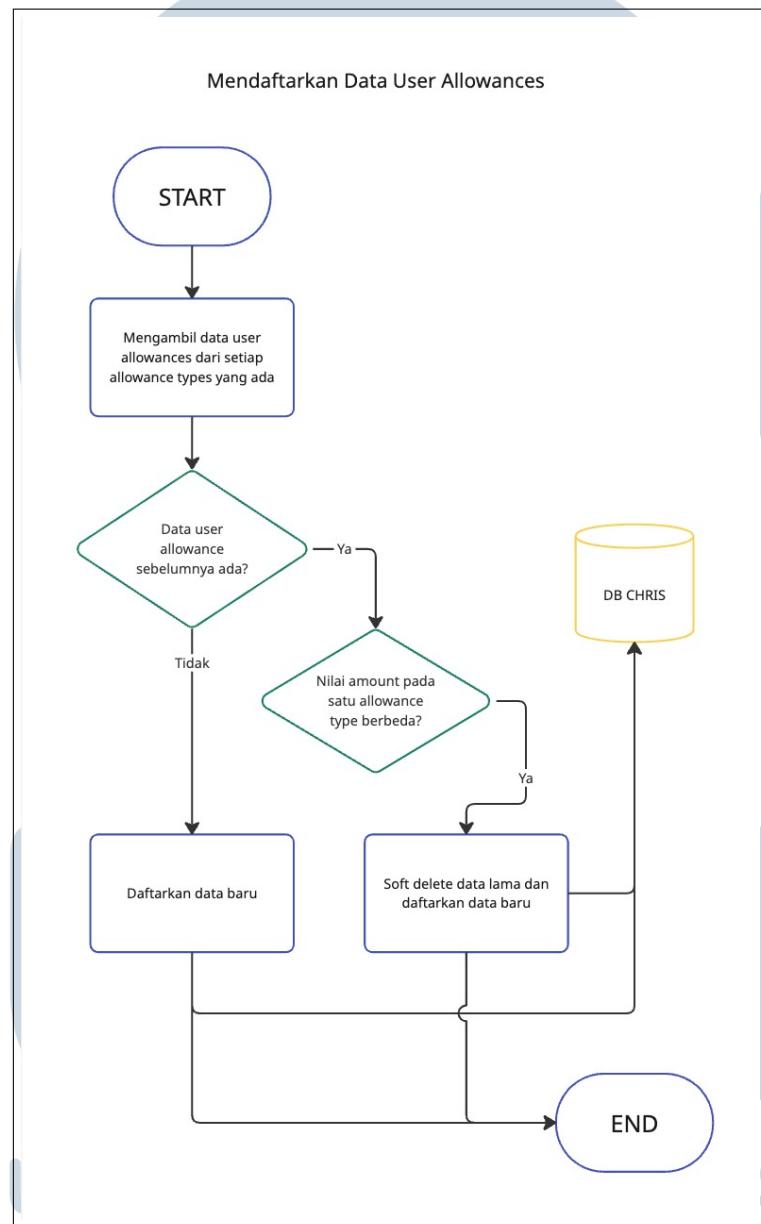
Gambar 3.16 menggambarkan alur proses pendaftaran data tunjangan pegawai secara umum. Proses ini diawali dengan pengambilan data pegawai dan *Payroll Configuration* yang telah tersedia. Selanjutnya, sistem mengekstraksi daftar tunjangan yang tercantum dalam *Payroll Configuration* tersebut. Setelah itu, sistem akan memperbarui informasi *Active Payroll* pada data pegawai, dan mendaftarkan entri baru pada *User Payroll Application* guna menghubungkan pegawai dengan konfigurasi payroll yang aktif. Terakhir, sistem akan mendaftarkan data *User Allowances* beserta nominal masing-masing tunjangan dan potongan yang telah ditentukan sebelumnya.



Gambar 3.17. Flowchart mendaftarkan data *User Payroll Application*

Gambar 3.17 menggambarkan proses pendaftaran data *User Payroll Application* yang diawali dengan pencarian entri sebelumnya pada pegawai terkait. Jika tidak ditemukan, sistem akan langsung membuat entri baru pada tabel *User Payroll Application*. Namun, apabila entri sudah ada, sistem akan terlebih dahulu

melakukan *soft delete* terhadap data tersebut, kemudian membuat entri baru guna memastikan hanya satu konfigurasi aktif yang tercatat untuk setiap pegawai. Pendekatan ini menjaga integritas historis tanpa menghapus data secara permanen.



Gambar 3.18. Flowchart mendaftarkan data *user allowances*

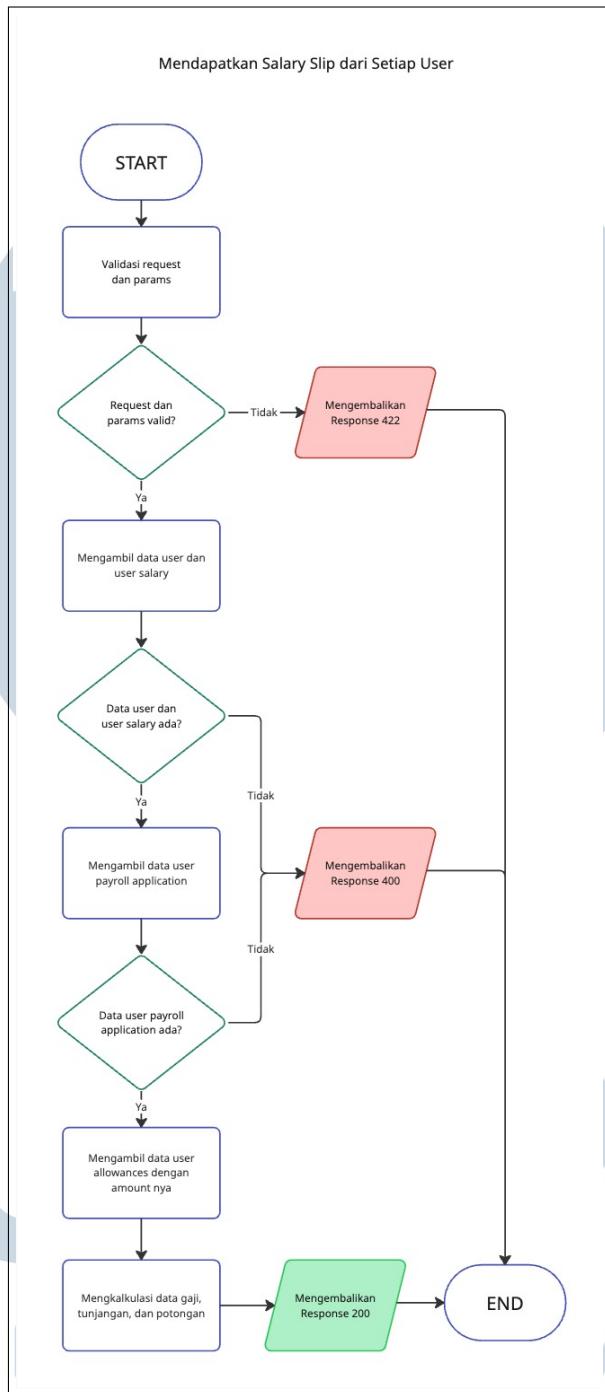
Gambar 3.18 menggambarkan proses pendaftaran data tunjangan pegawai yang diawali dengan pencarian data *User Allowances* berdasarkan setiap *Allowance Type* yang dimiliki pegawai. Apabila data tidak ditemukan, sistem akan membuat entri baru dengan mengisi *Allowance Type* beserta nominal tunjangan yang telah ditentukan. Namun, jika data ditemukan, sistem akan melakukan pengecekan

terhadap kesesuaian nominal tunjangan. Jika nominal yang ditemukan sama, maka tidak ada perubahan yang dilakukan. Sebaliknya, jika terdapat perbedaan, sistem akan memperbarui nilai nominal dengan yang baru, serta melakukan *soft delete* pada data lama. Pendekatan ini diterapkan untuk menjaga riwayat data dan memungkinkan pelacakan perubahan secara historis.

### B.3 Salary Slip

Setelah data gaji pegawai selesai dibuat, *Superadmin* dapat mengakses modul *Salary Slip* untuk melakukan finalisasi gaji. Proses finalisasi ini memungkinkan pengecekan akhir terhadap rincian gaji sebelum tanggal gajian. Di PT Ganda Visi Jayatama, proses penggajian dilakukan setiap tanggal 25, sehingga proses finalisasi disarankan dilakukan pada tanggal 24 setiap bulannya. Setelah tanggal 25, data tidak dapat lagi diubah.





Gambar 3.19. Flowchart mendapatkan salary slip dari setiap user

Gambar 3.19 menunjukkan proses mendapatkan slip gaji pegawai yang dimulai dengan mencari data pegawai berdasarkan *Payroll Configuration* yang telah dibuat sebelumnya. Setelah itu, sistem akan mengambil data gaji pegawai, tunjangan-tunjangan yang telah dibuatkan sebelumnya, dan membuat slip gaji

berdasarkan data tersebut.

Untuk pegawai yang telah memiliki data gaji terverifikasi, mereka dapat melihat slip gaji mereka masing-masing pada halaman *Salary Slip* dan mengunduhnya dalam format PDF. Hal ini memungkinkan pegawai untuk mengakses informasi gaji mereka secara transparan dan mudah.

Untuk mendapatkan slip gaji historis, sistem akan terlebih dahulu menentukan periode waktu berdasarkan tanggal yang diminta, yaitu awal hingga akhir bulan tersebut. Setelah itu, sistem akan mencari seluruh entri data gaji (*User Salary*) yang memiliki tanggal pembuatan (*created\_at*) sebelum atau sama dengan akhir periode, dan belum dihapus atau dihapus setelah periode tersebut berakhir. Seluruh data yang ditemukan kemudian diurutkan berdasarkan tanggal pembuatan dari yang terbaru ke yang terlama. Dari hasil pengurutan tersebut, sistem akan memilih satu entri data gaji paling terbaru yang masih berlaku pada periode tersebut. Entri tersebut kemudian digunakan untuk mengambil informasi tunjangan (*User Allowances*) dan pemotongan berdasarkan konfigurasi yang berlaku, lalu disusun menjadi slip gaji pegawai.

### 3.6 Implementasi Sistem

Implementasi sistem menghasilkan total 24 *endpoint API* yang mendukung fungsionalitas berbagai modul yang dikembangkan selama proyek. Setiap *endpoint* menggunakan salah satu dari lima metode HTTP standar, yaitu *GET*, *POST*, *PUT*, *DELETE*, dan *PATCH*. Metode *GET* digunakan untuk mengambil data dari server, *POST* untuk menambahkan data baru, *PUT* untuk mengganti keseluruhan data pada entitas tertentu, sedangkan *PATCH* memungkinkan pembaruan sebagian data tanpa harus mengirimkan seluruh payload. Sementara itu, metode *DELETE* digunakan untuk menghapus data secara permanen atau melakukan *soft delete* tergantung pada kebijakan implementasi sistem. Pentingnya penggunaan metode-metode ini dalam RESTful API telah dibahas secara mendalam dalam studi mengenai metodologi pengujian REST API [9], serta dokumentasi formal penambahan metode *PATCH* dalam protokol HTTP [10].

Setiap API memiliki standar struktur *request* dan *response* yang wajib diterapkan. Pada *request*, data identifikasi dikirim melalui parameter kueri (*query params*). Untuk *response*, format yang digunakan harus mencakup tiga elemen utama, yaitu *code* (kode status), *message* (pesan status), dan *data* (isi data). Contoh implementasi struktur *request* dan *response* dapat dilihat pada gambar yang

disediakan.

### 3.6.1 Implementasi *User Management* dan Validasi Data

#### A API Endpoints

Berikut adalah daftar *endpoint* yang telah dimodifikasi dan ditambahkan untuk modul *User Management*:

Tabel 3.2. *User Management API Endpoints*

No.	Endpoint	Method	Deskripsi Singkat
1	/auth/register	POST	Membuat entri data pegawai baru untuk menggunakan sistem CHRIS.
2	/users/update	PATCH	Mengubah data pegawai yang sudah ada sebelumnya.
3	/employee-statuses	GET	Mengambil daftar semua jenis kepegawaian ( <i>employment status</i> ).
4	/banks	GET	Mengambil daftar semua jenis bank .

Pada Tabel 3.2, ditampilkan sejumlah endpoint yang telah ditambahkan dan dimodifikasi untuk modul *User Management*. Setiap *endpoint* dilengkapi dengan metode HTTP yang sesuai dengan tujuannya: *POST* digunakan untuk pembuatan data baru, *PATCH* untuk pembaruan data, dan *GET* untuk pengambilan data dari basis data.

Secara khusus, *endpoint* seperti /users/register dan /users/update memiliki proses validasi input untuk memastikan data yang masuk telah memenuhi format dan ketentuan yang berlaku. *Field* yang divalidasi ulang meliputi:

- *fullname*: Nama lengkap pegawai, wajib diisi dan harus berupa karakter alfanumerik yang valid.
- *email*: Alamat *email* pegawai, Hanya menerima domain @concise.co.id dengan format yang sesuai menggunakan ekspresi reguler dan wajib diisi.

- *nik*: Nomor Induk Kependudukan pegawai, Wajib 16 digit angka.
- *contact* dan *emergency\_contact*: Nomor telepon dan nomor telepon darurat, wajib diisi dan hanya menerima nomor dengan awalan +62 atau 0.
- *bank\_holder*: Nama terdaftar pada bank, wajib diisi dan harus alfanumerik.
- *bank\_number*: Nomor rekening bank pegawai, wajib diisi dan hanya angka.

Validasi dilakukan untuk memastikan integritas data dan menghindari kesalahan yang dapat terjadi dalam pengolahan data pegawai, termasuk proses penggajian dan pelacakan informasi.

Sementara itu, beberapa *endpoint* seperti /employee-statuses dan /banks hanya berfungsi untuk mengambil data dari basis data dan tidak melalui proses validasi karena semata-mata digunakan untuk mengisi data pendukung pada *dropdown* dalam formulir pendaftaran atau pembaruan data pegawai.

