

**PENGEMBANGAN BACKEND MODUL PAYROLL SISTEM
INFORMASI KEPEGAWAIAN NINE TO SIX PADA
PT VISI KARYA NUSANTARA**



UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA

LAPORAN CAREER ACCELERATION PROGRAM

**JOSE ANDREAS LIE
00000067097**

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG
2026**

**PENGEMBANGAN BACKEND MODUL PAYROLL SISTEM
INFORMASI KEPEGAWAIAN NINE TO SIX PADA
PT VISI KARYA NUSANTARA**



LAPORAN CAREER ACCELERATION PROGRAM

Diajukan sebagai Salah Satu Syarat untuk Memperoleh

Gelar Sarjana S.Kom

JOSE ANDREAS LIE
Jose Andreas Lie

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG
2026

HALAMAN PERNYATAAN ORISINALITAS TIDAK PLAGIAT

Dengan ini saya,

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

Menyatakan dengan sesungguhnya bahwa Laporan Career Acceleration Program saya yang berjudul:

Pengembangan Backend Modul Payroll Sistem Informasi Kepegawaian Nine to Six pada PT Visi Karya Nusantara

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, 1 Januari 2026



(Jose Andreas Lie)

HALAMAN PERNYATAAN PENGGUNAAN BANTUAN KECERDASAN ARTIFISIAL (AI)

Saya yang bertanda tangan di bawah ini:

Nama : Jose Andreas Lie
NIM : 00000067097
Program Studi : Informatika
Judul Laporan : Pengembangan Backend Modul Payroll Sistem Informasi Kepegawaian Nine to Six pada PT Visi Karya Nusantara

Dengan ini saya menyatakan secara jujur menggunakan bantuan Kecerdasan Artifisial (AI) dalam penggerjaan Laporan sebagai berikut:

- Menggunakan AI sebagaimana diizinkan untuk membantu dalam menghasilkan ide-ide utama saja
- Menggunakan AI sebagaimana diizinkan untuk membantu menghasilkan teks pertama saja
- Menggunakan AI untuk menyempurnakan sintaksis dan tata bahasa untuk pengumpulan tugas
- Karena tidak diizinkan: Tidak menggunakan bantuan AI dengan cara apa pun dalam pembuatan tugas

Saya juga menyatakan bahwa:

- (1) Menyerahkan secara lengkap dan jujur penggunaan perangkat AI yang diperlukan dalam tugas melalui Formulir Penggunaan Perangkat Kecerdasan Artifisial (AI)
- (2) Mengakui telah menggunakan bantuan AI dalam tugas saya baik dalam bentuk kata, paraphrase, penyertaan ide atau fakta penting yang disarankan oleh AI dan saya telah menyantumkan dalam sitasi serta referensi
- (3) Terlepas dari pernyataan di atas, tugas ini sepenuhnya merupakan karya saya sendiri

Tangerang, 1 Januari 2026



(Jose Andreas Lie)

HALAMAN PERNYATAAN KEABSAHAN PERUSAHAAN

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

menyatakan bahwa saya melaksanakan kegiatan di:

Nama Perusahaan/Organisasi : PT. Visi Karya Nusantara
Alamat : Start Space Coworking Space Gading Serpong, Tangerang, Banten, 15810
Email Perusahaan/Organisasi : aftersixidn@gmail.com

1. Perusahaan/Organisasi tempat saya melakukan kegiatan dapat di validasi keberadaannya.
2. Jika dikemudian hari, terbukti ditemukan kecurangan/penyimpangan data yang tidak valid di perusahaan/organisasi tempat saya melakukan kegiatan, maka:
 - a. Saya bersedia menerima konsekuensi dinyatakan TIDAK LULUS untuk mata kuliah yang telah saya tempuh.
 - b. Saya bersedia menerima semua sanksi yang berlaku sebagaimana ditetapkan dalam peraturan yang berlaku di Universitas Multimedia Nusantara.

Pernyataan ini saya buat dengan sebenar-benarnya dan digunakan sebagaimana mestinya

Tangerang, (1 Januari 2026)

Mengetahui



(Atanasius Raditya Herkristito)

Menyatakan



(Jose Andreas Lie)

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 Career Acceleration Program

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, 1 Januari 2026

Yang menyatakan



Jose Andreas Lie

** 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. Ibu Fenina Adline Twince Tobing, S.Kom., M.Kom., sebagai Pembimbing pertama yang telah banyak meluangkan waktu untuk memberikan bimbingan, arahan dan motivasi atas terselesaiya tesis ini.
5. Keluarga dan teman-teman saya yang telah memberikan bantuan dukungan material dan moral, sehingga penulis dapat menyelesaikan tesis ini.

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, 1 Januari 2026



Jose Andreas Lie

**PENGEMBANGAN BACKEND MODUL PAYROLL SISTEM INFORMASI
KEPEGAWAIAN NINE TO SIX PADA
PT VISI KARYA NUSANTARA**

Jose Andreas Lie

ABSTRAK

Laporan ini menjelaskan pengembangan sistem informasi kepegawaian Nine to Six (NTS) di PT Visi Karya Nusantara, sebuah perusahaan konsultan teknologi informasi yang berfokus pada solusi digital inovatif. Digitalisasi pengelolaan sumber daya manusia (SDM) telah menjadi kebutuhan kritis untuk meningkatkan efisiensi operasional, akurasi penggajian, dan kepatuhan regulasi. Penelitian menunjukkan bahwa integrasi modul-modul seperti *Contract Management*, *Overtime Management*, dan *Payslip* secara signifikan mengurangi kesalahan manual dan meningkatkan transparansi data. Proyek ini dirancang untuk mengembangkan sistem terpadu yang menghubungkan data statis (kontrak karyawan), data transaksional (kehadiran dan lembur), dan proses finalisasi kompensasi (*payslip*) dalam satu platform berbasis *cloud*. Menggunakan PERN Stack (PostgreSQL, Express.js, React.js, Node.js), sistem dikembangkan dengan metodologi *Agile* selama periode magang dari 4 Agustus 2025 hingga 4 Januari 2026. Hasil pengembangan mencakup implementasi modul *Contract Management*, *Payroll* dengan komponen otomatis, *Overtime Management*, *On Boarding* berbasis CSV, serta *Payslip* terintegrasi dengan sistem *Daily Attendance*. Sistem ini diharapkan meningkatkan efisiensi HR, menjamin akurasi perhitungan kompensasi, dan memberikan transparansi penuh kepada karyawan melalui akses *self-service*.

Kata Kunci: *Backend development, Express, Human resource information system, On boarding, Payroll.*

**BACKEND ENHANCEMENT OF NINE TO SIX PAYROLL MODULE AT PT
VISI KARYA NUSANTARA**

Jose Andreas Lie

ABSTRACT

This report describes the development of the Nine to Six (NTS) human resource information system at PT Visi Karya Nusantara, a technology information consulting company focused on innovative digital solutions. The digitalization of human resource management has become a critical necessity to improve operational efficiency, payroll accuracy, and regulatory compliance. Research demonstrates that the integration of modules such as Contract Management, Overtime Management, and Payslip significantly reduces manual errors and enhances data transparency. This project was designed to develop an integrated system that connects static data (employee contracts), transactional data (attendance and overtime), and compensation finalization processes (payslip) within a single cloud-based platform. Using the PERN Stack (PostgreSQL, Express.js, React.js, Node.js), the system was developed with Agile methodology during the internship period from August 4, 2025 to January 4, 2026. The development outcomes include the implementation of Contract Management module, Payroll with automated components, Overtime Management, CSV-based On Boarding, and Payslip integrated with the Daily Attendance system. This system is expected to enhance HR efficiency, ensure accurate compensation calculations, and provide full transparency to employees through self-service access.

Keywords: Backend development, Express, Human resource information system, On boarding, Payroll.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

Lampiran 1	PRO-STEP-01 Cover Letter	85
Lampiran 2	PRO-STEP-02 Internship Card	86
Lampiran 3	PRO-STEP-03 Daily Task	87
Lampiran 4	PRO-STEP-04 Verification Form	98
Lampiran 5	PRO-STEP-05 Surat Penerimaan	99
Lampiran 6	Form Bimbingan	100
Lampiran 7	Hasil Pengecekan Similarity Turnitin	101
Lampiran 8	Formulir Penggunaan Perangkat Kecerdasan Artifisial (AI) . .	104

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Transformasi digital dalam pengelolaan sumber daya manusia (SDM) telah menjadi salah satu faktor utama dalam peningkatan efisiensi operasional perusahaan di era industri 4.0. Penerapan sistem informasi kepegawaian atau *Human Resource Information System* (HRIS) memungkinkan proses administrasi dan pengambilan keputusan berbasis data dilakukan secara lebih cepat dan akurat [?]. Digitalisasi fungsi SDM tidak hanya berdampak pada efisiensi kerja, tetapi juga membuka potensi untuk peningkatan transparansi dan konsistensi data pegawai [?].

Salah satu aspek penting dari digitalisasi SDM adalah otomatisasi proses penggajian atau *payroll automation*. Menurut penelitian oleh Onuotu dan Amaewhule (2024), penerapan sistem akuntansi terotomatisasi pada konteks usaha kecil dan menengah secara signifikan dapat meningkatkan efisiensi dan akurasi dalam persiapan penggajian, dengan mencatatkan pengurangan kesalahan perhitungan dan penurunan beban kerja manual [?]. Hal ini mengindikasikan bahwa, terutama dalam skenario transisi dari proses manual, integrasi sistem penggajian digital sangat efektif dalam mempercepat proses administrasi. Meskipun demikian, perlu dicatat bahwa otomatisasi *payroll* berfokus pada akurasi perhitungan dan tidak serta merta menjamin validitas data masukan, menyoroti pentingnya keakuratan data sumber.