## B Request dan Response

Berikut adalah struktur *request* dan *response* untuk *endpoint* /auth/register pada modul *User Management*:



```
1  {
2      "fullname": "Jose Andreas Lie",
3      "emergency_contact_holder": "Vannes Vincent Lie",
4      "bank_holder": "Jose Andreas Lie",
5      "employee_number": "18",
6      "email": "jose.andreas@concise.co.id",
7      "nik": "3659876543210987",
8      "contact": "85123456789",
9      "emergency_contact": "851234567898",
10     "password": "Passw0rd",
11     "confirmPassword": "Passw0rd",
12     "npwp": "-",
13     "bank_number": "6040928624",
14     "bank_branch": "Jakarta",
15     "gender": "male",
16     "job_title": "Backend Engineer",
17     "supervisor_id": "71f01488-5f38-4c1f-8d33-be176a03d8a1",
18     "employment_status_id": "a10cb53d-1dc9-4d9a-9d99-4665a2d1b7c2",
19     "bank_id": "4bf72b76-9635-4e52-a0f0-163d146fe47b",
20     "role_id": [
21         "5caa0a2c-a845-4049-96be-d3ce5014e57e"
22     ],
23     "roles": [
24         "Staff"
25     ],
26     "dob": "2004-08-22T00:00:00.000Z"
27 }
```

Gambar 3.20. Contoh struktur *request* untuk /auth/register



```
{  
    "code": 201,  
    "message": "The user has been successfully added.",  
    "data": {  
        "id": "87411100-4b84-11f0-991e-55a856be0234",  
        "is_pwd_resetted": false,  
        "fullname": "Jose Andreas Lie",  
        "emergency_contact_holder": "Vannes Vincent Lie",  
        "bank_holder": "Jose Andreas Lie",  
        "employee_number": "18",  
        "email": "jose.andreas@concise.co.id",  
        "nik": "3659876543210987",  
        "contact": "85123456789",  
        "emergency_contact": "851234567898",  
        "npwp": "-",  
        "bank_number": "6040928624",  
        "bank_branch": "Jakarta",  
        "gender": "male",  
        "job_title": "Backend Engineer",  
        "supervisor_id": "71f01488-5f38-4c1f-8d33-be176a03d8a1",  
        "employment_status_id": "a10cb53d-1dc9-4d9a-9d99-4665a2d1b7c2",  
        "bank_id": "4bf72b76-9635-4e52-a0f0-163d146fe47b",  
        "dob": "2004-08-22T00:00:00.000Z",  
        "is_active": true,  
        "updated_at": "2025-06-17T14:08:22.544Z",  
        "created_at": "2025-06-17T14:08:22.544Z",  
        "daily_status": null,  
        "bank_name": null,  
        "deleted_at": null,  
        "active_payroll_id": null  
    },  
    "tokens": {  
        "access": {  
            "token": "  
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI4NzQxMTEwMC00Yjg0LTExZjAtOTkxZS01NWE  
        "expires": "Wed Jun 18 2025 05:28:22 GMT+0700 (Western Indonesia Time)"  
        },  
        "refresh": {  
            "token": "  
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI4NzQxMTEwMC00Yjg0LTExZjAtOTkxZS01NWE  
        "expires": "Thu Jul 17 2025 21:08:22 GMT+0700 (Western Indonesia Time)"  
        }  
    }  
}
```

Gambar 3.21. Contoh struktur *response* untuk /auth/register

Berikut adalah struktur *request* dan *response* untuk *endpoint* /users/update pada modul *User Management*:

Path Params			
Name	Value	Type	
<input checked="" type="checkbox"/> user_id	87411100-4b84-11f0-991e-55a856be0234	string	 

Gambar 3.22. Contoh struktur *parameters* untuk /users/update

```
1  {
2    "fullname": "Jose Andreas Lie",
3    "job_title": "Backend Engineer",
4    "dob": "2004-08-22T00:00:00.000Z",
5    "role_id": [
6      "5caa0a2c-a845-4049-96be-d3ce5014e57e"
7    ],
8    "roles": [
9      "Staff"
10   ],
11   "nik": "3659876543210986",
12   "employee_number": "19",
13   "employment_status_id": "59be4135-9f4e-46a0-a041-d38d17caa951",
14   "contact": "85123456789",
15   "emergency_contact": "851234567898",
16   "emergency_contact_holder": "Vannes Vincent",
17   "supervisor_id": "71f01488-5f38-4c1f-8d33-be176a03d8a1",
18   "npwp": "-",
19   "bank_id": "4bf72b76-9635-4e52-a0f0-163d146fe47b",
20   "bank_number": "6040928624",
21   "bank_holder": "Jose Andreas Lie",
22   "bank_branch": "Jakarta",
23   "email": "jose.andreas@concise.co.id",
24   "password": "",
25   "confirmPassword": "",
26   "gender": "male",
27   "active_payroll_id": ""
28 }
```

Gambar 3.23. Contoh struktur *request* untuk /users/update

```
1  {
2    "code": 200,
3    "message": "User Information has been successfully updated."
4 }
```

Gambar 3.24. Contoh struktur *response* untuk /users/update

Pada gambar 3.20, terlihat struktur *request* yang dikirimkan ke *endpoint* /auth/register. Data yang dikirim mencakup informasi pegawai seperti nama lengkap, alamat email, nomor induk kependudukan (NIK), nomor telepon, kontak darurat, nama pemegang rekening bank, nomor rekening bank, dan masih banyak lagi. Setelah request berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.21, yang menunjukkan bahwa pendaftaran pegawai baru berhasil dengan kode status 201 (Created) dan pesan yang sesuai.

Pada gambar 3.22 dan 3.23, terlihat struktur *request* yang dikirimkan ke *endpoint* /users/update. Data yang dikirim sama seperti yang dikirim pada *request* pendaftaran, namun dengan tambahan *id* pegawai yang akan diperbarui pada bagian parameter. Setelah *request* berhasil diproses, sistem akan

mengembalikan respons seperti pada gambar 3.24, yang menunjukkan bahwa pembaruan data pegawai berhasil dengan kode status 200 (OK) dan pesan yang sesuai.

Berikut adalah struktur *response* untuk *endpoint* /employee-statuses pada modul *User Management*:

```
1  {
2      "code": 200,
3      "message": "List Employment Status Fetched Successfully",
4      "count": 4,
5      "data": [
6          {
7              "id": "a10cb53d-1dc9-4d9a-9d99-4665a2d1b7c2",
8              "employment_type": "Internship",
9              "created_at": "2025-06-13T08:19:32.008Z",
10             "updated_at": "2025-06-13T08:19:32.008Z",
11             "deleted_at": null
12         },
13         {
14             "id": "59be4135-9f4e-46a0-a041-d38d17caa951",
15             "employment_type": "Probation",
16             "created_at": "2025-06-13T08:19:32.008Z",
17             "updated_at": "2025-06-13T08:19:32.008Z",
18             "deleted_at": null
19         },
20         {
21             "id": "1b5254d2-10fa-4fcf-bd07-edf174098ed5",
22             "employment_type": "Part Time",
23             "created_at": "2025-06-13T08:19:32.008Z",
24             "updated_at": "2025-06-13T08:19:32.008Z",
25             "deleted_at": null
26         },
27         {
28             "id": "bf6be756-3be8-41e0-9dae-fff2824e340f",
29             "employment_type": "Full Time",
30             "created_at": "2025-06-13T08:19:32.008Z",
31             "updated_at": "2025-06-13T08:19:32.008Z",
32             "deleted_at": null
33         }
34     ]
35 }
```

Gambar 3.25. Contoh struktur *request* untuk /employee-statuses

Berikut adalah struktur *response* untuk *endpoint* /banks pada modul *User Management*:

```

1 ∵ {
2   "code": 200,
3   "message": "bank fetched successfully",
4   "data": [
5     {
6       "id": "4bf72b76-9635-4e52-a0f0-163d146fe47b",
7       "name": "Bank Central Asia",
8       "value": "bca",
9       "created_at": "2025-06-13T08:19:32.014Z",
10      "updated_at": "2025-06-13T08:19:32.014Z",
11      "deleted_at": null
12    },
13    {
14      "id": "99f5361a-d9e7-49c0-832e-e23c7485026d",
15      "name": "Bank Mandiri",
16      "value": "mandiri",
17      "created_at": "2025-06-13T08:19:32.014Z",
18      "updated_at": "2025-06-13T08:19:32.014Z",
19      "deleted_at": null
20    },
21    {
22      "id": "393b816e-91d0-478a-b436-1748db12a7d9",
23      "name": "BNI (Bank Negara Indonesia)",
24      "value": "bni",
25      "created_at": "2025-06-13T08:19:32.014Z",
26      "updated_at": "2025-06-13T08:19:32.014Z",
27      "deleted_at": null
28    },
29    {
30      "id": "05972fb1-16a9-4940-ac83-1a600c86baea",
31      "name": "Bank Rakyat Indonesia",
32      "value": "bri",
33      "created_at": "2025-06-13T08:19:32.014Z",
34      "updated_at": "2025-06-13T08:19:32.014Z",
35      "deleted_at": null
36    },
37    {
38      "id": "6dd9c6bd-10db-4730-89ea-55ee8aaa9415",
39      "name": "CIMB Niaga & CIMB Niaga Syariah",
40      "value": "cimb",
41      "created_at": "2025-06-13T08:19:32.014Z",
42      "updated_at": "2025-06-13T08:19:32.014Z",
43      "deleted_at": null

```

MULTIMEDIA  
NUSANTARA

### 3.6.2 Implementasi Fitur Autentikasi Biometrik pada CHRIS Mobile

#### A API Endpoints

Berikut adalah daftar *endpoint* yang telah ditambahkan untuk fitur autentikasi biometrik pada CHRIS Mobile:

Tabel 3.3. *Biometric Authentication API Endpoints*

No.	Endpoint	Method	Deskripsi Singkat
1	/auth/biometric/register	POST	Mendaftarkan data biometrik pegawai baru untuk autentikasi.
2	/auth/biometric/login	POST	Melakukan autentikasi pegawai menggunakan data biometrik yang telah didaftarkan.

Pada Tabel 3.3, ditampilkan daftar *endpoint* yang telah ditambahkan untuk fitur autentikasi biometrik pada CHRIS Mobile. Setiap *endpoint* dilengkapi dengan metode HTTP yang sesuai dengan tujuannya: *POST* digunakan untuk pendaftaran data biometrik pegawai baru dan autentikasi pegawai menggunakan data biometrik yang telah didaftarkan.

#### B Contoh Request dan Response

```
1  {
2      "biometric":  
3          "nbNmBuR8AWhgwgwwfKzg0bnR8P05KD3VJNbF8DHsN0hqu79M8ng7ce4VY35CSae6XEd0SPyEDGW6hP  
XRMnVc2Um1m3ES1JXLwb0wHJaZSCmTDGc12DRffFbpTwa4dCtXxQNdMbX1KHPgEpVg5cEuRwckTBXp9i2  
MCVLd95YDbf2ukd75aN6YqWrMSFt3quBiqhhjVFu9TzJn0egv7TSL4hkcEjuBvf iW10Y0Sne4wtaQx7A  
6A2qTKi91Smemj7tI
```

Gambar 3.27. Contoh struktur *request* untuk /auth/biometric/register

```
1  {
2    "code": 200,
3    "message": "Biometric data has been successfully added."
4 }
```

Gambar 3.28. Contoh struktur *response* untuk /auth/biometric/register

Pada gambar 3.27, terlihat struktur *request* yang dikirimkan ke *endpoint* /auth/biometric/register. Data yang dikirim berupa nilai *biometric* yang berisi string acak sepanjang 255 karakter yang dihasilkan oleh aplikasi CHRISM dan akan disimpan pada penyimpanan lokal perangkat. String ini berfungsi sebagai representasi dari data biometrik pengguna, bukan data biometrik aktual yang dikirim ke server. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.28, yang menunjukkan bahwa pendaftaran data biometrik pegawai berhasil dengan kode status 201 (*Created*) dan pesan yang sesuai.

```
1  {
2    "biometric":
3      "nbmUmUr8AWhghgvwfkZg0bnR8P05K03VJNbF80HSn0hqu7M8ng7ce4VY35Csae6XEd0SPyEDG6hPXNmNvC2Um1m3ES1JXLwb0wHJaZ5CmTDGc12DRffFbpTw
4      addCTxQNDnbXiKHPgEpVg5cEuRwcKTBp9i2MCVLd95Ydf2ukd75aN6YqMrMSFt3quBiqhhjVFu9tJn0egv7TSL4hkceJuubVfwiW10Y0Sne4wtaQx7A6A2qTK
5      i91SmemJ7t"
```

Gambar 3.29. Contoh struktur *request* untuk /auth/biometric/login



```

1   {
2     "code": 200,
3     "message": "Successfully Log In",
4     "data": {
5       "id": "87411100-4b84-11f0-991e-55a856be0234",
6       "fullname": "Jose Andreas Lie",
7       "is_active": true,
8       "is_pwd_resetted": false,
9       "supervisor_id": "71f01488-5f38-4c1f-8d33-be176a03d8a1",
10      "job_title": "Backend Engineer",
11      "dob": "2004-08-22T00:00:00.000Z",
12      "daily_status": null,
13      "contact": "85123456789",
14      "emergency_contact": "851234567898",
15      "employee_number": "19",
16      "email": "jose.andreas@concise.co.id",
17      "nik": "3659876543210986",
18      "npwp": "-",
19      "gender": "male",
20      "bank_id": "4bf72b76-9635-4e52-a0f0-163d146fe47b",
21      "bank_name": null,
22      "bank_number": "6040928624",
23      "bank_branch": "Jakarta",
24      "bank_holder": "Jose Andreas Lie",
25      "emergency_contact_holder": "Vannes Vincent",
26      "created_at": "2025-06-17T14:08:22.544Z",
27      "updated_at": "2025-06-17T16:00:42.233Z",
28      "deleted_at": null,
29      "active_payroll_id": null,
30      "supervisor": {
31        "id": "71f01488-5f38-4c1f-8d33-be176a03d8a1",
32        "fullname": "Hiroshi Tanaka"
33      },
34      "employment_status": {
35        "id": "59be4135-9f4e-46a0-a041-d38d17caa951",
36        "employment_type": "Probation"
37      },
38      "roles": {
39        "id": "5caa0a2c-a845-4049-96be-d3ce5014e57e",
40        "name": "Staff",
41        "level": 3,
42        "permissions": [
43          {

```

Gambar 3.30. Contoh struktur *response* untuk /auth/biometric/login

Pada gambar 3.29, terlihat struktur *request* yang dikirimkan ke *endpoint* /auth/biometric/login. Data yang dikirim hanya berupa nilai *biometric* yang berisi string acak sepanjang 255 karakter yang telah dibuat dan disimpan di storage lokal CHRISM saat proses registrasi biometrik sebelumnya. Nilai ini berfungsi sebagai representasi dari data biometrik pengguna. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.30, yang menunjukkan bahwa autentikasi pegawai menggunakan data biometrik berhasil dengan kode status 200 (*OK*), pesan yang sesuai dan data lainnya.

### 3.6.3 Implementasi *Leave Permit*

#### A API Endpoints

Berikut adalah daftar *endpoint* yang telah ditambahkan untuk modul *Leave Permit*:

Tabel 3.4. *Leave Permit API Endpoints*

No.	Endpoint	Method	Deskripsi Singkat
1	/leave-permit	GET	Menampilkan data leave permit pada user yang login.
2	/leave-permit/:id	GET	Mengambil data leave permit berdasarkan id.
3	/leave-permit	POST	Membuat data leave permit baru.
4	/leave-permit/:id	DELETE	Menghapus data leave permit berdasarkan id.
5	/leave-permit/dashboard	GET	Mengambil data leave permit yang telah diterima oleh atasan dan yang berjalan di minggu ini untuk dashboard page.
6	/leave-permit/all	GET	Mengambil data leave permit berdasarkan hirarki atasan.

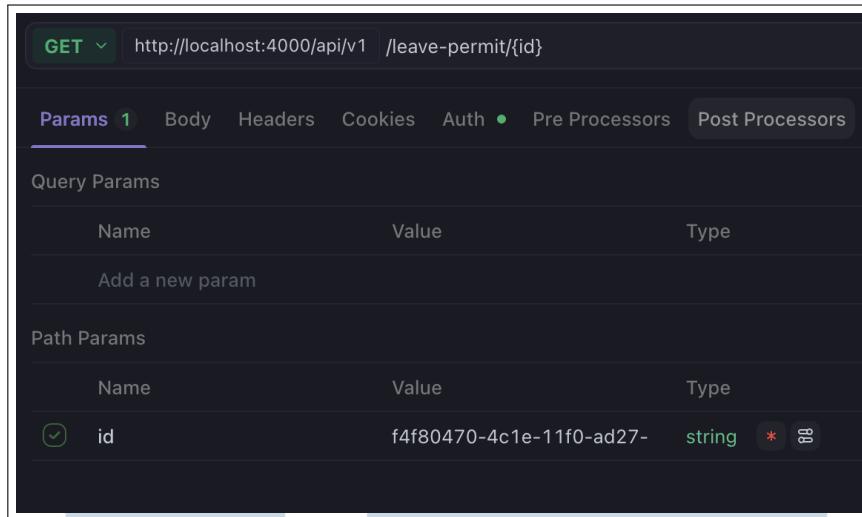
Pada Tabel 3.4, ditampilkan sebagian *endpoint* yang telah dimodifikasi dan ditambahkan untuk modul *Leave Permit*. Tabel ini tidak mencakup seluruh *endpoint* yang ada, melainkan hanya yang relevan dengan pengembangan selama masa kerja praktik. Setiap *endpoint* dilengkapi dengan metode HTTP yang sesuai dengan tujuannya: *GET* digunakan untuk mengambil data dari basis data, *POST* digunakan untuk pembuatan data baru, dan *DELETE* digunakan untuk menghapus data.

## B Contoh Request dan Response

```
1  {
2      "code": 200,
3      "message": "Success",
4      "data": {
5          "count": 2,
6          "rows": [
7              {
8                  "id": "f4f80470-4c1e-11f0-ad27-e34843b67a66",
9                  "status": "pending",
10                 "start_date": "2025-06-23T00:00:00.000Z",
11                 "end_date": "2025-06-23T00:00:00.000Z",
12                 "file_url": null,
13                 "reason": "Anak masuk rumah sakit",
14                 "leave_left": 29,
15                 "created_at": "2025-06-18T08:33:49.111Z",
16                 "updated_at": "2025-06-18T08:33:49.111Z",
17                 "oic": {
18                     "id": "e42e42e1-92ee-4125-9a43-82c84cd001ac",
19                     "fullname": "Kenji Nakamura"
20                 },
21                 "user": {
22                     "id": "71f01488-5f38-4c1f-8d33-be176a03d8a1",
23                     "fullname": "Hiroshi Tanaka"
24                 },
25                 "leave_type": {
26                     "id": "8e28c417-4d4a-4711-87e6-b352767ff046",
27                     "name": "WFH Permit",
28                     "duration": 30,
29                     "gender": "",
30                     "interval": "yearly"
31                 },
32                 "reviewed_by": null
33             },
34             {
35                 "id": "daf25b50-4aa9-11f0-ba5a-ddf1d091e5e5",
36                 "status": "accepted",
37                 "start_date": "2025-06-18T00:00:00.000Z",
38                 "end_date": "2025-06-18T00:00:00.000Z",
39             }
40         ]
41     }
42 }
```

Gambar 3.31. Contoh struktur *response* untuk GET /leave-permit

Pada gambar 3.35, terlihat struktur *response* yang dikembalikan oleh sistem ketika melakukan permintaan GET ke endpoint /leave-permit. Respons ini berisi daftar permohonan cuti yang telah dibuat oleh pegawai yang sedang login, dengan informasi seperti id, jenis cuti, tanggal mulai dan berakhir, status permohonan dan lain-lain.



Gambar 3.32. Contoh struktur *request* untuk GET /leave-permit/:id

```

1  {
2      "code": 200,
3      "message": "permit found",
4      "data": {
5          "id": "f4f80470-4c1e-11f0-ad27-e34843b67a66",
6          "leave_type_id": "8e28c417-4d4a-4711-87e6-b352767ff046",
7          "status": "pending",
8          "start_date": "2025-06-23T00:00:00.000Z",
9          "end_date": "2025-06-23T00:00:00.000Z",
10         "file_url": null,
11         "reason": "Anak masuk rumah sakit",
12         "leave_left": 29,
13         "oic": {
14             "id": "e42e42e1-92ee-4125-9a43-82c84cd001ac",
15             "fullname": "Kenji Nakamura"
16         },
17         "user": {
18             "id": "71f01488-5f38-4c1f-8d33-be176a03d8a1",
19             "fullname": "Hiroshi Tanaka",
20             "supervisor_id": null
21         },
22         "leave_type": {
23             "id": "8e28c417-4d4a-4711-87e6-b352767ff046",
24             "name": "WFH Permit"
25         },
26         "reviewed_by": null,
27         "file_name": ""
28     }
29 }
```

Gambar 3.33. Contoh struktur *response* untuk GET /leave-permit/:id

Pada gambar 3.32, terlihat struktur *request* yang dikirimkan ke *endpoint* /leave-permit/:id. Data yang dikirim hanya berupa *id* permohonan cuti

yang ingin diambil datanya. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.37, yang menunjukkan bahwa pengambilan data leave permit berdasarkan id berhasil dengan kode status 200 (*OK*) dan pesan yang sesuai.

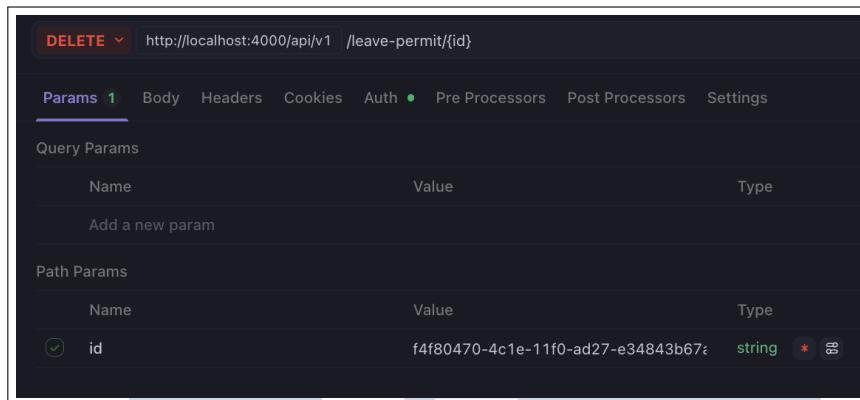
Name	Type	Description
start_date	string	*
end_date	string	*
file	file	*
reason	string	*
leave_type_id	string	*

Gambar 3.34. Contoh struktur *request* untuk POST /leave-permit

```
{  
    "code": 200,  
    "message": "request permit successfully"  
}
```

Gambar 3.35. Contoh struktur *response* untuk POST /leave-permit

Pada gambar 3.36, terlihat struktur *request* yang dikirimkan ke *endpoint* /leave-permit. Data yang dikirim berupa informasi terkait permohonan cuti, seperti jenis cuti, tanggal mulai dan berakhir, serta alasan permohonan. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.35, yang menunjukkan bahwa pembuatan data leave permit berhasil dengan kode status 201 (*Created*) dan pesan yang sesuai.



Gambar 3.36. Contoh struktur *request* untuk DELETE /leave-permit

```
1  {
2      "code": 200,
3      "message": "permit deleted successfully"
4 }
```

Gambar 3.37. Contoh struktur *response* untuk DELETE /leave-permit/:id

Pada gambar ??, terlihat struktur *request* yang dikirimkan ke *endpoint* /leave-permit/id. Data yang dikirim hanya berupa *id* permohonan cuti yang ingin dihapus. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.37, yang menunjukkan bahwa penghapusan data leave permit berdasarkan id berhasil dengan kode status 200 (*OK*) dan pesan yang sesuai.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

```
1  {
2      "code": 200,
3      "message": "Success",
4      "data": {
5          "count": 2,
6          "rows": [
7              {
8                  "id": "daf25b50-4aa9-11f0-ba5a-ddf1d091e5e5",
9                  "status": "accepted",
10                 "start_date": "2025-06-18T00:00:00.000Z",
11                 "end_date": "2025-06-18T00:00:00.000Z",
12                 "oic": {
13                     "id": "ad3de583-95de-46b0-9605-98ad0501be1f",
14                     "fullname": "Sakura Yamamoto"
15                 },
16                 "user": {
17                     "id": "71f01488-5f38-4c1f-8d33-be176a03d8a1",
18                     "fullname": "Hiroshi Tanaka"
19                 },
20                 "leave_type": {
21                     "id": "9f60890b-e0f5-4da9-9b2d-c84664c9adc3",
22                     "name": "Sick"
23                 },
24                 "reviewed_by": {
25                     "id": "6cf3b9ac-27ce-4514-b88e-bd4bd5b00769",
26                     "fullname": "Superadmin"
27                 }
28             },
29             {
30                 "id": "c10482b0-4c23-11f0-8661-07da1cc1c872",
31                 "status": "accepted",
32                 "start_date": "2025-06-20T00:00:00.000Z",
33                 "end_date": "2025-06-20T00:00:00.000Z"
34             }
35         ]
36     }
37 }
```

Gambar 3.38. Contoh struktur *response* untuk GET /leave-permit/dashboard

Pada gambar 3.38, terlihat struktur *response* yang dikembalikan oleh sistem ketika melakukan permintaan GET ke *endpoint* /leave-permit/dashboard. Respons ini berisi data leave permit yang telah diterima oleh atasan dan yang sedang berjalan di minggu ini, dengan informasi seperti id, jenis cuti, tanggal mulai dan berakhir, status permohonan, pegawai yang mengajukan cuti, pegawai yang menerima cuti, dan *officer in charge* yang dipilih. Data ini digunakan untuk menampilkan informasi terkait permohonan cuti pada halaman dashboard.

```
1  {
2      "code": 200,
3      "message": "Success",
4      "data": {
5          "count": 1,
6          "rows": [
7              {
8                  "id": "84adc640-4c24-11f0-8661-07da1cc1c872",
9                  "status": "pending",
10                 "start_date": "2025-06-19T00:00:00.000Z",
11                 "end_date": "2025-06-20T00:00:00.000Z",
12                 "file_url": null,
13                 "reason": "Keperluan di Bank",
14                 "leave_left": 28,
15                 "created_at": "2025-06-18T09:13:37.700Z",
16                 "updated_at": "2025-06-18T09:13:37.700Z",
17                 "oic": {
18                     "id": "71f01488-5f38-4c1f-8d33-be176a03d8a1",
19                     "fullname": "Hiroshi Tanaka"
20                 },
21                 "user": {
22                     "id": "ad3de583-95de-46b0-9605-98ad0501be1f",
23                     "fullname": "Sakura Yamamoto",
24                     "supervisor_id": "71f01488-5f38-4c1f-8d33-be176a03d8a1"
25                 },
26                 "leave_type": {
27                     "id": "8e28c417-4d4a-4711-87e6-b352767ff046",
28                     "name": "WFH Permit",
29                     "duration": 30,
30                     "gender": "",
31                     "interval": "yearly"
32                 },
33                 "reviewed_by": null
34             }
35         ]
36     }
37 }
```

Gambar 3.39. Contoh struktur *response* untuk GET /leave-permit/all

Pada gambar 3.39, terlihat struktur *response* yang dikembalikan oleh sistem ketika melakukan permintaan GET ke *endpoint* /leave-permit/all. Respons ini berisi data leave permit berdasarkan hirarki atasan, dengan informasi seperti id, jenis cuti, tanggal mulai dan berakhir, status permohonan, pegawai yang mengajukan cuti, pegawai yang menerima cuti, dan *officer in charge* yang dipilih. Data ini digunakan untuk menampilkan informasi terkait permohonan cuti pada halaman yang dapat diakses oleh atasan.