Lebih lanjut, menurut Meenugu (2025), evolusi sistem penggajian dari proses manual menuju otomatisasi cerdas tidak hanya meningkatkan efisiensi dan akurasi, tetapi juga memperkuat kepatuhan terhadap regulasi [?]. Meskipun demikian, masih banyak sistem penggajian yang berjalan secara terpisah dari modul HRIS utama, sehingga menimbulkan inefisiensi akibat kebutuhan untuk memasukkan data secara berulang dari sistem lain seperti absensi dan manajemen cuti. Kondisi ini kembali membuka peluang terjadinya kesalahan manusia dan menghambat efektivitas kerja. Oleh karena itu, diperlukan integrasi yang mulus antara pengelolaan data pegawai dan proses penggajian dalam satu platform terpadu.

Perhitungan variabel lembur dan absensi menjadi sumber kerumitan dan potensi sengketa. Tanpa sistem yang terintegrasi, penetapan jam lembur yang akurat

memerlukan waktu dan rentan terhadap kesalahan manual atau manipulasi data (*buddy punching*). Modul Lembur dan Absensi (*Overtime and Absence Module*) dalam HRIS modern berfungsi sebagai alat untuk memastikan pencatatan waktu yang tepat dan otomatis. Penggunaan sistem pencatatan waktu kerja elektronik kini dianggap sebagai komponen penting untuk memastikan kepatuhan terhadap regulasi jam kerja (*working time regulations*), karena memungkinkan pemantauan jam kerja yang akurat dan efisien [?]. Data dari modul ini sangat vital karena merupakan input transaksional yang mempengaruhi Modul Payslip.

Integrasi dan akurasi data yang diupayakan oleh modul variabel seperti lembur dan penggajian bergantung secara fundamental pada ketersediaan data master pegawai yang konsisten dan terstandardisasi [?]. Oleh karena itu, Modul Kontrak Karyawan (*Employee Contract Module*) harus berfungsi sebagai basis data primer di dalam HRIS, yang menyimpan informasi krusial seperti struktur gaji dasar, tunjangan tetap, periode kontrak, dan ketentuan jadwal kerja. Konsep ini sesuai dengan evolusi HRIS modern berbasis komputasi awan (*cloud computing*) yang dirancang untuk menjadi sumber tunggal kebenaran (*single source of truth*) bagi seluruh data kepegawaian organisasi [?]. Kegagalan dalam sentralisasi ini, seperti yang terjadi pada sistem tradisional, menyebabkan data kepegawaian tersebar dalam dokumen fisik atau file terisolasi, yang menyulitkan pembaruan serentak dan validasi silang data.

Digitalisasi Modul Kontrak memiliki peran ganda sebagai fondasi data dan sebagai penjaga kepatuhan. Secara inheren, modul ini berfungsi untuk mengeliminasi risiko ketidakpatuhan hukum (*legal compliance risk*) yang timbul dari *template* kontrak yang tidak seragam atau klausul yang sudah usang. Dengan mengimplementasikan sistem yang mengacu pada kebijakan perusahaan dan regulasi ketenagakerjaan yang berlaku, HRIS dapat memastikan standardisasi kontrak bagi seluruh karyawan. Selain itu, sentralisasi data kontrak ini memungkinkan integrasi yang mulus antara data personalia dan database penggajian, sebuah faktor kritis yang telah terbukti meningkatkan transparansi dan akuntabilitas dalam pengelolaan kompensasi [?]. Hal ini menyoroti bahwa kualitas informasi SDM sangat bergantung pada seberapa baik data kontrak dikelola sejak awal.

Setelah data dasar (Data Statis), seperti yang terdapat di Modul Kontrak, dan data variabel (Data Transaksional), seperti yang diperoleh dari Modul Lembur, Absensi, dan Reimbursement berhasil dikumpulkan dan divalidasi, tantangan selanjutnya adalah pada proses finalisasi kompensasi, yang diwujudkan dalam

Modul Payslip Karyawan (*Payslip Module*). Proses pembuatan payslip secara manual adalah tugas yang sangat kompleks dan rentan kesalahan, karena harus menggabungkan perhitungan gaji pokok, tunjangan, potongan pajak, iuran wajib (statutory deduction), serta komponen variabel seperti lembur dan keterlambatan. Studi kasus menunjukkan bahwa sistem perhitungan gaji manual menghadapi inefisiensi yang signifikan [?]. Oleh karena itu, digitalisasi dan otomatisasi modul ini sangat penting untuk memastikan akurasi perhitungan dan efektivitas [?].

Peran Modul Payslip digital tidak hanya terbatas pada pencetakan dokumen, tetapi sebagai titik akhir validasi data. Dengan mengadopsi sistem berbasis web terintegrasi, Modul Payslip menyediakan audit trail yang jelas, memungkinkan HR untuk meninjau secara rinci sumber data dari setiap komponen gaji [?]. Bagi karyawan, modul ini meningkatkan transparansi karena mereka dapat mengakses slip gaji mereka secara mandiri (self-service) dan melihat perincian potongan atau penambahan secara real-time. Adopsi sistem HRIS seperti ini merupakan revolusi yang diadopsi korporasi untuk meningkatkan operasi harinya [?], yang pada akhirnya meningkatkan kepercayaan dan pengalaman pegawai (employee experience) secara keseluruhan [?].

Meskipun berbagai literatur dan implementasi HRIS telah menunjukkan manfaat signifikan dari digitalisasi fungsi SDM secara terpisah—baik itu efisiensi penggajian [?] maupun kepatuhan manajemen waktu [?] masih terdapat kesenjangan operasional dalam realisasi sistem yang benar-benar terintegrasi dan berpusat pada data master. Banyak organisasi masih menghadapi kesulitan dalam menyinkronkan data statis dari Modul Kontrak dengan data transaksional dari Modul Lembur dan Absensi untuk diolah di Modul Payslip, menunjukkan adanya *silo* data yang menghambat otomatisasi menyeluruh [?]. Kegagalan dalam mengintegrasikan data personalia dan *database* penggajian, misalnya, merupakan tantangan utama yang harus diatasi untuk menjamin transparansi dan akuntabilitas sistem kompensasi [?].

Tantangan integrasi ini menuntut adanya perancangan HRIS yang tidak hanya menyediakan fungsi-fungsi tersebut, tetapi juga berfokus pada arsitektur data tunggal yang mulus (*single source of truth*). Konsep arsitektur ini, yang didukung oleh sistem berbasis *cloud* modern, menjadi kunci untuk mendapatkan informasi SDM yang konsisten dan akurat [?]. Proyek perancangan ini hadir untuk menjembatani kesenjangan operasional tersebut. Tujuan dari perancangan sistem ini adalah untuk mengembangkan dan mengimplementasikan modul HRIS yang mengintegrasikan Modul Kontrak, Modul Lembur, dan Modul Payslip dalam satu

platform terpadu. Integrasi ini diharapkan dapat meningkatkan efisiensi operasional HR secara signifikan, memastikan keakuratan data master, menjamin kepatuhan perhitungan kompensasi, dan menyediakan transparansi penuh kepada karyawan, sehingga mendukung pengambilan keputusan strategis yang lebih baik.

1.2 Maksud dan Tujuan Kerja Magang

Pelaksanaan kerja magang di PT Visi Karya Nusantara bertujuan untuk memberikan pengalaman langsung kepada mahasiswa dalam menghadapi tantangan nyata di industri pengembangan perangkat lunak, serta berkontribusi terhadap transformasi digital di bidang pengelolaan sumber daya manusia. Melalui kegiatan magang ini, peserta memperoleh kesempatan untuk menerapkan pengetahuan akademik ke dalam praktik profesional, khususnya dalam pengembangan sistem informasi kepegawaian berbasis web.

Secara khusus, maksud dari pelaksanaan kerja magang ini adalah:

- Mengimplementasikan pengetahuan yang telah diperoleh selama perkuliahan ke dalam lingkungan kerja profesional yang menerapkan standar industri.
- Mengasah keterampilan teknis dalam pengembangan sistem berbasis *PERN stack* (PostgreSQL, Express.js, React, Node.js) serta kemampuan kolaborasi dalam tim lintas divisi.
- Memahami proses kerja profesional dalam pengembangan dan pemeliharaan sistem *backend* yang terstruktur, terdokumentasi, dan terintegrasi dengan sistem lainnya.

Adapun tujuan utama dari pelaksanaan kerja magang ini, yang difokuskan pada pengembangan sistem Nine to Six (NTS), meliputi:

1. Merancang dan mengimplementasikan modul Contract Management untuk mendukung pembuatan serta penugasan kontrak kerja kepada karyawan secara efisien.
2. Merancang dan mengimplementasikan modul Payslip Management yang terintegrasi dengan modul Daily Attendance, sehingga proses perhitungan gaji dapat dilakukan secara otomatis berdasarkan data kehadiran.

3. Menyediakan fitur pengurangan otomatis terhadap ketidakhadiran tanpa keterangan (*Missing in Action (MIA)*) agar perhitungan penggajian menjadi lebih akurat dan transparan.
4. Meningkatkan integrasi sistem antara sisi *backend* dan *frontend* untuk memastikan alur data yang konsisten dan *real-time*.
5. Mengembangkan *Application Programming Interface (API)* terstandarisasi yang dapat digunakan oleh HR maupun karyawan untuk mengakses data penggajian secara aman dan terukur.

Dengan pencapaian tujuan-tujuan tersebut, diharapkan sistem Nine to Six (NTS) dapat menjadi fondasi penting dalam mendukung efisiensi operasional perusahaan serta meningkatkan transparansi dan akurasi dalam pengelolaan penggajian di PT Visi Karya Nusantara.