#### 3.6.4 Implementasi Payroll

##### A API Endpoints

Berikut adalah daftar *endpoint* yang telah ditambahkan untuk modul *Payroll*:

Tabel 3.5. *Payroll API Endpoints*

No.	Endpoint	Method	Deskripsi Singkat
1	/payroll-configuration/all	GET	Mengambil daftar data konfigurasi penggajian yang telah dibuat.
2	/payroll-configuration/:id	GET	Mengambil data konfigurasi penggajian berdasarkan id.
3	/payroll-configuration	POST	Membuat data konfigurasi penggajian baru.
4	/payroll-configuration/:id	DELETE	Menghapus data konfigurasi penggajian berdasarkan id.
5	/payroll-configuration/:id	PATCH	Mengubah data konfigurasi penggajian berdasarkan id.
6	/users/:user_id/payroll-configuration	GET	Mengambil daftar data konfigurasi penggajian berdasarkan id pegawai.
7	/users/:user_id/salary	GET	Mengambil data gaji pegawai berdasarkan id pegawai.
8	/user-allowance/:user_id/:payroll_configuration_id	GET	Mengambil data tunjangan berdasarkan id pegawai dan id konfigurasi penggajian.
9	/users/:user_id/with-details	PATCH	Menyimpan data pegawai, gaji, dan semua tunjangan yang sudah dikirimkan.
10	/salary-slip/all?date=YYYY-MM	GET	Mengambil daftar slip gaji pegawai yang telah dibuat.
Lanjut di halaman berikutnya			

Tabel 3.5 *Payroll API Endpoints* (lanjutan)

No.	Endpoint	Method	Deskripsi Singkat
11	/salary-slip/:id	GET	Mengambil data slip gaji pegawai berdasarkan id.
12	/salary-slip?date=YYYY-MM	GET	Mengambil data slip gaji pengguna yang sedang aktif pada bulan tertentu.

Pada Tabel 3.5, ditampilkan daftar *endpoint* yang telah ditambahkan untuk modul *Payroll*. Setiap *endpoint* dilengkapi dengan metode HTTP yang sesuai dengan tujuannya: *GET* digunakan untuk mengambil data dari basis data, *POST* digunakan untuk pembuatan data baru, *DELETE* digunakan untuk menghapus data, dan *PATCH* digunakan untuk memperbarui data yang ada.



## B Contoh Request dan Response

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": {  
5     "count": 5,  
6     "rows": [  
7       {  
8         "id": "90dafc51-7b33-4351-be7e-055cb59b88ed",  
9         "name": "Senior Part Time",  
10        "created_at": "2025-06-13T08:21:46.620Z",  
11        "updated_at": "2025-06-13T08:21:46.620Z",  
12        "employment_status": {  
13          "id": "1b5254d2-10fa-4fcf-bd07-edf174098ed5",  
14          "employment_type": "Part Time"  
15        }  
16      },  
17      {  
18        "id": "6027a18c-fd82-4b6a-9ecc-6a337f92e21a",  
19        "name": "Newbie",  
20        "created_at": "2025-06-13T08:21:58.631Z",  
21        "updated_at": "2025-06-13T08:21:58.631Z",  
22        "employment_status": {  
23          "id": "a10cb53d-1dc9-4d9a-9d99-4665a2d1b7c2",  
24          "employment_type": "Internship"  
25        }  
26      },  
27      {  
28        "id": "588eb642-4d3b-4040-8a27-fc37b39c16da",  
29        "name": "Fulltime Senior",  
30        "created_at": "2025-06-13T08:21:27.633Z",  
31        "updated_at": "2025-06-13T14:37:14.675Z",  
32        "employment_status": {  
33          "id": "bf6be756-3be8-41e0-9dae-fff2824e340f",  
34          "employment_type": "Full Time"  
35        }  
36      },  
37      {  
38        "id": "1f063e94-c12d-4d7a-9b58-b9bb465f5e0a",  
39        "name": "Newbie",  
40      }  
41    }  
42  }  
43 }
```

Gambar 3.40. Contoh struktur *request* untuk GET /payroll-configuration/all

Pada gambar 3.40, terlihat struktur *response* yang dikembalikan oleh sistem ketika melakukan permintaan GET ke endpoint /payroll-configuration/all. Respons ini berisi daftar konfigurasi penggajian yang telah dibuat, dengan informasi seperti id, nama konfigurasi, dan status kepegawaian.

Path Params				
Name	Value	Type	Description	
payroll_configuration_id	90dafc51-7b33-4351-be7e-055cb59b8	string	*	

Gambar 3.41. Contoh struktur *request* untuk GET /payroll-configuration/:id

```

1  {
2      "code": 200,
3      "message": "Successfully Fetched A Single Payroll",
4      "data": {
5          "id": "90dafc51-7b33-4351-be7e-055cb59b88ed",
6          "name": "Senior Part Time",
7          "employment_status": {
8              "id": "1b5254d2-10fa-4fcd-bd07-edf174098ed5",
9              "employment_type": "Part Time"
10         },
11         "allowance_types": [
12             {
13                 "id": "870461ca-ec97-4786-96fc-09124a0cc811",
14                 "name": "THR"
15             },
16             {
17                 "id": "31a4ea9d-2142-43ad-b38b-162ce98828de",
18                 "name": "Tunjangan Kendaraan"
19             }
20         ]
21     }
22 }
```

Gambar 3.42. Contoh struktur *response* untuk GET /payroll-configuration/:id

Pada gambar 3.41, terlihat struktur *request* yang dikirimkan ke *endpoint* /payroll-configuration/id. Data yang dikirim hanya berupa *id* konfigurasi penggajian yang ingin diambil datanya. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.42, yang menunjukkan bahwa pengambilan data konfigurasi penggajian berdasarkan *id* berhasil dengan kode status 200 (*OK*) dan pesan yang sesuai.

MULTIMEDIA  
NUSANTARA

```

1  {
2    "employment_status_id": "a10cb53d-1dc9-4d9a-9d99-4665a2d1b7c2",
3    "name": "Intern Backend Engineer",
4    "allowance_types": [
5      "THR",
6      "Uang makan"
7    ]
8 }

```

Gambar 3.43. Contoh struktur *request* untuk POST /payroll-configuration

```

1  {
2    "code": 201,
3    "message": "Successfully Created A Single Payroll",
4    "data": {
5      "id": "f393f86c-1d8a-4d5e-9d80-e9bc05ed000e",
6      "name": "Intern Backend Engineer",
7      "employment_status_id": "a10cb53d-1dc9-4d9a-9d99-4665a2d1b7c2",
8      "allowance_types": [
9        {
10          "id": "60acf1f2f-668e-42c6-96ed-e4f9a7b7135c",
11          "name": "THR"
12        },
13        {
14          "id": "26a36e8d-7a47-4f92-be9f-b4df9ce42179",
15          "name": "Uang makan"
16        }
17      ]
18    }
}

```

Gambar 3.44. Contoh struktur *response* untuk POST /payroll-configuration

Pada gambar 3.43, terlihat struktur *request* yang dikirimkan ke *endpoint* /payroll-configuration. Data yang dikirim berupa informasi terkait konfigurasi penggajian, seperti nama konfigurasi, status kepegawaian, dan tunjangan-tunjangan. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.44, yang menunjukkan bahwa pembuatan data konfigurasi penggajian berhasil dengan kode status 201 (*Created*) dan pesan yang sesuai.

Path Params		
Name	Value	Type
payroll_configuration_id	f393f86c-1d8a-4d5e-9d80-e9bc05ed01	string

Gambar 3.45. Contoh struktur *request* untuk DELETE /payroll-configuration/:id

```
1  {
2    "code": 200,
3    "message": "Successfully Deleted A Single Payroll"
4 }
```

Gambar 3.46. Contoh struktur *response* untuk DELETE /payroll-configuration/:id

Pada gambar 3.45, terlihat struktur *request* yang dikirimkan ke *endpoint* /payroll-configuration/id. Data yang dikirim hanya berupa *id* konfigurasi penggajian yang ingin dihapus. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.46, yang menunjukkan bahwa penghapusan data konfigurasi penggajian berdasarkan *id* berhasil dengan kode status 200 (*OK*) dan pesan yang sesuai.

Path Params				
Name	Value	Type	Description	
id	2d9ae3c9-cd59-4d85-979c-3bce3936f	string	*	

Gambar 3.47. Contoh struktur *params* untuk PATCH /payroll-configuration/:id

```
1  {
2    "name": "Internship Frontend Engineer",
3    "employment_status_id": "a10cb53d-1dc9-4d9a-9d99-4665a2d1b7c2",
4    "allowance_types": [
5      "THR",
6      "Uang bensin"
7    ]
8 }
```

Gambar 3.48. Contoh struktur *request* untuk PATCH /payroll-configuration/:id

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

```

1  {
2    "code": 200,
3    "message": "Successfully Updated A Single Payroll",
4    "data": {
5      "id": "2d9ae3c9-cd59-4d85-979c-3bce3936f49b",
6      "name": "Internship Frontend Engineer",
7      "employment_status": {
8        "id": "a10cb53d-1dc9-4d9a-9d99-4665a2d1b7c2",
9        "employment_type": "Internship"
10     },
11     "allowance_types": [
12       "THR",
13       "Uang bensin"
14     ]
15   }
16 }
```

Gambar 3.49. Contoh struktur *response* untuk PATCH /payroll-configuration/:id

Pada gambar ??, terlihat parameter yang dikirimkan ke path *endpoint* /payroll-configuration/:id yang berisi ID konfigurasi penggajian yang ingin dimodifikasi. Sementara pada gambar 3.48, terlihat struktur *request body* yang berisi field-field yang ingin direvisi, seperti nama konfigurasi, status kepegawaian, dan tunjangan-tunjangan. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.49, yang menunjukkan bahwa pembaruan data konfigurasi penggajian berhasil dengan kode status 200 (*OK*) dan pesan yang sesuai.

Path Params			
Name	Value	Type	
user_id	71f01488-5f38-4c1f-8d33-be176a03d	string	*

Gambar 3.50. Contoh struktur *request* untuk GET /users/:user\_id/payroll-configuration

```

1  {
2    "code": 200,
3    "message": "Successfully Fetched All Users",
4    "data": {
5      "count": 1,
6      "rows": [
7        {
8          "id": "588eb642-4d3b-4040-8a27-fc37b39c16da",
9          "name": "Fulltime Senior"
10        }
11      ]
12    }
13  }

```

Gambar 3.51. Contoh struktur *response* untuk GET /users/:user\_id/payroll-configuration

Pada gambar 3.51, terlihat struktur *request* yang dikirimkan ke *endpoint* /users/:user\_id/payroll-configuration, yang berisi parameter id pegawai yang ingin diambil data konfigurasi penggajinya. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.50, yang menunjukkan daftar konfigurasi penggajian yang tersedia berdasarkan status kepegawaian pegawai tersebut, dengan kode status 200 (*OK*) dan pesan yang sesuai.

Path Params				
Name	Value	Type	Required	Description
id	71f01488-5f38-4c1f-8d33-be176a03d	string	*	

Gambar 3.52. Contoh struktur *request* untuk GET /users/:user\_id/salary

```

1  {
2    "code": 200,
3    "message": "User salary retrieved successfully",
4    "data": {
5      "amount": 6000000
6    }
7  }

```

Gambar 3.53. Contoh struktur *response* untuk GET /users/:user\_id/salary

Pada gambar 3.52, terlihat struktur *request* yang dikirimkan ke *endpoint* /users/:user\_id/salary, yang berisi parameter id pegawai yang ingin diambil data gajinya. Setelah *request* berhasil diproses, sistem akan mengembalikan respons

seperti pada gambar 3.53, yang menunjukkan data gaji pegawai berdasarkan id pegawai tersebut, dengan kode status 200 (*OK*) dan pesan yang sesuai.

GET http://localhost:4000/api/v1/user-allowance/:user\_id/:payroll\_configuration\_id

Params 2 Body Headers Cookies Auth Pre Processors Post Processors Settings

Query Params

Name	Value	Type	Description
Add a new param			

Path Params

Name	Value	Type	Description
user_id	7f01488-5f38-4c1f-8d33-be176a03d	string *	
payroll_configuration_id	588eb642-4d3b-4040-8a27-fc37b39c	string *	

Gambar 3.54. Contoh struktur *request* untuk GET /user-allowance/:user\_id/:payroll\_configuration\_id

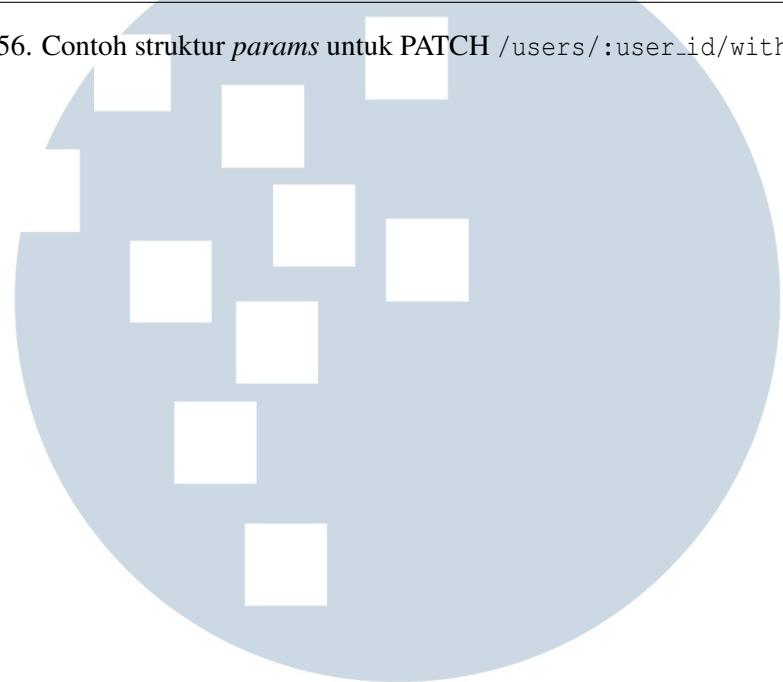
```
1 {
2   "code": 200,
3   "message": "Successfully Fetched All User Allowance",
4   "data": [
5     "allowances": [
6       {
7         "allowance_type_id": "3d1eba2b-00ba-4238-9a3c-0a334cadd79",
8         "allowance_type_name": "THR",
9         "is_custom": false,
10        "is_deduction": false,
11        "amount": "2000000"
12      },
13      {
14        "allowance_type_id": "936f42c2-bd7d-4537-bb34-a5fc9eae7fc5",
15        "allowance_type_name": "Tunjangan Makan",
16        "is_custom": false,
17        "is_deduction": false,
18        "amount": "1000000"
19      }
20    ]
21  }
22 }
```

Gambar 3.55. Contoh struktur *response* untuk GET /user-allowance/:user\_id/:payroll\_configuration\_id

Pada gambar 3.54, terlihat struktur *request* yang dikirimkan ke *endpoint* /user-allowance/:user\_id/:payroll\_configuration\_id, yang berisi parameter id pegawai dan id konfigurasi penggajian yang ingin diambil data tunjangannya. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.55, yang menunjukkan data tunjangan pegawai berdasarkan id pegawai dan id konfigurasi penggajian tersebut, dengan kode status 200 (*OK*) dan pesan yang sesuai.

Path Params				
Name	Value	Type	Description	
user_id	71f01488-5f38-4c1f-8d33-be176a03d	string	*	

Gambar 3.56. Contoh struktur *params* untuk PATCH /users/:user\_id/with-details



UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

```

1   {
2     "userData": {
3       "id": "71f01488-5f38-4c1f-8d33-be176a03d8a1",
4       "fullname": "Hiroshi Tanaka",
5       "is_active": true,
6       "is_pwd_reseted": true,
7       "supervisor_id": null,
8       "job_title": "Project Manager",
9       "dob": "1985-11-11T00:00:00.000Z",
10      "employment_status_id": "bf6be756-3be8-41e0-9dae-fff2824e340f",
11      "daily_status": null,
12      "contact": "8012345678",
13      "emergency_contact": "8087654321",
14      "employee_number": "1",
15      "email": "hiroshi.tanaka@concise.co.id",
16      "nik": "1234567890123456",
17      "npwp": "123456789012345",
18      "gender": "male",
19      "bank_id": "4bf72b76-9635-4e52-a0f0-163d146fe47b",
20      "bank_name": null,
21      "bank_number": "1234567890",
22      "bank_branch": "Tokyo Central Branch",
23      "bank_holder": "Hiroshi Tanaka",
24      "emergency_contact_holder": "Taro Yamada",
25      "biometric": null,
26      "created_at": "2025-06-13T08:19:32.176Z",
27      "updated_at": "2025-06-15T13:58:25.109Z",
28      "deleted_at": null,
29      "active_payroll_id": "588eb642-4d3b-4040-8a27-fc37b39c16da",
30      "roles": [
31        {
32          "id": "5caa0a2c-a845-4049-96be-d3ce5014e57e",
33          "name": "Staff",
34          "level": 3
35        },
36        {
37          "id": "e32555e3-63b3-4d9c-be45-e23a631964a4",
38          "name": "Supervisor",
39          "level": 2
40        }
41      ],
42    },
43    "salary": "6000000",
44    "allowances": {
45      "payroll_configuration_id": "588eb642-4d3b-4040-8a27-fc37b39c16da",
46      "allowances": [
47        {
48          "allowance_type_id": "3d1eba2b-00ba-4238-9a3c-0a334cadd79",
49          "amount": "2000000"
50        },
51        {
52          "allowance_type_id": "936f42c2-bd7d-4537-bb34-a5fc9eae7fc5",
53          "amount": "1000000"
54        },
55        {
56          "name": "Tambahkan untuk Hiroshi",
57          "amount": "10000000"
58        },
59        {
60          "name": "Merusak kursi kantor",
61          "amount": "700000",
62          "is_deduction": true
63        }
64      ]
65    }
66  }

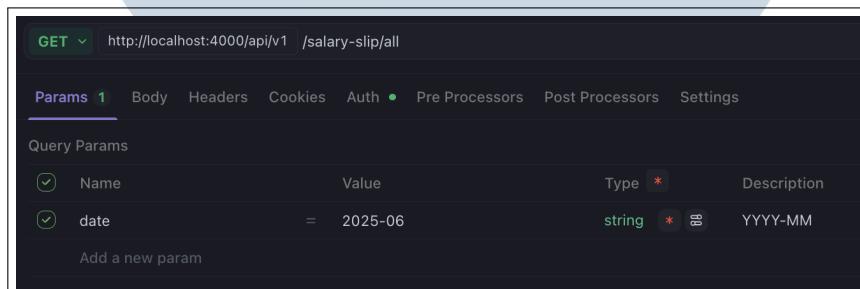
```

Gambar 3.57. Contoh struktur *request* untuk PATCH /users/:user\_id/with-details

```
1 ✓ {
2   "code": 200,
3   "message": "User updated successfully with all information"
4 }
```

Gambar 3.58. Contoh struktur *response* untuk PATCH /users/:user\_id/with-details

Pada gambar 3.56, terlihat parameter yang dikirimkan ke path *endpoint* /users/:user\_id/with-details yang berisi ID pegawai yang ingin dimodifikasi. Sementara pada gambar 3.57, terlihat struktur *request body* yang berisi field-field yang ingin direvisi, seperti data pegawai, gaji, dan tunjangan-tunjangan. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.58, yang menunjukkan bahwa pembaruan data pegawai, gaji, dan tunjangan berhasil dengan id pegawai berhasil dengan kode status 200 (*OK*) dan pesan yang sesuai.



Gambar 3.59. Contoh struktur *request* untuk GET /salary-slip/all

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

```
1  {
2    "code": 200,
3    "message": "Success",
4    "data": {
5      "count": 2,
6      "rows": [
7        {
8          "user": {
9            "id": "71f01488-5f38-4c1f-8d33-be176a03d8a1",
10           "fullname": "Hiroshi Tanaka",
11           "employment_status": {
12             "id": "bf6be756-3be8-41e0-9dae-fff2824e340f",
13             "employment_type": "Full Time"
14           }
15         }
16       },
17       {
18         "user": {
19           "id": "ad3de583-95de-46b0-9605-98ad0501be1f",
20           "fullname": "Sakura Yamamoto",
21           "employment_status": {
22             "id": "1b5254d2-10fa-4fcf-bd07-edf174098ed5",
23             "employment_type": "Part Time"
24           }
25         }
26       }
27     ]
28   }
29 }
```

Gambar 3.60. Contoh struktur *response* untuk GET /salary-slip/all

Pada gambar 3.59, terlihat struktur *request* yang dikirimkan ke *endpoint* /salary-slip/all, yang berisi query parameter date=YYYY-MM untuk menyaring slip gaji pada bulan dan tahun tertentu. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.60, yang menunjukkan daftar slip gaji pegawai yang telah dibuat untuk periode yang diminta, dengan kode status 200 (*OK*) dan pesan yang sesuai.

The screenshot shows a POSTMAN interface with a 'GET' request. The URL is `http://localhost:4000/api/v1/salary-slip/:user_id`. The 'Params' tab is selected, showing two query parameters: 'date' (value: '2025-06') and 'user\_id' (value: '71f01488-5f38-4c1f-8d33-be17'). The 'user\_id' parameter is highlighted with a red border around its row.

Gambar 3.61. Contoh struktur *request* untuk GET /salary-slip/:user\_id



```

1  {
2      "code": 200,
3      "message": "Salary slip generated successfully",
4      "data": {
5          "user_data": {
6              "id": "71f01488-5f38-4clf-8d33-be176a03d8a1",
7              "fullname": "Hirosi Tanaka",
8              "is_active": true,
9              "supervisor_id": null,
10             "job_title": "Project Manager",
11             "dob": "1985-11-11T00:00:00.000Z",
12             "employment_status_id": "bf6be756-3be8-4le0-9dae-fff2824e340f",
13             "daily_status": null,
14             "contact": "8012345678",
15             "emergency_contact": "8087654321",
16             "employee_number": "1",
17             "email": "hirosi.tanaka@concise.co.id",
18             "nik": "1234567890123456",
19             "npwp": "123456789012345",
20             "gender": "male",
21             "bank_id": "4bf72b76-9635-4e52-a0f0-163d146fe47b",
22             "bank_name": null,
23             "bank_number": "1234567890",
24             "bank_branch": "Tokyo Central Branch",
25             "bank_holder": "Hirosi Tanaka",
26             "emergency_contact_holder": "Taro Yamada",
27             "active_payroll_id": "588eb642-4d3b-4840-8a27-fc37b39c16da",
28             "employment_status": {
29                 "id": "bf6be756-3be8-4le0-9dae-fff2824e340f",
30                 "employment_type": "Full Time"
31             },
32             "bank": {
33                 "id": "4bf72b76-9635-4e52-a0f0-163d146fe47b",
34                 "name": "Bank Central Asia"
35             },
36             "salary": [
37                 {
38                     "amount": "6000000",
39                     "created_at": "2025-06-18T10:18:14.616Z"
40                 }
41             ],
42             "payroll_application": [
43                 {
44                     "id": "3b8a8cb7-fdcf-461a-b420-5936b7aee096",
45                     "payroll_configuration_id": "588eb642-4d3b-4840-8a27-fc37b39c16da"
46                 }
47             ]
48         },
49         "allowances": [
50             {
51                 "name": "THR",
52                 "amount": 2000000,
53                 "is_deduction": false
54             },
55             {
56                 "name": "Tunjangan Makan",
57                 "amount": 1000000,
58                 "is_deduction": false
59             },
60             {
61                 "name": "Tambahkan untuk Hirosi",
62                 "amount": 10000000,
63                 "is_deduction": false
64             },
65             {
66                 "name": "Merusak kursi kantor",
67                 "amount": 700000,
68                 "is_deduction": true
69             }
70         ],
71         "salary_data": {
72             "basic_salary": 6000000,
73             "total_allowances": 19000000,
74             "total_deductions": 700000,
75             "take_home_pay": 18300000
76         },
77         "period": {
78             "start_date": "2025-05-25T00:00:00.000Z",
79             "end_date": "2025-06-24T23:59:59.999Z"
80         },
81         "date": "June 2025"
82     }
83 }

```

Gambar 3.62. Contoh struktur *response* untuk GET /salary-slip/:user\_id

Pada gambar 3.61, terlihat struktur *request* yang dikirimkan ke *endpoint* /salary-slip/:id, yang berisi parameter id pegawai yang ingin diambil datanya. Selain itu, terdapat query parameter date=YYYY-MM yang digunakan untuk menyaring slip gaji pada bulan dan tahun tertentu. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.62, yang menunjukkan data slip gaji pegawai berdasarkan id pegawai tersebut pada periode waktu yang diminta, dengan kode status 200 (*OK*) dan pesan yang sesuai.

The screenshot shows a POSTMAN interface with the following details:

- Method:** GET
- URL:** http://localhost:4000/api/v1 /salary-slip
- Params:** 1
- Body:** Body
- Headers:** Headers
- Cookies:** Cookies
- Auth:** Auth
- Pre Processors:** Pre Processors
- Post Processors:** Post Processors
- Settings:** Settings

**Query Params:**

Name	Value	Type	Description
date	2025-06	string	*

Add a new param

Gambar 3.63. Contoh struktur *request* untuk GET /salary-slip