1.3 Waktu dan Prosedur Pelaksanaan Kerja Magang

Pelaksanaan kerja magang di PT Visi Karya Nusantara (NTS) berlangsung dari tanggal 4 Agustus 2025 sampai dengan 4 Januari 2026 berdasarkan kontrak kerja yang telah disepakati dengan perusahaan. Selama periode magang ini, penulis dibimbing oleh Bapak Raditya selaku *Supervisor* dalam tim *Software Development*. Program ini dirancang untuk memberikan pengalaman langsung dalam proses pengembangan perangkat lunak profesional, termasuk pemahaman terhadap alur kerja kolaboratif dan siklus pengembangan produk di lingkungan industri.

Jadwal dan ketentuan kerja magang di PT Visi Karya Nusantara diatur sebagai berikut:

1. Aktivitas kerja magang dilaksanakan secara *remote*, dengan total waktu kerja sebesar 35 jam per minggu.
2. Waktu kerja bersifat fleksibel namun tetap mengikuti tanggung jawab proyek dan jadwal yang telah disepakati bersama *supervisor*.
3. Seluruh aktivitas kerja dilakukan secara daring dengan koordinasi melalui *platform* komunikasi *Discord*.

Selama menjalani program kerja magang, peserta wajib mengikuti sejumlah prosedur yang telah ditetapkan oleh perusahaan, antara lain:

1. Mengikuti sesi *onboarding* dan *orientation meeting* pada awal masa magang untuk memahami sistem kerja dan proyek yang akan dikerjakan.
2. Melakukan pelaporan harian melalui *daily updates* yang mencakup tugas yang telah diselesaikan (*yesterday tasks*), rencana pekerjaan (*today tasks*), serta hambatan yang dihadapi (*blocking issues*).
3. Berpartisipasi dalam *sprint meeting* mingguan untuk melaporkan progres, melakukan evaluasi hasil kerja, serta membahas rencana pengembangan modul selanjutnya.
4. Melakukan komunikasi aktif dengan tim pengembang lain, baik pada sisi *backend*, *frontend* maupun *design*, guna memastikan integrasi sistem berjalan dengan optimal.
5. Menyerahkan laporan akhir dan hasil proyek kepada *supervisor* pada akhir periode magang sebagai bentuk evaluasi kinerja dan kontribusi selama program berlangsung.

BAB 2

GAMBARAN UMUM PERUSAHAAN

2.1 Sejarah Singkat Perusahaan

PT Visi Karya Nusantara didirikan pada tahun 2025, berdasarkan informasi yang diperoleh melalui wawancara langsung dengan Bapak Raditya selaku *Software Engineer Manager*. Sebagai perusahaan konsultan teknologi informasi, PT Visi Karya Nusantara berfokus pada penyediaan solusi IT yang inovatif dan efektif bagi berbagai klien di Indonesia. Dalam kurun waktu yang singkat, perusahaan telah berhasil membangun reputasi yang solid di industri ini dengan menyelesaikan berbagai proyek menantang dan memberikan nilai tambah yang signifikan. Berlokasi di Start Space Coworking Space Gading Serpong, Tangerang, PT Visi Karya Nusantara saat ini aktif mengerjakan beberapa proyek utama, di antaranya adalah Nine to Six (NTS), AKS, dan KAIYA [?]. Logo resmi perusahaan dapat dilihat pada Gambar ??.



Gambar 2.1. Logo Perusahaan PT Visi Karya Nusantara

Gambar ?? merupakan logo dari perusahaan PT Visi Karya Nusantara. Berdasarkan wawancara langsung dengan Bapak Raditya selaku *Software Engineer Manager*,

2.2 Visi dan Misi Perusahaan

Visi dan misi perusahaan PT Visi Karya Nusantara diperoleh melalui wawancara langsung dengan Bapak Raditya selaku *Software Engineer Manager*.

Visi dari PT Visi Karya Nusantara adalah:

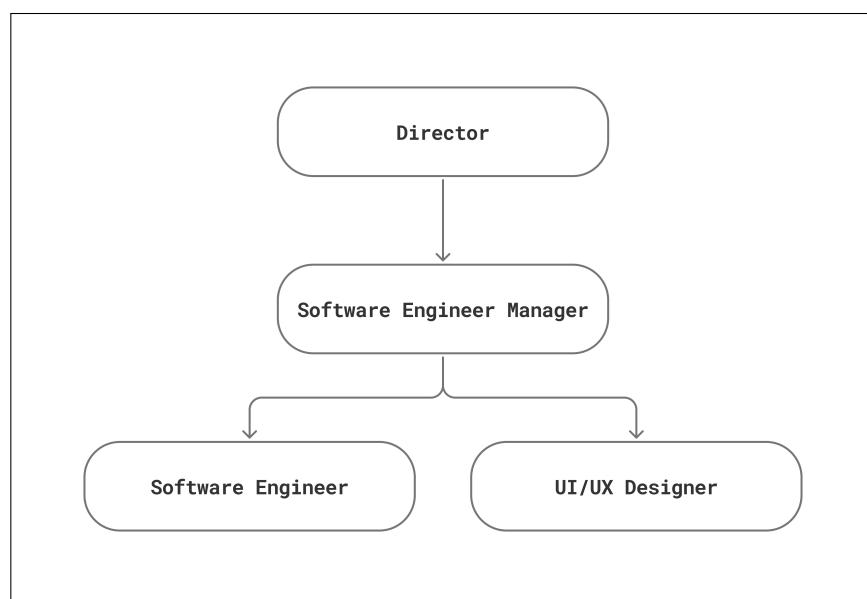
"Menjadi perusahaan konsultan teknologi informasi yang memberikan solusi yang optimal, efektif, dan efisien, serta membangun budaya perusahaan yang berintegritas, profesional, dan berkelanjutan."

Misi dari PT Visi Karya Nusantara adalah:

1. Menyediakan layanan konsultasi teknologi informasi yang berorientasi pada efektivitas dan efisiensi, guna memastikan setiap solusi yang diberikan selaras dengan kebutuhan dan tujuan klien.
2. Membangun organisasi yang profesional dan berintegritas melalui pengembangan sumber daya manusia yang adaptif, kompeten, dan berdaya saing, serta menumbuhkan budaya kerja yang kolaboratif, inovatif, dan berkelanjutan.
3. Berperan aktif dalam mendukung transformasi digital di Indonesia dengan mengedepankan praktik konsultasi yang beretika, berkualitas, dan berorientasi pada hasil jangka panjang.

2.3 Struktur Organisasi Perusahaan

Struktur organisasi perusahaan PT Visi Karya Nusantara dapat dilihat pada Gambar ??.



Gambar 2.2. Struktur organisasi perusahaan PT Visi Karya Nusantara

Struktur organisasi PT Visi Karya Nusantara terdiri atas tiga tingkatan utama yang membentuk alur koordinasi kerja secara efektif dan terarah. Pada tingkat tertinggi, terdapat Director, yang memegang tanggung jawab utama dalam menentukan visi, arah strategis, serta pengambilan keputusan penting perusahaan.

Di bawahnya terdapat Software Engineer Manager, yang berperan dalam mengatur dan mengawasi seluruh kegiatan teknis yang berkaitan dengan pengembangan perangkat lunak. Posisi ini juga bertugas untuk memberikan arahan teknis, melakukan evaluasi hasil kerja tim, serta memastikan proyek pengembangan berjalan sesuai standar dan tengat waktu yang telah ditetapkan.

Selanjutnya, di bawah koordinasi *Software Engineer Manager*, terdapat dua peran penting yaitu Software Engineer dan UI/UX Designer. *Software Engineer* bertanggung jawab atas proses pengembangan aplikasi, mulai dari implementasi fitur hingga integrasi sistem. Sementara itu, *UI/UX Designer* berfokus pada perancangan antarmuka pengguna serta pengalaman pengguna agar produk yang dikembangkan tidak hanya berfungsi dengan baik, tetapi juga memiliki tampilan yang menarik dan mudah digunakan.

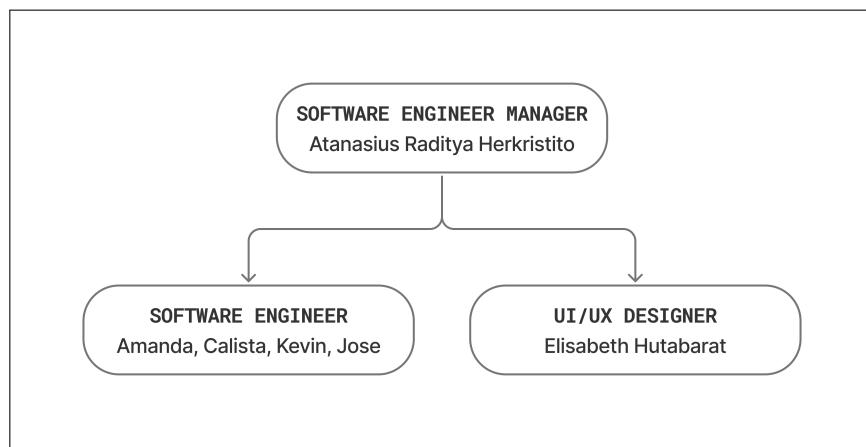
Struktur organisasi ini menggambarkan alur kerja yang ringkas namun efektif, dengan fokus pada kolaborasi lintas peran untuk menghasilkan produk digital yang berkualitas tinggi dan sesuai dengan kebutuhan pengguna.

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama pelaksanaan kerja magang di PT Visi Karya Nusantara (NTS), penulis menempati posisi sebagai *Software Engineer Intern* dalam *Software Development Team*. Peran utama yang dijalankan berfokus pada pengembangan dan pengujian *Application Programming Interface* (API) menggunakan Node.js dan Express.js, serta integrasinya dengan basis data PostgreSQL. Selain itu, penulis juga berkolaborasi dengan tim *frontend* dalam membangun dan mengimplementasi modul *Payroll* yang terhubung dengan sistem *Daily Attendance*.