```

1  {
2      "code": 200,
3      "message": "Salary slip generated successfully",
4      "data": {
5          "user_data": {
6              "id": "71f01488-5f38-4clf-8d33-be176a03d8a1",
7              "fullname": "Hiroshi Tanaka",
8              "is_active": true,
9              "supervisor_id": null,
10             "job_title": "Project Manager",
11             "dob": "1985-11-11T00:00:00Z",
12             "employment_status_id": "bf6be756-3be8-4le0-9dae-fff2824e340f",
13             "daily_status": null,
14             "contact": "8012345678",
15             "emergency_contact": "8087654321",
16             "employee_number": "1",
17             "email": "hiroshi.tanaka@concise.co.id",
18             "nik": "1234567890123456",
19             "npwp": "123456789012345",
20             "gender": "male",
21             "bank_id": "4bf72b76-9635-4e52-a0f0-163d146fe47b",
22             "bank_name": null,
23             "bank_number": "1234567890",
24             "bank_branch": "Tokyo Central Branch",
25             "bank_holder": "Hiroshi Tanaka",
26             "emergency_contact_holder": "Taro Yamada",
27             "active_payroll_id": "588eb642-4d3b-4040-8a27-fc37b39c16da",
28             "employment_status": {
29                 "id": "bf6be756-3be8-4le0-9dae-fff2824e340f",
30                 "employment_type": "Full Time"
31             },
32             "bank": {
33                 "id": "4bf72b76-9635-4e52-a0f0-163d146fe47b",
34                 "name": "Bank Central Asia"
35             },
36             "salary": [
37                 {
38                     "amount": "6000000",
39                     "created_at": "2025-06-18T10:18:14.616Z"
40                 }
41             ],
42             "payroll_application": [
43                 {
44                     "id": "3b8a8cb7-fdcf-461a-b428-5936b7aee096",
45                     "payroll_configuration_id": "588eb642-4d3b-4040-8a27-fc37b39c16da"
46                 }
47             ]
48         },
49         "allowances": [
50             {
51                 "name": "THR",
52                 "amount": 2000000,
53                 "is_deduction": false
54             },
55             {
56                 "name": "Tunjangan Makan",
57                 "amount": 1000000,
58                 "is_deduction": false
59             },
60             {
61                 "name": "Tambahkan untuk Hiroshi",
62                 "amount": 1000000,
63                 "is_deduction": false
64             },
65             {
66                 "name": "Merusak kursi Kantor",
67                 "amount": 700000,
68                 "is_deduction": true
69             }
70         ],
71         "salary_data": {
72             "basic_salary": 6000000,
73             "total_allowances": 19000000,
74             "total_deductions": 700000,
75             "take_home_pay": 18300000
76         },
77         "period": {
78             "start_date": "2025-05-25T00:00:00Z",
79             "end_date": "2025-06-24T23:59:59.999Z"
80         },
81         "date": "June 2025"
82     }
83 }

```

Pada gambar 3.63, terlihat struktur *request* yang dikirimkan ke *endpoint* /salary-slip, yang berisi query parameter date=YYYY-MM untuk menyaring slip gaji pada bulan dan tahun tertentu. Setelah *request* berhasil diproses, sistem akan mengembalikan respons seperti pada gambar 3.64, yang menunjukkan data slip gaji pengguna yang sedang aktif pada periode waktu yang diminta, dengan kode status 200 (*OK*) dan pesan yang sesuai.

### 3.7 Kendala dan Solusi yang Ditemukan

#### 3.7.1 Kendala

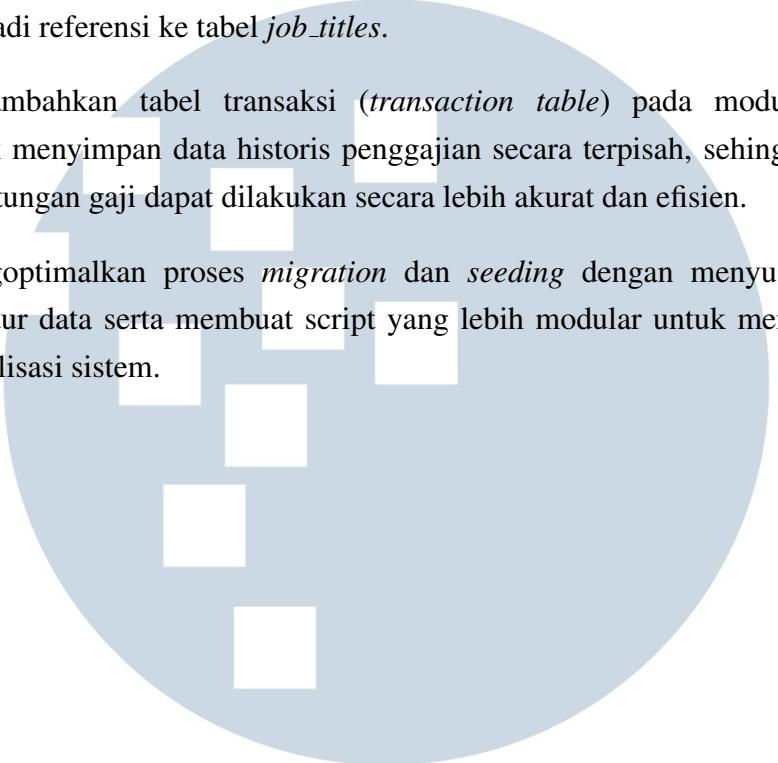
Selama pelaksanaan kerja praktik, ditemukan beberapa kendala teknis dan struktural dalam proses pengembangan sistem, yaitu sebagai berikut:

1. Beberapa bagian kode tidak mengikuti prinsip *best practice* dalam pengembangan backend, sehingga menyulitkan proses pengembangan lanjutan dan pemeliharaan sistem.
2. Terdapat penggunaan nilai yang masih ditulis secara *hardcoded*, seperti *job title*, yang seharusnya dikelola melalui relasi ke dalam basis data agar lebih fleksibel dan dapat disesuaikan.
3. Modul Payroll memiliki kompleksitas tinggi, khususnya dalam pengambilan data historis pengguna, yang menjadi tantangan dalam perhitungan gaji berdasarkan riwayat data.
4. Proses *migration* dan *seeding* data belum optimal dan membutuhkan waktu lama karena struktur data yang belum efisien.

#### 3.7.2 Solusi

Berikut adalah langkah-langkah solusi yang dilakukan untuk mengatasi kendala-kendala tersebut:

1. Melakukan proses *code review* secara berkala bersama tim *backend* untuk memastikan bahwa setiap kode yang ditulis mengikuti standar *best practice* dan dapat dikembangkan dengan mudah di masa depan.

- 
2. Melakukan *refactoring* pada bagian kode yang masih menggunakan pendekatan *hardcoded*, seperti pengelolaan *job title*, dengan menggantinya menjadi referensi ke tabel *job\_titles*.
  3. Menambahkan tabel transaksi (*transaction table*) pada modul Payroll untuk menyimpan data historis penggajian secara terpisah, sehingga proses perhitungan gaji dapat dilakukan secara lebih akurat dan efisien.
  4. Mengoptimalkan proses *migration* dan *seeding* dengan menyusun ulang struktur data serta membuat script yang lebih modular untuk mempercepat inisialisasi sistem.



## BAB 4

### SIMPULAN DAN SARAN

#### @todo

Tuliskan simpulan dan saran terkait dengan pekerjaan yang dilakukan.

#### 4.1 Simpulan

Selama pelaksanaan kerja praktik di PT Ganda Visi Jayatama, pengembangan lanjutan backend pada sistem informasi kepegawaian CHRIS telah berhasil diselesaikan dengan mencakup beberapa modul penting, seperti *User Management*, *Leave Permit*, Sistem Hierarki Supervisi, serta modul *Payroll*. Pengembangan dilakukan melalui penambahan fitur baru, refaktorisasi struktur data, validasi *form input*, serta optimalisasi logika sistem sesuai kebutuhan bisnis.

Pengembangan ini menghasilkan berbagai endpoint API yang telah diuji secara manual dan terintegrasi secara fungsional dengan sisi frontend. Salah satu pencapaian utama adalah penyempurnaan alur cuti berbasis sistem hierarki jabatan, penambahan fitur cancel leave permit, serta perhitungan gaji otomatis berdasarkan konfigurasi payroll dan data tunjangan. Selain itu, fitur otentifikasi biometrik juga telah berhasil diimplementasikan guna meningkatkan kecepatan akses pada aplikasi mobile CHRIS (CHRISM).

Secara keseluruhan, pengembangan ini berkontribusi dalam meningkatkan efisiensi proses internal, memperkuat keamanan sistem, dan memastikan transparansi serta keterlacakkan data kepegawaian secara lebih terstruktur dan terintegrasi.

#### 4.2 Saran

Ada beberapa saran yang dapat dipertimbangkan untuk pengembangan sistem CHRIS ke depannya:

- Otomasi Pengujian:** Disarankan untuk menambahkan unit testing dan integrasi testing otomatis pada setiap modul yang dikembangkan. Hal ini akan memastikan bahwa setiap perubahan kode dapat langsung tervalidasi, sehingga meminimalkan risiko bug yang tidak terdeteksi.

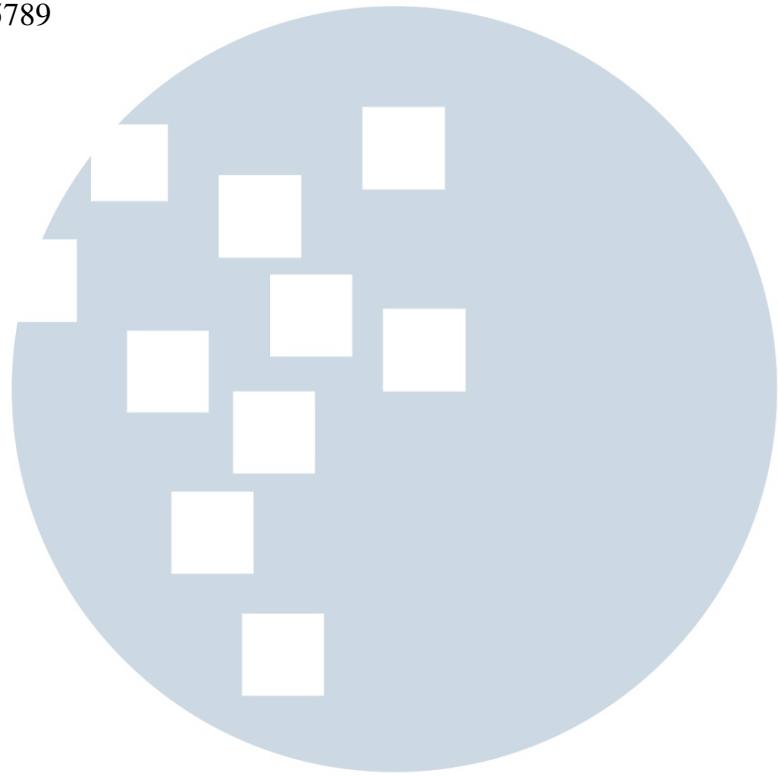
2. **Standarisasi Kode:** Perlu dilakukan standarisasi terhadap penulisan kode dan struktur direktori backend. Hal ini akan membantu menjaga konsistensi dalam pengembangan di masa mendatang, sehingga memudahkan pemeliharaan dan kolaborasi antar pengembang.
3. **Integrasi Notifikasi:** Modul seperti Leave Permit atau Payroll dapat dikembangkan lebih lanjut dengan integrasi sistem notifikasi real-time (misal: email atau push notification). Hal ini akan meningkatkan responsivitas pengguna terhadap perbaruan status, sehingga mereka dapat segera mengambil tindakan yang diperlukan.



## DAFTAR PUSTAKA

- [1] M. K. Normalini, N. M. Kassim, T. Ramayah, and S. Kurnia, “Antecedents and outcomes of human resource information system (hris) use,” *International Journal of Productivity and Performance Management*, vol. 61, no. 6, pp. 603–623, 2012. [Online]. Available: <https://doi.org/10.1108/17410401211249184>
- [2] E. H. H. Panjaitan, “Implementing human resource information system (hris) for efficient human resource management,” *International Journal of Science and Society*, vol. 5, no. 2, pp. 128–139, 2023. [Online]. Available: <https://doi.org/10.54783/ijsoc.v5i2.676>
- [3] S. Ahmed and M. Ali, “Web-based payroll management system: design, implementation, and evaluation,” *Journal of Electrical Systems and Information Technology*, vol. 10, no. 1, p. 15, 2023. [Online]. Available: <https://doi.org/10.1186/s43067-023-00082-5>
- [4] R. Ujjianto, G. F. Salsabilla, M. A. Dewi, and T. Darwis, “Human resource information system design pt fidac inovasi teknologi,” in *2024 9th International Conference on Business and Industrial Research (ICBIR)*, May 2024, pp. 1–5. [Online]. Available: <https://doi.org/10.1109/ICBIR61386.2024.10875679>
- [5] C. Evrendilek, I. H. Toroslu, and S. Hashemikhbir, “Task assignment in tree-like hierarchical structures,” *Journal of Combinatorial Optimization*, vol. 34, no. 2, pp. 631–655, 2017. [Online]. Available: <https://doi.org/10.1007/s10878-016-0097-6>
- [6] S. Jahan, “Human resources information system (hris): A theoretical perspective,” *Journal of Human Resource and Sustainability Studies*, vol. 2, no. 2, pp. 33–39, 2014. [Online]. Available: <https://www.scirp.org/journal/paperinformation.aspx?paperid=46129>
- [7] P. Wandhe, “A role of effectiveness of human resource information system (hris) in 21st century,” *International Journal of Management and Social Sciences Research*, vol. 9, no. 10, pp. 45–52, 2020. [Online]. Available: <https://doi.org/10.2139/ssrn.3718247>
- [8] F. Tekkry, “Rancang bangun frontend human resource information system di pt ganda visi jayatama,” [https://kc.umn.ac.id/id/eprint/32761/3/BAB\\_II.pdf](https://kc.umn.ac.id/id/eprint/32761/3/BAB_II.pdf), 2024, diakses pada April 2025.
- [9] A. Ehsan, M. A. M. E. Abuhalqa, C. Catal, and D. Mishra, “Restful api testing methodologies: Rationale, challenges, and solution directions,” *Applied Sciences*, vol. 12, no. 9, p. 4369, 2022. [Online]. Available: <https://doi.org/10.3390/app12094369>

- [10] L. Dusseault and J. Snell, “Patch method for http,” <https://doi.org/10.17487/RFC5789>, 2010, rFC 5789. [Online]. Available: <https://doi.org/10.17487/RFC5789>



UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## Lampiran 1. MBKM-01 Cover Letter MBKM Internship Track 1

**MBKM-01 Cover Letter MBKM Internship Track 1**

Tangerang, June 17th 2025

No : 1/UMN/TI/Internship Track 1/I/2025  
Subject : **Student's Application for MBKM Internship Track 1**

Dear. Head of Human Resource Department  
Concise (PT Ganda Visi Jayatama)

Universitas Multimedia Nusantara's providing the MBKM Internship Track 1, a work-integrated learning program, for students to hone their skills according to their talents and interests into the real work environments. Students directly doing Internship Track 1, in the company to learn solving problems based on knowledge that gained in campus, to link and match Internship Track 1 program with the curriculum as preparation for their future careers.

We pleased to inform the student with the following details:

Student ID	:	00000067097
Student Name	:	Jose Andreas Lie
Academic Program	:	Informatics
Email	:	jose.andreas1@student.umn.ac.id
Mobile Phone	:	6285156119138

Company will be received the student as an employee and Internship Track 1 participant, he/she express their willingness to follow 640 working hours or 100 working days prior to work rules 8 hours per day. Therefore, UMN's student must obey all regulations stipulated by company from time to time.

Along with respect, we considered our student to get selected in the Internship Track 1 program from your company. We thank you and look forward to hear employment acceptance letter of our student's.

Sincerely,

**Head of Departement Informatics Program  
Multimedia Nusantara University**



( Arya Wicaksana, S.Kom., M.Eng.Sc., OCA )

**U  
M  
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**

Kampus UMN, Scientia Garden | jl. Boulevard Gading Serpong - Tangerang | P. +62 21 5422 0808 | F. +62 21 5422 0800 | [www.umn.ac.id](http://www.umn.ac.id)

## Lampiran 2. MBKM-02 MBKM Internship Track 1 Card

MBKM-02 MBKM Internship Track 1 Card																					
<b>MBKM INTERNSHIP TRACK 1 CARD</b>																					
																					
<table border="1"><tr><td>Name : Jose Andreas Lie</td><td>Student ID : 00000067097</td></tr><tr><td>Address : Jl. Permata Raya No.1, RT.005/RW.002, Poris Plawad Utara, Kec. Cipondoh, Kota Tangerang, Banten 15141</td><td>Mobile Phone : 6285156119138</td></tr><tr><td>MBKM Internship Track 1 Acceptance Letter No: 018/HRD/OL/XII/2024</td><td>Letter Date : 31-12-2024</td></tr><tr><td colspan="2">Advisor's Name : Dr. Ivransa Zuhdi Pane, B.Eng., M.Eng.</td></tr><tr><td colspan="2">Company Name : Concise (PT Ganda Visi Jayatama)</td></tr><tr><td>Company Address : Ruko Crystal 1 No. 19, Tangerang, Banten 15810</td><td>City : Tangerang Postal Code: 15810</td></tr><tr><td>Company Website : www.concise.co.id</td><td>Company Phone : -</td></tr><tr><td>Supervisor's Name : Edo Setiawan Supervisor's Phone : +62 878 1823 9408 Supervisor's Email : edo.setiawan@concise.co.id</td><td>Supervisor's Position : Head Of Development Supervisor's Ext. : -</td></tr><tr><td>Department : IT Consultant</td><td>Position : Backend Engineer</td></tr><tr><td colspan="2">Acceptance Date : 13-01-2025</td></tr></table>		Name : Jose Andreas Lie	Student ID : 00000067097	Address : Jl. Permata Raya No.1, RT.005/RW.002, Poris Plawad Utara, Kec. Cipondoh, Kota Tangerang, Banten 15141	Mobile Phone : 6285156119138	MBKM Internship Track 1 Acceptance Letter No: 018/HRD/OL/XII/2024	Letter Date : 31-12-2024	Advisor's Name : Dr. Ivransa Zuhdi Pane, B.Eng., M.Eng.		Company Name : Concise (PT Ganda Visi Jayatama)		Company Address : Ruko Crystal 1 No. 19, Tangerang, Banten 15810	City : Tangerang Postal Code: 15810	Company Website : www.concise.co.id	Company Phone : -	Supervisor's Name : Edo Setiawan Supervisor's Phone : +62 878 1823 9408 Supervisor's Email : edo.setiawan@concise.co.id	Supervisor's Position : Head Of Development Supervisor's Ext. : -	Department : IT Consultant	Position : Backend Engineer	Acceptance Date : 13-01-2025	
Name : Jose Andreas Lie	Student ID : 00000067097																				
Address : Jl. Permata Raya No.1, RT.005/RW.002, Poris Plawad Utara, Kec. Cipondoh, Kota Tangerang, Banten 15141	Mobile Phone : 6285156119138																				
MBKM Internship Track 1 Acceptance Letter No: 018/HRD/OL/XII/2024	Letter Date : 31-12-2024																				
Advisor's Name : Dr. Ivransa Zuhdi Pane, B.Eng., M.Eng.																					
Company Name : Concise (PT Ganda Visi Jayatama)																					
Company Address : Ruko Crystal 1 No. 19, Tangerang, Banten 15810	City : Tangerang Postal Code: 15810																				
Company Website : www.concise.co.id	Company Phone : -																				
Supervisor's Name : Edo Setiawan Supervisor's Phone : +62 878 1823 9408 Supervisor's Email : edo.setiawan@concise.co.id	Supervisor's Position : Head Of Development Supervisor's Ext. : -																				
Department : IT Consultant	Position : Backend Engineer																				
Acceptance Date : 13-01-2025																					
<p>This MBKM Internship Track 1 Card has been completed with my real information and can be accounted for. I am ready to be disqualified if the data given are incorrect.</p>																					
Tangerang, June 18th 2025 Student's signature	  																				
Jose Andreas Lie	 Edo Setiawan																				
																					

### Lampiran 3. MBKM-03 Daily Task - Internship Track 1

MBKM-03 Daily Task - Internship Track 1					
 UNIVERSITAS MULTIMEDIA NUSANTARA					
<h2>Daily Task</h2>					
<b>STUDENT ID</b> : 00000067097					
<b>STUDENT NAME</b> : Jose Andreas Lie					
<b>COMPANY NAME</b> : Concise (PT Ganda Visi Jayatama)					
<b>No</b>	<b>Date</b>	<b>In</b>	<b>Out</b>	<b>Duties /Responsibilities</b>	<b>Supervisor's Sign</b>
1	13/01/2025	09:00	18:00	- On Boarding - Learning Back End Boilerplate	Approved at 08 April 2025 13:08
2	14/01/2025	08:00	19:00	- Brief Roadmap - Learning Back End Boilerplate - Trying CRUD API - Installations	Approved at 08 April 2025 13:08
3	15/01/2025	08:00	19:00	- Learning Back End Boilerplate - Starting Mini Project to Learn Back End	Approved at 08 April 2025 13:08
4	16/01/2025	08:00	19:00	- Learning Back End Boilerplate - Continue the Mini Project	Approved at 08 April 2025 13:08
5	17/01/2025	08:00	19:00	- Learning Back End Boilerplate - Sprint Retro 1 on Chris Project	Approved at 08 April 2025 13:08
6	18/01/2025	09:00	17:00	- Learning Back End Boilerplate - Creating API to Retrieve Employee Status List	Approved at 08 April 2025 13:08
7	20/01/2025	08:00	19:00	- Continue Create API to Retrieve Employee Status List	Approved at 08 April 2025 13:08
8	21/01/2025	08:00	20:00	- Revamp Employee Form Validation - Continue Create API to Retrieve Employee Status List	Approved at 08 April 2025 13:08
9	22/01/2025	08:00	20:00	[CHRIS] Create API to Retrieve Employee Status List [CHRIS] Create Employee Status Table	Approved at 08 April 2025 13:08

**Notes:**

1. Copied Form must be attached in report when registering for exam

In witness whereof the company,  
  
Edu Setiawan  
Head Of Development

Please sign along with the Company's stamp



## Daily Task

**STUDENT ID** : 00000067097

**STUDENT NAME** : Jose Andreas Lie

**COMPANY NAME** : Concise (PT Ganda Visi Jayatama)

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
10	23/01/2025	08:00	20:00	[CHRIS] Create API to Retrieve Employee Status List (cont.)	Approved at 08 April 2025 13:08
11	24/01/2025	08:00	20:00	[CHRIS] Finishing User and Employee Status Table [Internal] Backend Internal Meeting	Approved at 08 April 2025 13:09
12	25/01/2025	08:00	17:00	[CHRIS] Minor revises for API responses in the employee form (cont.)	Approved at 08 April 2025 13:09
13	30/01/2025	08:00	20:00	[CHRIS] Minor revises for API responses in the employee form (cont.)	Approved at 08 April 2025 13:09
14	31/01/2025	08:00	20:00	[CHRIS] Sprint Retro	Approved at 08 April 2025 13:09
15	03/02/2025	08:00	20:00	[CHRIS] Code Review [CHRIS] Implement Pagination at User, Leave, and Daily	Approved at 08 April 2025 13:09
16	04/02/2025	08:00	20:00	[CHRIS] Implement Pagination at Leave, and Daily Attendance	Approved at 08 April 2025 13:09
17	05/02/2025	08:00	20:00	[CHRIS] Implement Pagination at Leave	Approved at 08 April 2025 13:09
18	06/02/2025	08:00	20:00	[CHRIS] Implement Pagination at Leave (cont.) [Internal] Monthly Meeting	Approved at 08 April 2025 13:09

**Notes:**

1. Copied Form must be attached in report when registering for exam



In witness whereof the company,



Head Of Development  
Please sign along with the Company's stamp



# N U S A N T A R A

## Daily Task

**STUDENT ID** : 00000067097

**STUDENT NAME** : Jose Andreas Lie

**COMPANY NAME** : Concise (PT Ganda Visi Jayatama)

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
19	07/02/2025	08:00	20:00	[CHRIS] Making new column for the new leave permit table [CHRIS] Code Review	Approved at 08 April 2025 13:09
20	08/02/2025	08:00	17:00	[CHRIS] Creating form leave request submission API	Approved at 08 April 2025 13:09
21	10/02/2025	08:00	20:00	[CHRIS] Implement Pagination at Leave Types [CHRIS] Code Review	Approved at 08 April 2025 13:09
22	11/02/2025	08:00	20:00	[CHRIS] Revise Pagination at Leave Permit	Approved at 08 April 2025 13:09
23	12/02/2025	08:00	20:00	[CHRIS] Adding more API for leave [CHRIS] Integrate with FE	Approved at 08 April 2025 13:09
24	13/02/2025	08:00	20:00	[CHRIS] Revise Response Leave for Integrating with FE	Approved at 08 April 2025 13:09
25	14/02/2025	08:00	20:00	[CHRIS] Revise Leave Accept and Reject [CHRIS] Chris Show Case	Approved at 08 April 2025 13:09
26	15/02/2025	08:00	17:00	[CHRIS] Revise Leave Permit APIs (cont.)	Approved at 08 April 2025 13:09
27	17/02/2025	08:00	20:00	[CHRIS] File upload at leave permit [CHRIS] Revise Response Leave	Approved at 08 April 2025 13:09
28	18/02/2025	08:00	20:00	[CHRIS] Leave Permit Dashboard Filtering	Approved at 08 April 2025 13:09

**Notes:**

1. Copied Form must be attached in report when registering for exam



In witness whereof the company,



Head Of Development

Please sign along with the Company's stamp



**N U S A N T A R A**

## Daily Task

**STUDENT ID** : 00000067097

**STUDENT NAME** : Jose Andreas Lie

**COMPANY NAME** : Concise (PT Ganda Visi Jayatama)

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
29	19/02/2025	08:00	20:00	[CHRIS] Cancel button for Leave Permit	Approved at 08 April 2025 13:09
30	20/02/2025	08:00	21:00	[CHRIS] Code Review [CHRIS] Revise Codes	Approved at 08 April 2025 13:09
31	21/02/2025	08:00	20:00	[CHRIS] Code Review [Internal] BE Weekly Meeting	Approved at 08 April 2025 13:10
32	22/02/2025	09:00	18:00	[CHRIS] Leave Management Filtering	Approved at 08 April 2025 13:10
33	24/02/2025	08:00	20:00	[CHRIS] Testing [CHRIS] Leave Management Filtering (cont.)	Approved at 08 April 2025 13:10
34	25/02/2025	08:00	20:00	[CHRIS] UAT [CHRIS] Testing and Preparation for UAT	Approved at 08 April 2025 13:10
35	26/02/2025	08:00	20:00	[CHRIS] Testing [CHRIS] Handle APIs for CHRISM (Mobile)	Approved at 08 April 2025 13:10
36	27/02/2025	08:00	20:00	[CHRIS] Revise Migration [CHRIS] Removing Reason Clock Out Early	Approved at 08 April 2025 13:10
37	28/02/2025	08:00	20:00	[CHRIS] Testing [CHRIS] Using tree for permission on accept reject leave permit	Approved at 08 April 2025 13:10

**Notes:**

1. Copied Form must be attached in report when registering for exam



In witness whereof the company,



Concise  
Head Of Development  
Please sign along with the Company's stamp



# N U S A N T A R A

## Daily Task

**STUDENT ID** : 00000067097

**STUDENT NAME** : Jose Andreas Lie

**COMPANY NAME** : Concise (PT Ganda Visi Jayatama)

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
38	01/03/2025	08:00	18:00	[CHRIS] Making tree system based on Job Titles	Approved at 08 April 2025 13:10
39	03/03/2025	13:01	21:00	[CHRIS] Sprint Retro [CHRIS] Using tree for permission on accept reject leave permit (cont.)	Approved at 08 April 2025 13:10
40	04/03/2025	13:01	21:00	[CHRIS] Making a tree function that called all of user supervisee	Approved at 08 April 2025 13:10
41	05/03/2025	13:01	21:00	[CHRIS] Implement Biometrics on Login API	Approved at 08 April 2025 13:10
42	06/03/2025	13:01	21:00	[CHRIS] Update APIDog for Biometrics [CHRIS] Prepare APIs for People Report	Approved at 08 April 2025 13:10
43	07/03/2025	13:01	21:00	[CHRIS] Creating user report new Summary API	Approved at 08 April 2025 13:10
44	08/03/2025	08:00	20:00	[CHRIS] Creating user report new Summary API (cont.)	Approved at 08 April 2025 13:10