Gambar 3.1. Struktur Tim Pengembang PT Visi Karya Nusantara (NTS)

Struktur tim pengembang dapat dilihat pada Gambar ??, yang terdiri dari beberapa anggota dengan peran yang saling terintegrasi. Tim ini berada di bawah koordinasi Bapak Atanasius Raditya Herkristito selaku *Software Engineer Manager*, yang berperan sebagai pembimbing sekaligus pengarah teknis selama masa magang. Beliau bertanggung jawab dalam memberikan umpan balik, melakukan evaluasi mingguan terhadap progres proyek, serta mengadakan sesi *code review* untuk memastikan kualitas kode sesuai standar perusahaan.

Dalam pelaksanaan proyek, kolaborasi dilakukan bersama anggota tim lainnya yang terdiri dari:

- Amanda — *Software Engineer Intern*
- Calista — *Software Engineer Intern*

- Kevin — *Software Engineer Intern*
- Elisabeth Hutabarat — *UI/UX Designer*

Seluruh kegiatan magang dilaksanakan secara *remote*, sehingga koordinasi tim dilakukan secara daring melalui *platform Discord* dan *repository management system GitHub*. *Discord* digunakan sebagai sarana komunikasi utama untuk berdiskusi, menyampaikan pembaruan progres, serta melakukan rapat mingguan. Sementara itu, *GitHub* digunakan untuk mengelola kode sumber, melakukan *pull request*, serta memfasilitasi proses *code review* dan kolaborasi antar anggota tim.

Setiap minggu diadakan *sprint meeting* yang dipimpin oleh Raditya untuk membahas perkembangan proyek, hambatan teknis, serta rencana pengembangan fitur berikutnya. Selain itu, setiap anggota tim juga melakukan *asynchronous daily updates* melalui *Discord* yang berisi laporan progres harian dan kendala yang dihadapi. Evaluasi hasil kerja dilakukan secara berkala melalui sesi *code review* dan *sprint retrospective* untuk memastikan kualitas pengembangan tetap terjaga serta selaras dengan standar teknis perusahaan.

3.2 Tugas yang Dilakukan

Selama pelaksanaan kerja magang di PT Visi Karya Nusantara, penulis terlibat dalam proyek pengembangan aplikasi *Internal System*, yaitu sebuah sistem kepegawaian berbasis web yang dikembangkan tidak hanya untuk kebutuhan internal perusahaan, tetapi juga ditujukan sebagai produk yang dapat digunakan oleh perusahaan lain. Fokus utama pekerjaan mencakup pengembangan sisi sistem dan integrasi antara backend dan frontend untuk mendukung fungsionalitas dan skalabilitas produk.

Tugas-tugas yang dilakukan meliputi:

1. Mengembangkan dan memelihara API untuk aplikasi *Internal System*, termasuk pembuatan fitur baru, serta memastikan kompatibilitas dengan sistem lain yang akan menggunakan produk ini.
2. Melakukan dokumentasi API menggunakan *platform Apidog* untuk mempermudah kolaborasi antar tim dan memfasilitasi integrasi dengan pihak eksternal.

3. Melakukan pengujian terhadap API dan fitur yang dikembangkan, mencakup validasi data, penanganan kesalahan, serta pengujian fungsionalitas untuk memastikan sistem berjalan stabil.
4. Melakukan koordinasi dan kolaborasi dengan tim pengembang melalui *Discord* dan *GitHub* terkait pembagian tugas, pembahasan revisi, serta proses integrasi antar modul.

Seluruh aktivitas tersebut berperan penting dalam mendukung terciptanya sistem yang andal, terukur, dan siap diimplementasikan secara luas. Dengan adanya dokumentasi dan proses pengembangan yang terstruktur, sistem ini tidak hanya memenuhi kebutuhan operasional internal, tetapi juga mampu beradaptasi dengan kebutuhan perusahaan lain yang menjadi calon pengguna produk. Hal ini menjadi bagian dari upaya strategis perusahaan dalam membangun solusi digital yang efisien dan berkelanjutan.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel ??.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Melakukan perancangan awal modul <i>Payroll</i> , mencakup pembuatan skema basis data, model, serta pengembangan awal API untuk <i>Contract</i> , <i>Payslip</i> , dan <i>Payroll Setting</i> .
2	Melanjutkan pengembangan <i>Contract API</i> dengan penambahan fitur pembatalan, persetujuan, dan penghapusan kontrak, melakukan refaktor pada <i>controller</i> , serta menambahkan seeding data untuk tabel pendukung seperti <i>misc</i> , <i>penalty rules</i> , dan <i>schedules</i> .
3	Melakukan perbaikan pada <i>Contract API</i> , menambahkan fitur baru untuk <i>Job Title</i> termasuk perbaikan routing dan pembuatan API baru, serta melakukan revamp pada modul <i>Leave Permit</i> dan penyusunan ulang logika penalti pada kontrak.
Lanjut pada halaman berikutnya	

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

Minggu Ke -	Pekerjaan yang dilakukan
4	Revamp pada modul <i>User Management</i> , <i>Leave Management</i> , dan <i>Permit Detail</i> , termasuk memperbaiki validator autentikasi serta fungsi perhitungan hari untuk meningkatkan konsistensi sistem.
5	Menambah API baru untuk persetujuan cuti, memperbaiki fungsi perhitungan hari, melakukan refaktor pada fungsi pengguna dan paginasi, serta memperbaiki data terkait proyek, produk, dan jabatan.
6	Melakukan pembaruan besar pada laporan pengguna, kontrak, dan API Stand Up, menambahkan detail pada kontrak serta item payroll, memperbaiki relasi model dan API payslip, serta meningkatkan sistem paginasi dan manajemen jenis cuti.
7	Melakukan finalisasi pada API Payslip, memperbaiki bug pada People Report, menambahkan fitur paginasi dan penyortiran, melakukan migrasi baru untuk Payslip Details, serta memperbaiki validator sistem.
8	Melakukan pengembangan fitur ekspor data ke format XLSX, perbaikan API pembuatan pengguna, perbaikan berkas migrasi dan seeding, serta penambahan sistem perizinan baru pada setiap modul.
9	Melakukan perbaikan pada berbagai modul seperti kontrak, akun, dan pengguna, menambahkan fungsi baru untuk pengelolaan kontrak aktif, serta merencanakan fitur input data klien melalui pembacaan berkas CSV untuk proses onboarding.
10	Melanjutkan pengembangan fitur onboarding klien dengan melanjutkan pengembangan sistem pembacaan berkas CSV, pembuatan tabel serta fungsi Import Mapping, dan perbaikan FileController untuk mendukung proses impor data secara efisien.
Lanjut pada halaman berikutnya	

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

Minggu Ke -	Pekerjaan yang dilakukan
11	Menyelesaikan pengembangan layanan Onboarding dengan menerapkan proses parsing data pada setiap modul, sehingga sistem impor data klien dapat berjalan secara otomatis dan terintegrasi dengan baik.
12	Menambahkan fitur penanganan kesalahan pada fungsi Onboarding untuk proses unggah dan impor berkas CSV, serta memungkinkan pengguna untuk mengimpor bagian modul secara bertahap guna meningkatkan fleksibilitas dan keandalan sistem.

3.4 Pengumpulan dan Analisis Kebutuhan

Kebutuhan sistem dalam proyek ini diperoleh melalui koordinasi langsung dengan *Software Engineer Manager*. Sebagian besar *requirement* ditentukan secara iteratif berdasarkan kebutuhan bisnis dan *sprint* mingguan yang telah direncanakan oleh tim. Proses pengumpulan *requirement* dilakukan melalui diskusi teknis mingguan di *retrospective meeting*. Setiap *requirement* yang diidentifikasi kemudian dianalisis untuk menentukan cakupan fungsionalitas, prioritas implementasi, serta dampaknya terhadap sistem yang ada. Hasil analisis ini menjadi dasar dalam perancangan dan pengembangan fitur-fitur baru yang sesuai dengan kebutuhan pengguna dan tujuan bisnis perusahaan.

Berikut ini adalah uraian *requirement* utama yang berhasil diidentifikasi selama masa kerja praktik:

A Modul *Contract*

Sebagai landasan utama bagi sistem penggajian (*Payroll*), perusahaan membutuhkan mekanisme pengelolaan data kontrak yang terpusat. Sistem harus mampu menangani seluruh siklus hidup kontrak kerja pegawai, mulai dari pembuatan draf hingga pemantauan masa berlaku. Selain itu, modul ini harus memfasilitas validasi status kontrak melalui fitur persetujuan (*approval*),

pengakhiran (*terminate*), serta penghapusan data untuk menjamin fleksibilitas dalam pengelolaan hubungan kerja.

B Modul Payroll Item

Manajemen kompensasi yang dinamis membutuhkan pengelolaan komponen gaji yang terstruktur. Sistem harus mampu mendefinisikan berbagai kategori penghasilan seperti gaji pokok, tunjangan, hingga potongan secara spesifik. Modul ini dirancang untuk memfasilitasi penyesuaian komponen gaji berdasarkan kebijakan perusahaan yang berlaku, sehingga setiap perubahan parameter gaji dapat terpetakan dengan akurat ke dalam sistem.

C Modul Payslip

Proses finalisasi penggajian membutuhkan akurasi tinggi dalam kalkulasi pembayaran. Sistem harus mampu melakukan kalkulasi otomatis berdasarkan integrasi data kontrak dan variabel lainnya untuk menghasilkan slip gaji yang akurat. Modul ini memfasilitasi penyediaan informasi rincian gaji, periode pembayaran, dan catatan historis secara transparan, yang dapat diakses secara mandiri oleh pegawai.

D Modul On Boarding

Untuk menangani ekspansi data klien, operasional perusahaan membutuhkan metode input data yang cepat dan masal. Sistem harus mampu memproses impor data melalui berkas *Comma Separated Values* (CSV) sesuai dengan *template* yang telah ditentukan. Fitur ini memfasilitasi validasi integritas dan konsistensi data secara otomatis, sehingga mempercepat waktu implementasi bagi pengguna baru.

E Modul Overtime

Pencatatan jam kerja tambahan membutuhkan sistem pemantauan yang transparan untuk menghindari kesalahan perhitungan manual. Sistem harus mampu mencatat durasi lembur secara tepat dan mengelola alur persetujuan oleh atasan. Modul ini memfasilitasi pengintegrasian kompensasi lembur ke dalam sistem

penggajian, memastikan setiap jam kerja tambahan pegawai tercatat dan dibayarkan sesuai regulasi perusahaan.

3.5 Perancangan dan Pengembangan Sistem

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

3.5.1 Modul Contract

Modul *Contract* berfungsi sebagai fondasi utama dalam sistem penggajian (*Payroll*) yang dikembangkan. Modul ini mencakup pembuatan, pengelolaan, dan pemantauan kontrak kerja pegawai, termasuk fitur-fitur seperti persetujuan kontrak, pembatalan, serta penghapusan kontrak. Struktur basis data dirancang untuk mendukung berbagai jenis kontrak dan aturan terkait, sehingga memungkinkan fleksibilitas dalam pengelolaan hubungan kerja.

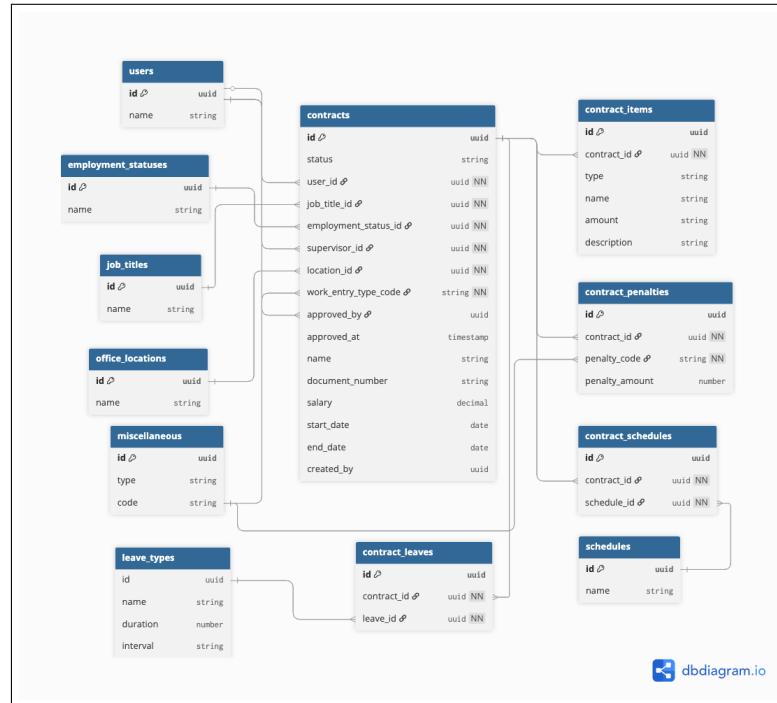
A Diagram ERD Modul *Contract*

Berikut merupakan *database diagram* untuk modul *Contract* yang telah dibuat selama pelaksanaan kerja praktik.

Pada gambar ?? menunjukkan struktur basis data untuk modul *Contract*. Terdapat beberapa tabel utama yang saling berhubungan, yaitu:

- *Contracts*: Tabel ini menyimpan data kontrak kerja pegawai, termasuk informasi seperti tanggal mulai dan berakhir kontrak, status persetujuan, serta referensi ke pegawai yang bersangkutan.
- *Employment Statuses*: Tabel ini menyimpan berbagai status kepegawaian yang dapat dimiliki oleh pegawai, seperti aktif, cuti, atau tidak aktif.
- *Job Titles*: Tabel ini menyimpan informasi mengenai jabatan atau posisi yang dimiliki oleh pegawai dalam perusahaan.
- *Office Location*: Tabel ini menyimpan data lokasi kantor tempat pegawai bekerja, yang dapat digunakan untuk keperluan absensi.

- *Miscellaneous*: Tabel ini menyimpan data tambahan terkait kontrak, seperti jenis kontrak, durasi kerja, dan informasi lainnya yang mendukung pengelolaan kontrak.
- *Schedules*: Tabel ini menyimpan jadwal kerja yang terkait dengan kontrak, termasuk jam kerja dan hari kerja yang ditetapkan.
- *Leave Types*: Tabel ini menyimpan jenis-jenis cuti yang tersedia bagi pegawai, yang dapat dikaitkan dengan kontrak kerja.
- *Contract Items*: Tabel ini menyimpan rincian *item-item* yang termasuk dalam kontrak, seperti gaji pokok, tunjangan, dan potongan.
- *Contract Penalties*: Tabel ini menyimpan data penalti yang dapat dikenakan dalam kontrak kerja, termasuk jenis pelanggaran dan besaran penalti.
- *Contract Schedules*: Tabel ini menyimpan hubungan antara kontrak dan jadwal kerja yang ditetapkan.
- *Contract Leaves*: Tabel ini menyimpan hubungan antara kontrak dan jenis cuti yang tersedia bagi pegawai.



Gambar 3.2. Diagram ERD untuk modul *Contract*

B Contract API Endpoints

Berikut adalah daftar *endpoints* API yang telah dikembangkan untuk modul *Contract*:

Tabel 3.2. Daftar *endpoints* API untuk modul *Contract*

Nama API	Metode	Endpoint	Deskripsi
Get All User's Contracts	GET	/contract/user	Mengambil semua kontrak milik pengguna
Get User's Contract Detail by ID	GET	/contract/user/:id	Mengambil detail kontrak <i>user</i> berdasarkan ID
Approve Contract by ID	PATCH	/contract/:id/approve	Menyetujui kontrak <i>user</i> berdasarkan ID
Terminate Contract by ID	DELETE	/contract/:id	Mengakhiri kontrak <i>user</i> berdasarkan ID
Delete Contract by ID	DELETE	/contract/:id/delete	Menghapus kontrak <i>user</i> berdasarkan ID

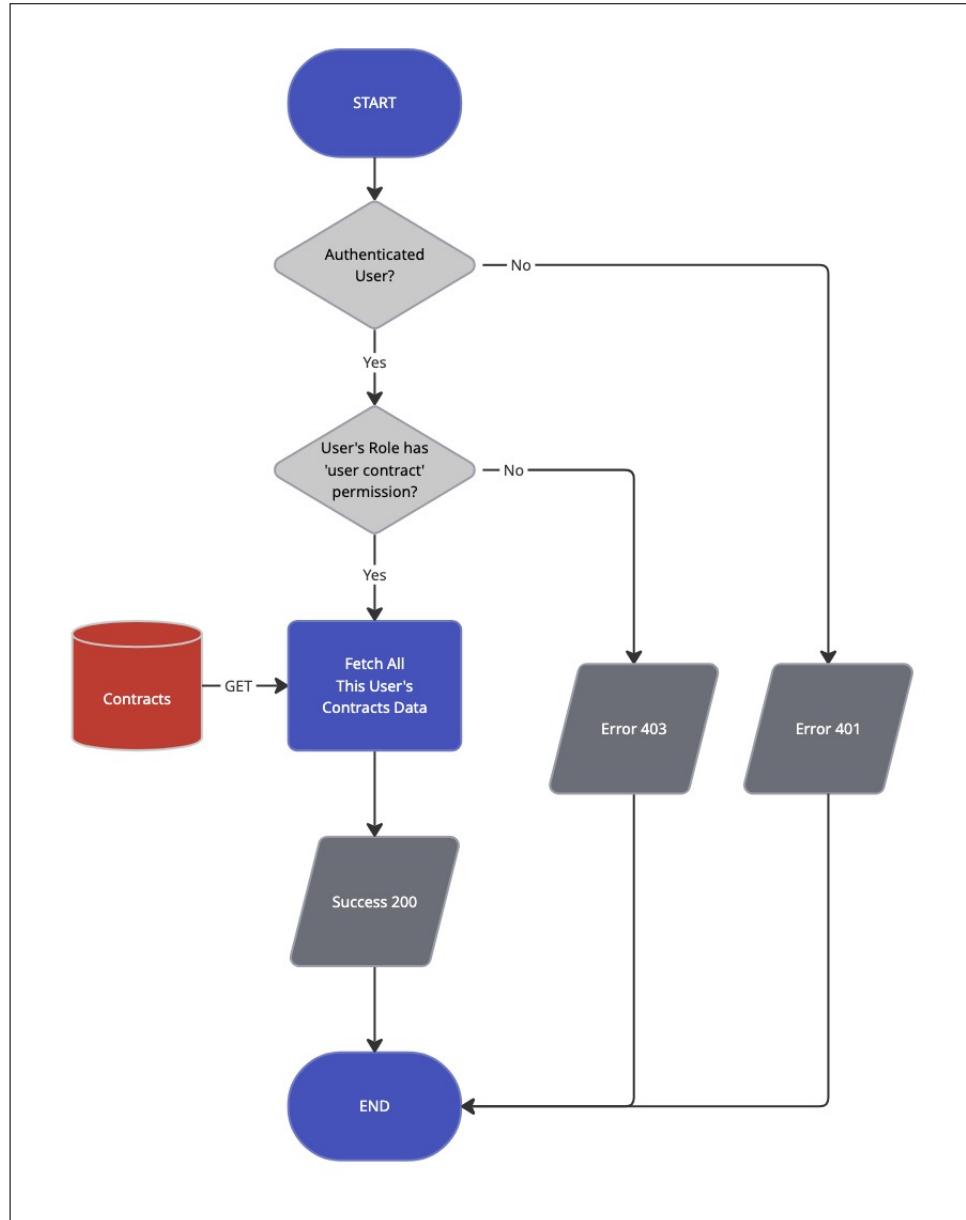
Berikut merupakan daftar *endpoints* API yang telah dikembangkan untuk modul *Contract* seperti pada Tabel ???. Setiap *endpoint* memiliki fungsi spesifik dalam mengelola kontrak kerja pegawai, mulai dari pengambilan data kontrak, persetujuan, hingga penghapusan kontrak. Selanjutnya, akan dijelaskan alur sistem untuk *endpoints* dalam pada modul ini.

Get All User's Contracts

Get All User's Contracts adalah *endpoint* API yang berfungsi untuk mengambil semua kontrak yang dimiliki oleh pengguna tertentu. API ini digunakan di sisi *frontend* untuk menampilkan daftar kontrak yang terkait dengan pengguna yang sedang masuk (*logged in user*) dalam page *Contract*. Gambar ?? dan ?? menunjukkan contoh *request* dan *response* untuk API *Get All User's Contracts*.

Flowchart dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan

memeriksa *permission user contract* untuk mengakses data kontrak. Jika pengguna tidak ada *permission user contract*, sistem akan mengembalikan *error 403*. Setelah itu, sistem mengambil data kontrak dari basis data berdasarkan ID pengguna yang terautentifikasi. Data kontrak yang diambil kemudian dikemas dalam format JSON dan dikirimkan kembali sebagai respons kepada pengguna.



Gambar 3.3. Flowchart alur sistem API Get All User's Contracts

Berikut merupakan contoh *request* dan *response* untuk API *Get All User's Contracts*. Pada gambar ?? menunjukkan contoh *request* yang dikirimkan ke *endpoint* /contract/user menggunakan metode HTTP *GET* dengan menyertakan

query parameters untuk *pagination*.

The screenshot shows the 'Get All User's Contracts' API endpoint. It includes a 'Request' section with fields for Authorization, pagination, page, row, sort, search, and filter[].

Field	Type	Description
pagination	boolean	required
page	integer	required
row	integer	required
sort	string	optional
search	string	optional
filter[]	string	optional

Gambar 3.4. Contoh *request* API *Get All User's Contracts*

Pada gambar ?? menunjukkan contoh *response* yang diterima dari *server*, berisi daftar kontrak yang dimiliki oleh pengguna beserta informasi terkait seperti *contract ID*, nama kontrak, status kontrak, *start date*, *end date*, nama karyawan, pekerjaan karyawan, *employment status* karyawan, dan karyawan yang membuat kontrak tersebut.

```
{
  "code": 200,
  "message": "Success",
  "data": {
    "count": 1,
    "rows": [
      {
        "id": "45f7d2c6-4fe1-46b4-935b-71eb0d779bcf",
        "status": "ON-GOING",
        "name": "Staff 1 Contract",
        "start_date": "2024-12-12T00:00:00.000Z",
        "end_date": "2026-12-12T00:00:00.000Z",
        "created_at": "2025-12-17T10:27:43.221Z",
        "updated_at": "2025-12-17T10:27:43.221Z",
        "is_lunch": true,
        "employee_name": {
          "id": "a02f1296-8265-43f5-81e5-2dbad43a1180",
          "fullname": "Staff 1"
        },
        "job_title": {
          "id": "641b1453-2b50-4d6d-a3c9-65faf617e4db",
          "name": "Marketing"
        },
        "employment_status": {
          "id": "45794ced-4ff4-4ea3-9419-d9657cfa9f1d",
          "name": "Full Time"
        },
        "created": {
          "id": "b3f5713e-2fc4-4e63-8f3c-69dab2341633",
          "fullname": "Supervisor 1"
        }
      }
    ]
  }
}
```

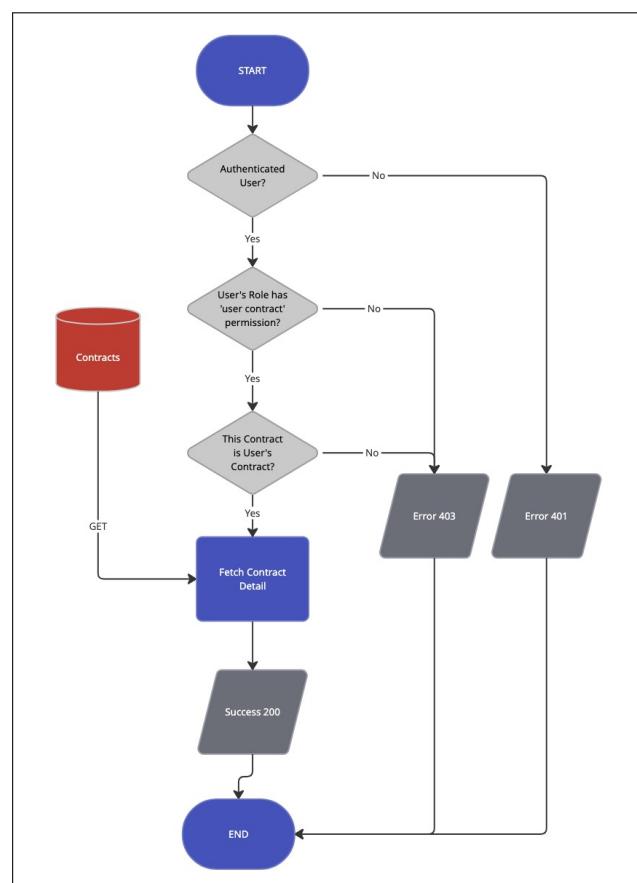
Gambar 3.5. Contoh *response API Get All User's Contracts*

Get User's Contract Detail by ID

Get User's Contract Detail by ID adalah *endpoint API* yang berfungsi untuk mengambil detail kontrak milik pengguna berdasarkan *ID* kontrak yang diberikan. API ini digunakan di sisi *frontend* untuk menampilkan informasi rinci mengenai kontrak tertentu dalam page *Contract Detail*.

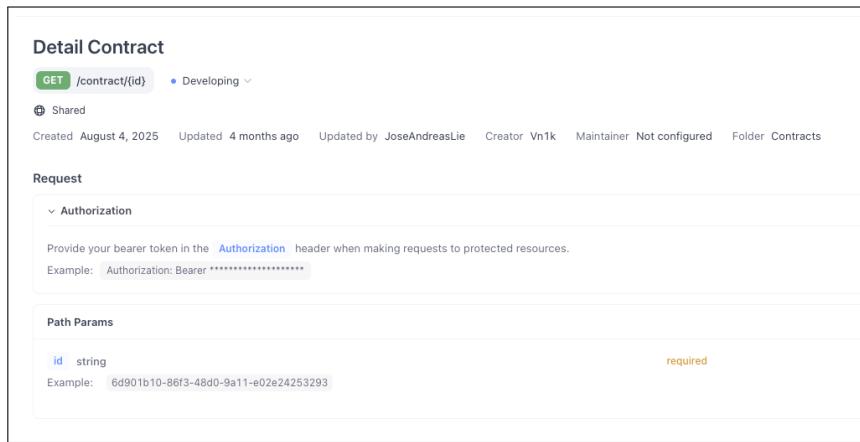
Pada gambar ?? merupakan *flowchart API Get User's Contract Detail by ID*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan memeriksa *permission user contract* untuk mengakses data kontrak. Jika pengguna tidak ada *permission user contract*, sistem akan mengembalikan *error 403*. Setelah

itu, sistem akan memeriksa apabila pemilik kontrak yang diminta tidak sama dengan pengguna yang melakukan permintaan. Jika tidak, sistem akan mengembalikan *error 403*. Setelah itu, sistem mengambil data detail kontrak dari basis data berdasarkan *ID* kontrak yang diberikan dari *path parameters*, *contract_id*. Data detail kontrak yang diambil kemudian dikemas dalam format JSON dan dikirimkan kembali sebagai respons kepada pengguna.



Gambar 3.6. Flowchart alur sistem API *Get User's Contract Detail by ID*

Pada gambar ?? menunjukkan contoh *request* yang dikirimkan ke *endpoint* /contract/user/:id menggunakan metode HTTP *GET* dengan menyertakan *path parameters*, *contract_id*, untuk menentukan kontrak yang ingin diambil detailnya.



Gambar 3.7. Contoh *request* API *Get User's Contract Detail by ID*

Pada gambar ?? menunjukkan contoh *response* yang diterima dari *server*, berisi informasi rinci mengenai kontrak tertentu, termasuk data kontrak, data pegawai terkait, jadwal kerja, jenis cuti, rincian komponen gaji kontrak (*contract items*), serta aturan penalti yang berlaku dalam kontrak tersebut.

Berikut merupakan contoh *response* untuk API *Get User's Contract Detail by ID*.

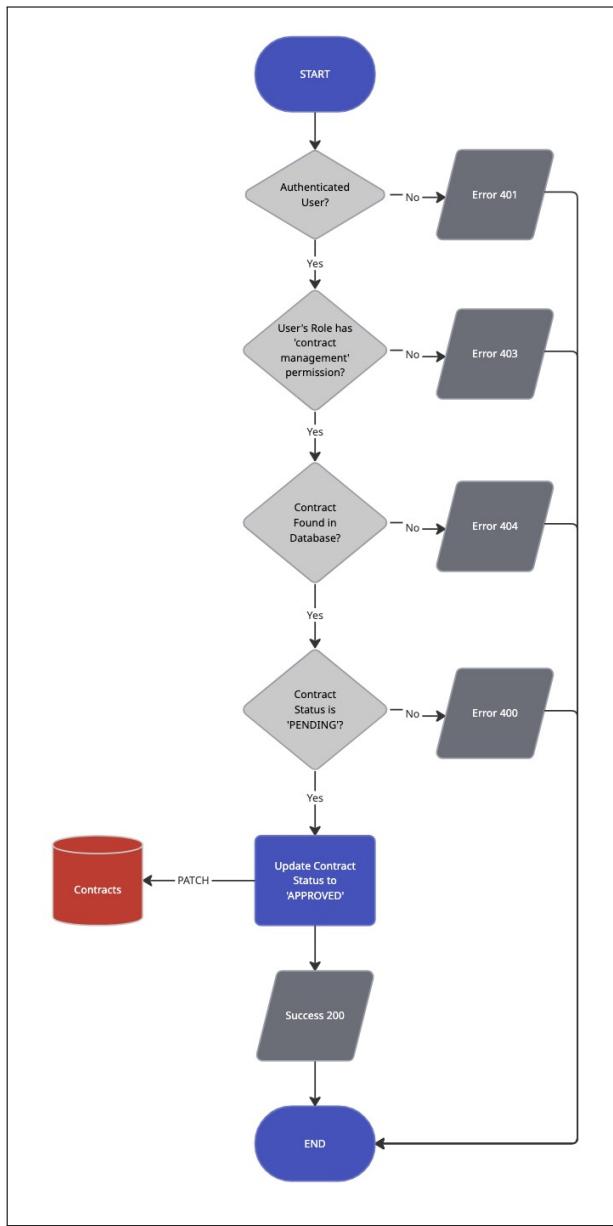
```
{
  "code": 200,
  "message": "Success",
  "data": {
    "staff": {
      "staff_id": "001-G03Hc",
      "employee_name": "C",
      "id": "a021296-2265-43f5-81e5-2dbad43a1180",
      "fullname": "Staff 1"
    },
    "name": "Staff 1 Contract",
    "job_title": {
      "id": "641b1453-2b58-4d6d-a3c9-65faf617e4db",
      "name": "Marketing"
    },
    "employment_status": {
      "id": "45794ced-ffff-4ea3-9419-d9657cf9f1d",
      "name": "Full Time"
    },
    "supervisor": {
      "id": "b3f5713e-2fca-4e63-8f3c-69dab2341633",
      "fullname": "Supervisor 1"
    },
    "location": {
      "id": "abd8144-e9d1-4184-b5a3-379c47faf43f",
      "name": "Tangerang Office (GDP)",
      "address": "Jl. Grand Boulevard, BSD Green Office Park 9, BSD City, Sampora, Kec. Cisauk, Kabupaten Tangerang, Banten 15345"
    },
    "start_date": "2024-12-12T00:00:00.000Z",
    "end_date": "2026-12-12T00:00:00.000Z",
    "leave_type": [
      {
        "id": "8cc1bed6-8ef7-4a52-b004-90fc3ff57745",
        "name": "Annual"
      }
    ],
    "work_entry_type": {
      "id": 1,
      "type": "schedule",
      "code": "FIXED",
      "value": "Fixed",
      "description": ""
    },
    "schedule": [
      {
        "id": "480c4205-d8c4-4d92-9e25-f266a06f6033",
        "name": "Senin - Jumat (09:00 - 18:00)",
        "shift_name": null,
        "code": null,
        "sort_order": null
      },
      {
        "id": "b9b2c7ad-1d6f-47c8-a1ef-9e33e8d578f",
        "name": "Lunch Time (12:00 - 13:00)",
        "shift_name": null
      }
    ]
  }
}
```

Gambar 3.8. Contoh *response* API *Get User's Contract Detail by ID*

Approve Contract by ID

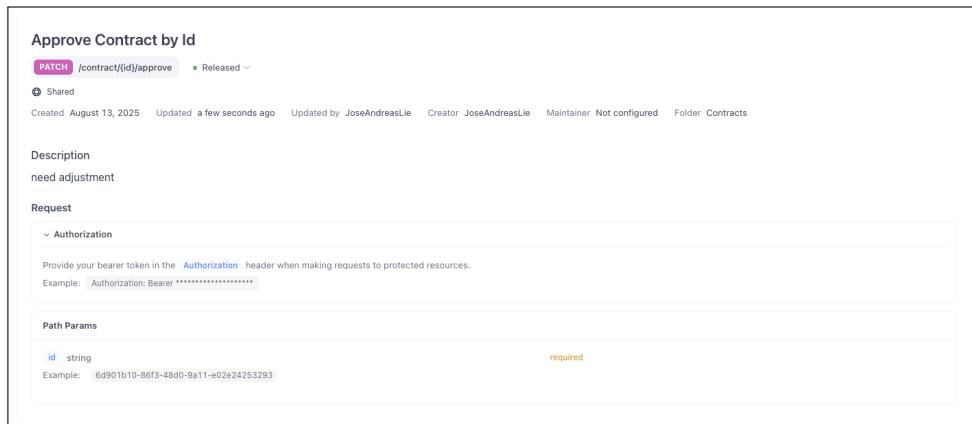
Approve Contract by ID adalah *endpoint API* yang berfungsi untuk menyetujui kontrak milik pengguna berdasarkan *ID* kontrak yang diberikan. API ini digunakan di sisi *frontend* untuk mengubah status kontrak menjadi disetujui (*Approved*) dalam page *Contract Detail*.

Pada gambar ?? merupakan *flowchart API Approve Contract by ID*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan memeriksa *permission contract management* untuk menyetujui kontrak. Jika pengguna tidak ada *permission contract management*, sistem akan mengembalikan *error 403*. Setelah itu, sistem akan mencari kontrak berdasarkan *ID* yang diberikan pada *request path parameters*. Jika kontrak tidak ditemukan, sistem akan mengembalikan *error 404*. Lalu, kontrak yang ditemukan akan diperiksa status kontrak. Apabila status kontrak bukan *Pending*, sistem akan mengembalikan *error 400*. Setelah sistem memperbarui status kontrak menjadi disetujui (*Approved*) berdasarkan *ID* yang diberikan dari *path parameters*, *contract_id*. Setelah status kontrak berhasil diperbarui, sistem mengembalikan *success 200* dan menyatakan bahwa kontrak berhasil di-



Gambar 3.9. Flowchart alur sistem API Approve Contract by ID

Pada gambar ?? merupakan contoh *request* untuk API *Approve Contract by ID*. *Request* dikirimkan ke *endpoint* /contract/id/approve dengan metode HTTP *PATCH* dengan menyertakan *path parameters*, *contract_id* untuk menentukan kontrak yang ingin dimodifikasi.



Gambar 3.10. Contoh *request* API *Approve Contract by ID*

Pada gambar ?? merupakan contoh *response* untuk API *Approve Contract by ID*. *Response* yang diberikan hanya berupa *status code* 200 dengan pesan sukses.



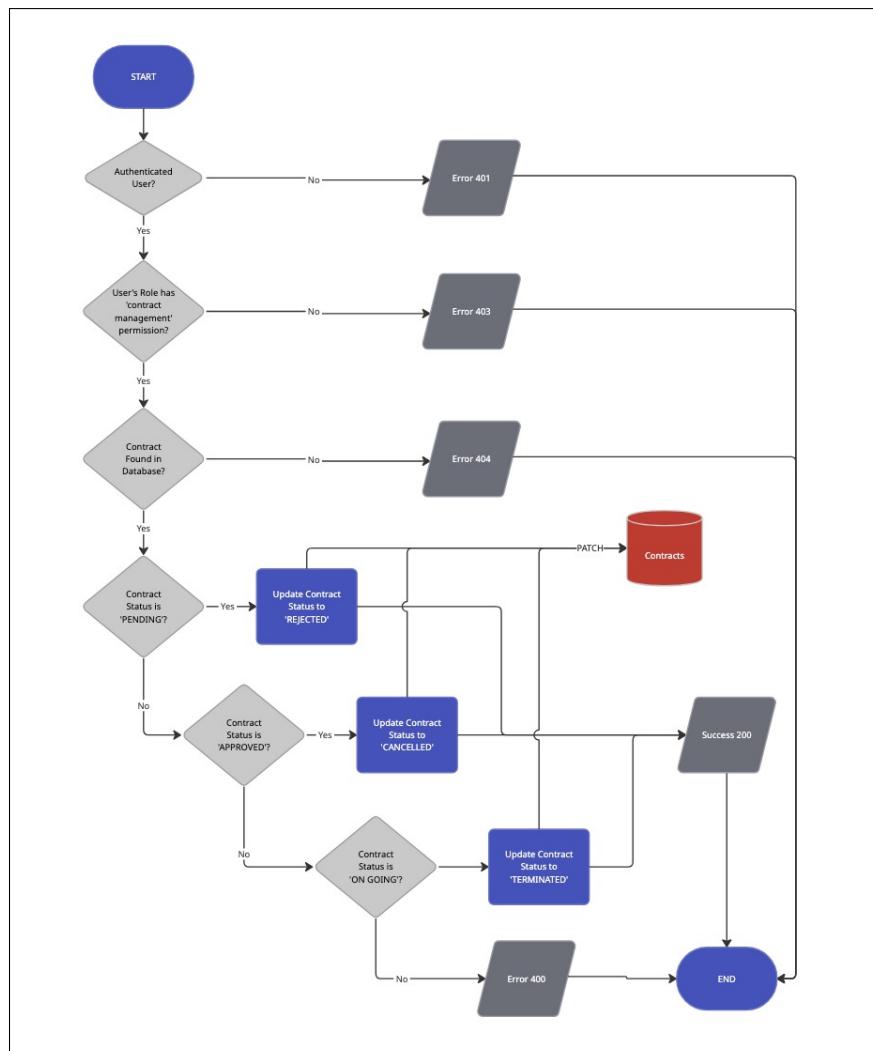
Gambar 3.11. Contoh *response* API *Approve Contract by ID*

Terminate Contract by ID

Terminate Contract by ID adalah *endpoint* API yang berfungsi untuk mengakhiri kontrak milik pengguna berdasarkan *ID* kontrak yang diberikan. API ini digunakan di sisi *frontend* untuk mengubah status kontrak menjadi dihentikan (*Terminated*) dalam page *Contract Detail*.

Pada gambar ?? merupakan *flowchart* API *Terminate Contract by ID*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error* 401. Jika autentikasi berhasil, sistem akan memeriksa *permission contract management* untuk mengakhiri kontrak. Jika pengguna tidak ada *permission contract management*, sistem akan mengembalikan *error* 403. Setelah itu, sistem akan mencari kontrak berdasarkan *ID* yang diberikan pada *request path parameters*. Jika kontrak tidak ditemukan, sistem akan mengembalikan *error* 404. Lalu, kontrak yang ditemukan akan diperiksa status kontrak. Apabila status kontrak tidak sesuai dengan ketentuan, sistem akan mengembalikan *error* 400. Setelah itu, sistem akan memeriksa apa status

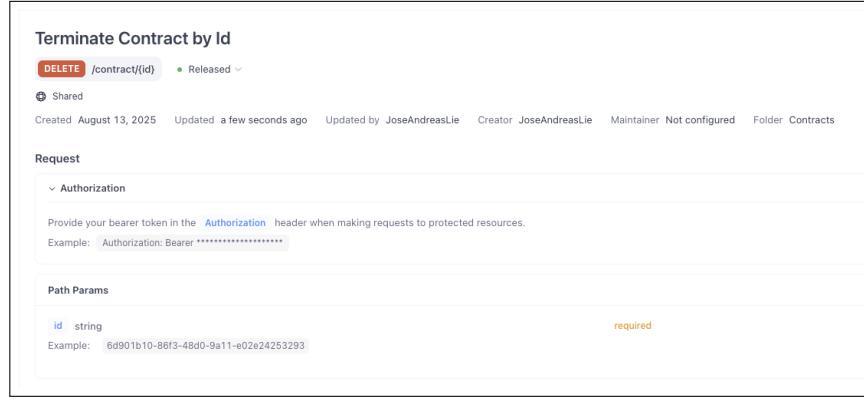
kontrak sekarang. Kalau status kontak sekarang adalah *Pending*, maka status kontrak akan diperbarui menjadi *Rejected* mengindikasikan bahwa kontrak ditolak. Jika status kontrak sekarang adalah *Approved*, status kontrak diperbarui menjadi *Cancelled* mengindikasikan bahwa kontrak sudah diakhiri sebelum dimulai. Jika status kontrak sekarang adalah *On Going*, status kontrak akan diperbarui menjadi *Terminated* mengindikasikan bahwa kontrak sempat aktif lalu diakhiri. Jika status kontrak tidak ada yang sesuai dengan yang diatas, sistem akan mengembalikan *error 400*. Setelah status kontrak berhasil diperbarui, sistem mengembalikan respons (*response*) kepada pengguna yang berisi informasi kontrak yang telah dihentikan.



Gambar 3.12. Flowchart alur sistem API *Terminate Contract by ID*

Pada gambar ?? dan ?? menunjukkan contoh *request* dan *response*

untuk API *Terminate Contract by ID*. Request dikirimkan ke endpoint /contract/{id} dengan metode HTTP *DELETE* dengan menyertakan path parameters, contract_id untuk menentukan kontrak yang ingin diakhiri.



Gambar 3.13. Contoh *request* API *Terminate Contract by ID*

Pada gambar ?? merupakan contoh *response* untuk API *Terminate Contract by ID*. Response yang diberikan berisi *status code* 200 dengan pesan sukses mengindikasikan bahwa kontrak berhasil dihentikan.



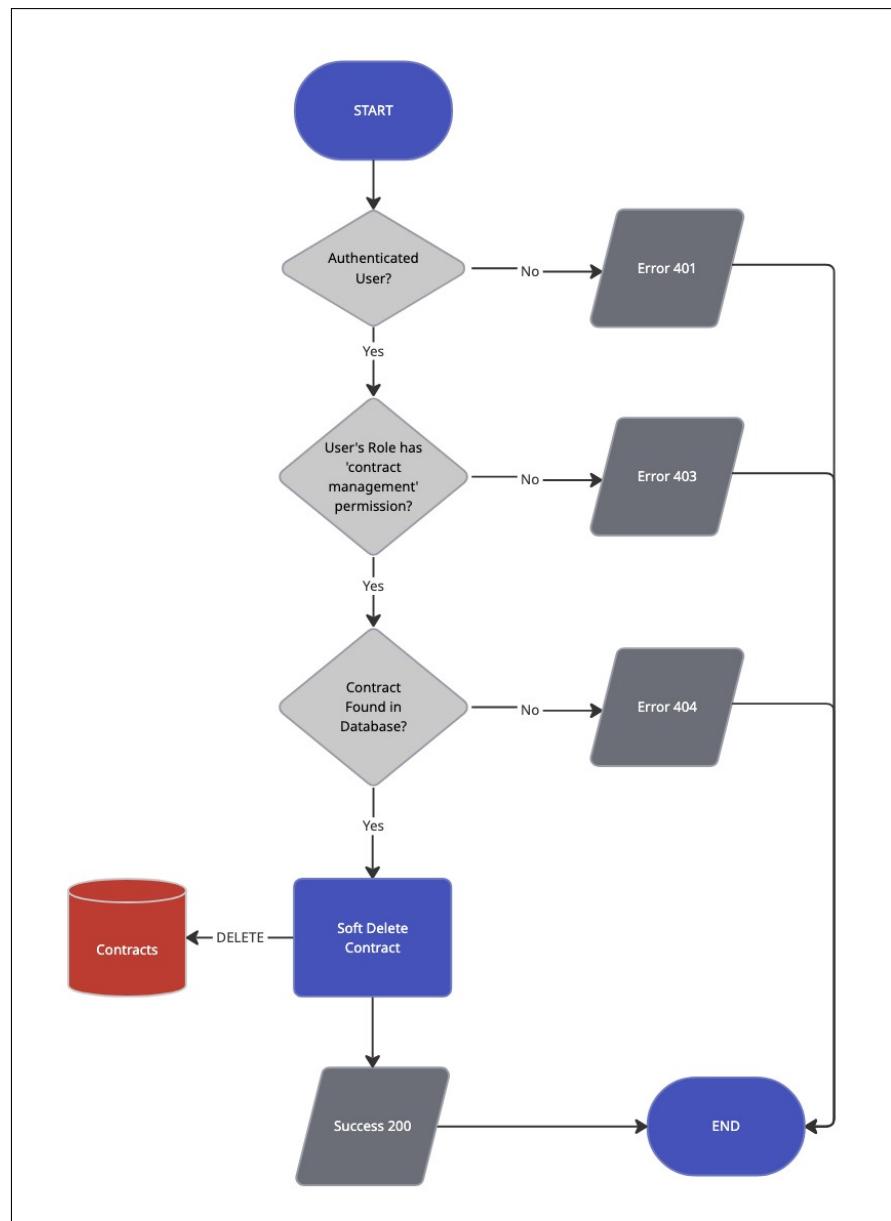
Gambar 3.14. Contoh *response* API *Terminate Contract by ID*

Delete Contract by ID

Delete Contract by ID adalah endpoint API yang berfungsi untuk menghapus kontrak dari sistem berdasarkan *ID* kontrak yang diberikan. API ini digunakan di sisi *frontend* untuk menghapus data kontrak dalam page *Contract Detail*.

Pada gambar ?? merupakan *flowchart* API *Delete Contract by ID*. Flowchart dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error* 401. Jika autentikasi berhasil, sistem akan memeriksa *permission contract management* untuk menghapus kontrak. Jika pengguna tidak ada *permission contract management*, sistem akan mengembalikan *error*

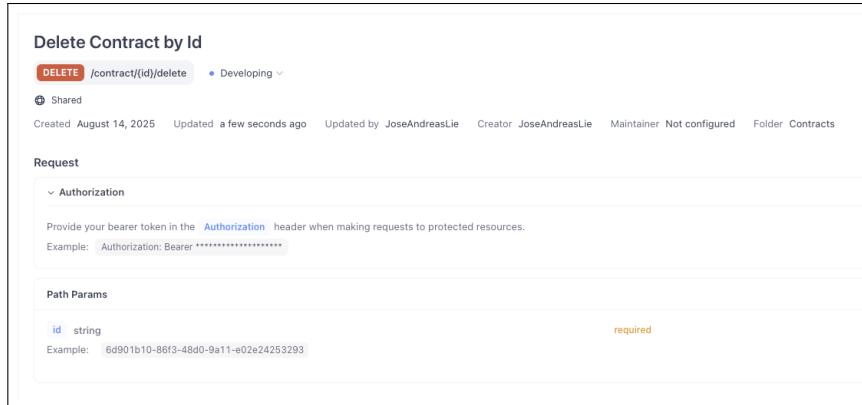
403. Setelah itu, sistem akan mencari kontrak berdasarkan *ID* yang diberikan pada *request path parameters*. Jika kontrak tidak ditemukan, sistem akan mengembalikan *error 404*. Setelah itu, sistem menghapus data kontrak berdasarkan *ID* yang diberikan dari *path parameters*, *contract_id*. Setelah data kontrak berhasil dihapus, sistem mengembalikan respons (*response*) kepada pengguna yang berisi informasi keberhasilan penghapusan kontrak.



Gambar 3.15. Flowchart alur sistem API *Delete Contract by ID*

Pada gambar ?? menunjukkan contoh *request* untuk API *Delete Contract by ID* yang dikirimkan ke *endpoint* /contract/*id*/delete menggunakan metode

HTTP *DELETE* dengan menyertakan *path parameters*, `contract_id`, untuk menentukan kontrak yang ingin dihapus.



Gambar 3.16. Contoh *request* API *Delete Contract by ID*

Pada gambar ?? merupakan contoh *response* untuk API *Delete Contract by ID*. *Response* yang diberikan berisi *status code* 200 dengan pesan sukses mengindikasikan bahwa kontrak berhasil dihapus.