45	10/03/2025	13:01	21:00	[CHRIS] Creating user report new Summary API (cont.)	Approved at 08 April 2025 13:10
46	11/03/2025	13:01	21:00	[CHRIS] Creating user report new Summary API (cont.)	Approved at 08 April 2025 13:10
47	12/03/2025	13:01	21:00	[CHRIS] Creating user report filter date	Approved at 08 April 2025 13:10

**Notes:**

1. Copied Form must be attached in report when registering for exam



In witness whereof the company,

  
 PT GANDA VISI JAYATAMA  
 IT SOLUTIONS  
 Eriko Setiawan  
 Head Of Development  
**Please sign along with the Company's stamp**



N U S A N T A R A

## Daily Task

**STUDENT ID** : 00000067097

**STUDENT NAME** : Jose Andreas Lie

**COMPANY NAME** : Concise (PT Ganda Visi Jayatama)

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
48	13/03/2025	13:01	21:00	[CHRIS] Testing	Approved at 08 April 2025 13:10
49	14/03/2025	13:01	21:00	[CHRIS] Refactor Leave Permit Dashboard Logic [CHRIS] User Report Minor Fix	Approved at 08 April 2025 13:10
50	15/03/2025	08:00	20:00	[CHRIS] User Report Minor Fix (cont.)	Approved at 08 April 2025 13:10
51	17/03/2025	13:01	21:00	[CHRIS] Sprint Retro [CHRIS] Testing	Approved at 08 April 2025 13:10
52	18/03/2025	13:01	21:00	[CHRIS] Revise APIDog [CHRIS] Research Payroll	Approved at 08 April 2025 13:10
53	19/03/2025	13:01	21:00	[CHRIS] Research Payroll (cont.)	Approved at 08 April 2025 13:10
54	20/03/2025	13:01	21:00	[CHRIS] Research Payroll (cont.)	Approved at 08 April 2025 13:10
55	21/03/2025	13:01	21:00	[CHRIS] Research Payroll and Table Scheme (cont.)	Approved at 08 April 2025 13:10
56	22/03/2025	08:00	20:00	[CHRIS] Research Payroll and Table Scheme (cont.)	Approved at 08 April 2025 13:10
57	24/03/2025	13:01	21:00	[CHRIS] Research Payroll and Table Scheme (cont.)	Approved at 08 April 2025 13:10
58	25/03/2025	13:01	21:00	[CHRIS] Update Payroll Apidog [CHRIS] Creating Payroll API	Approved at 08 April 2025 13:10

**Notes:**

1. Copied Form must be attached in report when registering for exam

In witness whereof the Company,



Eddy Setiawan

Head Of Development

Please sign along with the Company's stamp



N U S A N T A R A

## Daily Task

**STUDENT ID** : 00000067097

**STUDENT NAME** : Jose Andreas Lie

**COMPANY NAME** : Concise (PT Ganda Visi Jayatama)

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
59	26/03/2025	13:01	21:00	[CHRIS] Developing Payrolls APIs	Approved at 08 April 2025 13:10
60	27/03/2025	13:01	21:00	[CHRIS] Developing Payrolls APIs (cont.)	Approved at 08 April 2025 13:10
61	08/04/2025	13:01	18:00	[CHRIS] Sprint Closing	Approved at 16 Juni 2025 14:32
62	09/04/2025	13:01	18:00	[CHRIS] Backlog Planning [CHRIS] Develop Payroll API	Approved at 16 Juni 2025 14:32
63	10/04/2025	13:01	18:00	[CHRIS] Develop Payroll API	Approved at 16 Juni 2025 14:32
64	11/04/2025	13:01	18:00	[CHRIS] Develop Payroll API (cont.)	Approved at 16 Juni 2025 14:32
65	14/04/2025	13:01	18:00	[CHRIS] Developing Payrolls	Approved at 16 Juni 2025 14:32
66	15/04/2025	13:01	18:00	[CHRIS] Developing Payrolls (cont.)	Approved at 16 Juni 2025 14:32
67	16/04/2025	13:01	18:00	[CHRIS] Developing Payrolls API: Update and Delete	Approved at 16 Juni 2025 14:32
68	17/04/2025	13:01	18:00	[CHRIS] Developing Payrolls API: Each User Allowances	Approved at 16 Juni 2025 14:32
69	21/04/2025	13:01	18:00	[CHRIS] Developing Payrolls API: Each User Allowances (cont.)	Approved at 16 Juni 2025 14:32

**Notes:**

1. Copied Form must be attached in report when registering for exam



In witness whereof the Company,

Edu Setiawan  
Head Of Development

Please sign along with the Company's stamp



N U S A N T A R A

## Daily Task

**STUDENT ID** : 00000067097

**STUDENT NAME** : Jose Andreas Lie

**COMPANY NAME** : Concise (PT Ganda Visi Jayatama)

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
70	22/04/2025	13:01	18:00	[CHRIS] Developing Payrolls API: Each User Allowances (cont.)	Approved at 16 Juni 2025 14:32
71	23/04/2025	13:01	18:00	[CHRIS] Developing Payrolls API: Each User Allowances (cont.)	Approved at 16 Juni 2025 14:33
72	24/04/2025	13:01	18:00	[CHRIS] Developing Payrolls API: Each User Allowances (cont.)	Approved at 16 Juni 2025 14:33
73	25/04/2025	13:01	18:00	[CHRIS] Developing Payrolls API: Each User Allowances (cont.)	Approved at 16 Juni 2025 14:33
74	28/04/2025	13:01	18:00	[CHRIS] Developing Salary Slip APIs	Approved at 16 Juni 2025 14:33
75	29/04/2025	13:01	18:00	[CHRIS] Developing Salary Slip APIs (cont.)	Approved at 16 Juni 2025 14:33
76	30/04/2025	13:01	18:00	[CHRIS] Bugfix User Allowance APIs [CHRIS] Developing Salary Slip APIs	Approved at 16 Juni 2025 14:33
77	02/05/2025	13:01	18:00	[CHRIS] Refactor User Allowance and Payroll Configuration for FE [CHRIS] Developing Salary Slip APIs (cont.)	Approved at 16 Juni 2025 14:33
78	05/05/2025	13:01	18:00	[CHRIS] Fix Updated Endpoint on Payroll Configuration [CHRIS] Add User Allowance Delete API	Approved at 16 Juni 2025 14:33

**Notes:**

1. Copied Form must be attached in report when registering for exam



In witness whereof the company,



Head Of Development

Please sign along with the Company's stamp



# N U S A N T A R A

## Daily Task

**STUDENT ID** : 00000067097

**STUDENT NAME** : Jose Andreas Lie

**COMPANY NAME** : Concise (PT Ganda Visi Jayatama)

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
79	06/05/2025	13:01	18:00	[CHRIS] Adding new column on user, and it will be updated when user allowance created [CHRIS] Salary Slip APIs	Approved at 16 Juni 2025 14:33
80	07/05/2025	13:01	18:00	[CHRIS] Bugfix on User Allowance [CHRIS] Salary Slip APIs	Approved at 16 Juni 2025 14:33
81	08/05/2025	13:01	18:00	[CHRIS] Creating new API so User Management only need to hit one API for update user detail, payroll active, and the allowances	Approved at 16 Juni 2025 14:33
82	09/05/2025	13:01	18:00	[CHRIS] Salary Slip APIs	Approved at 16 Juni 2025 14:33
83	14/05/2025	13:01	18:00	[CHRIS] Integrating with FE for Salary Slip	Approved at 16 Juni 2025 14:33
84	15/05/2025	13:01	18:00	[CHRIS] Integrating with FE for Salary Slip (cont.)	Approved at 16 Juni 2025 14:33
85	16/05/2025	13:01	18:00	[CHRIS] Testing Payroll, Allowance, and Salary Slip	Approved at 16 Juni 2025 14:33
86	19/05/2025	13:01	18:00	[CHRIS] Fix Pagination on Salary Slip and Payroll Config	Approved at 16 Juni 2025 14:33
87	20/05/2025	13:01	18:00	[CHRIS] Fix Salary Slip Logic	Approved at 16 Juni 2025 14:33
88	21/05/2025	13:01	18:00	[CHRIS] Refactor Payroll and Salary Slip	Approved at 16 Juni 2025 14:33

**Notes:**

1. Copied Form must be attached in report when registering for exam



In witness whereof the company,

  
 PT GANDA VISI JAYATAMA  
 Head Of Development  
 Please sign along with the Company's stamp  


# N U S A N T A R A

## Daily Task

**STUDENT ID** : 00000067097

**STUDENT NAME** : Jose Andreas Lie

**COMPANY NAME** : Concise (PT Ganda Visi Jayatama)

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
89	22/05/2025	13:01	18:00	[CHRIS] Refactor/Research Payroll and Salary Slip (cont.)	Approved at 16 Juni 2025 14:33
90	23/05/2025	13:01	18:00	[CHRIS] Internal Report and Showcase Payroll	Approved at 16 Juni 2025 14:33
91	26/05/2025	13:01	18:00	[CHRIS] Showcase Payroll System	Approved at 16 Juni 2025 14:33
92	27/05/2025	13:01	18:00	[CHRIS] Backlog-ing for Organization	Approved at 16 Juni 2025 14:33
93	28/05/2025	13:01	18:00	[CHRIS] Organization Database Structuring	Approved at 16 Juni 2025 14:33
94	02/06/2025	13:01	18:00	[CHRIS] Migrations for Organization	Approved at 16 Juni 2025 14:33
95	03/06/2025	13:01	18:00	[CHRIS] Organization APIs	Approved at 16 Juni 2025 14:33
96	04/06/2025	13:01	18:00	[CHRIS] Organization CRUD APIs (cont.)	Approved at 16 Juni 2025 14:33
97	05/06/2025	13:01	18:00	[CHRIS] Organization CRUD APIs (cont.)	Approved at 16 Juni 2025 14:33
98	10/06/2025	13:01	18:00	[CHRIS] Organization Progress Review	Approved at 16 Juni 2025 14:33
99	11/06/2025	13:01	18:00	[CHRIS] Organization API Get	Approved at 16 Juni 2025 14:33

**Notes:**

1. Copied Form must be attached in report when registering for exam



In witness whereof the company,



Concise

Edo Setiawan

Head Of Development

Please sign along with the Company's stamp



N U S A N T A R A



## Daily Task

**STUDENT ID** : 00000067097

**STUDENT NAME** : Jose Andreas Lie

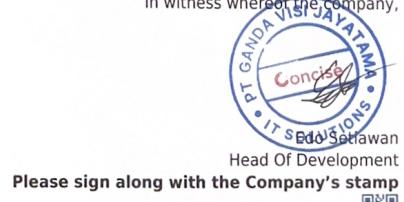
**COMPANY NAME** : Concise (PT Ganda Visi Jayatama)

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
100	12/06/2025	13:01	18:00	[CHRIS] Replan Organization Feature (IA, User Flow, Lo-fi)	Approved at 16 Juni 2025 14:33
101	13/06/2025	13:01	18:00	[CHRIS] Replan Organization Feature (IA, User Flow, Lo-fi) (cont.)	Approved at 16 Juni 2025 14:34

**Notes:**

- Copied Form must be attached in report when registering for exam**

In witness whereof the company,



N U S A N T A R A

## Lampiran 4. MBKM-04 Verification Form of Internship Report MBKM Internship Track 1

MBKM-04 Verification Form of Internship Report MBKM Internship Track 1



### VERIFICATION FORM OF INTERNSHIP REPORT MBKM INTERNSHIP TRACK 1

Student's Internship Track 1 Advisor

Name : Dr. Ivransa Zuhdi Pane, B.Eng., M.Eng.

I, who signed below

Name : Edo Setiawan

Position : Head Of Development

Company : Concise (PT Ganda Visi Jayatama)

had received, read and approved the Internship Report from

Student ID : 00000067097

Student Name : Jose Andreas Lie

Period : June 2025

Report Title : Backend Enhancement of the CHRIS HR Information System at PT Ganda Visi Jayatama

Tangerang , 18 Juni 2025

A handwritten signature in black ink, appearing to read "Ivransa Zuhdi Pane".

Dr. Ivransa Zuhdi Pane, B.Eng., M.Eng.  
Student Advisor

Tangerang 18 Jun 2025



Supervisor  
Sign along with the Company's stamp



Kampus UMN, Scientia Garden | Jl. Boulevard Gading Serpong - Tangerang | P. +62 21 5422 0808 | F. +62 21 5422 0800 | www.umn.ac.id

## Lampiran 5. Form Bimbingan

### Form Bimbingan Internship Report Program Studi Informatika Semester Genap 2024/2025

Nama : Jose Andreas Lie  
NIM : 00000067097  
Angkatan : 2022  
Dosen Pembimbing : Dr. Ivransa Zuhdi Pane, B.Eng., M.Eng.

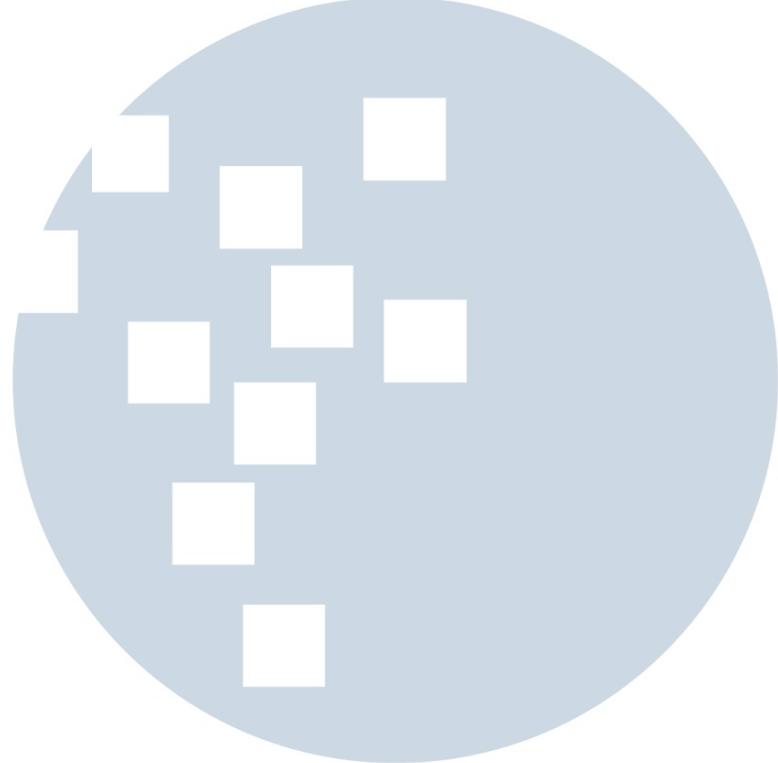
Meeting	Tanggal	Jam	Keterangan	Tanggal Approval
1	22 Februari 2025	09:00	Pertemuan 1 : Koordinasi Awal (Mekanisme Magang)	20 Mei 2025
2	15 Maret 2025	10:00	Presentasi profil perusahaan tempat magang, deskripsi penugasan, masalah atau kendala, solusi masalah atau kendala, rencana kerja berikutnya	20 Mei 2025
3	12 April 2025	10:00	Demo progress hasil kerja (Backend)	20 Mei 2025
4	10 Mei 2025	16:00	Update progress internship Update progress internship report	20 Mei 2025
5	31 Mei 2025	16:00	Update progress laporan dan pekerjaan magang	13 Juni 2025
6	14 Juni 2025	16:00	Progress Magang - Hasil kerja, kendala - Demo hasil kerja next meeting - Reminder supervisor evaluation grading Laporan Magang - Submit ke pak Pane sebelum submit sistem - ttd basah di verification form - turnitin max 30% Deadline Regis Magang 23/6/25 Sidang Magang - dipercepat menjadi 30/6/25 - 09/07/25 - waktu 45 menit, 15 demo present + 30 QNA session - Demo Real Time	18 Juni 2025
7	19 Juni 2025	11:00	Finalisasi laporan dan tanda tangan	18 Juni 2025
8	21 Juni 2025	16:00	Demo hasil kerja dan simulasi presentasi	18 Juni 2025

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



## Lampiran 6. Transkrip Wawancara

appendix appendix appendix appendix appendix appendix



**UMN**  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA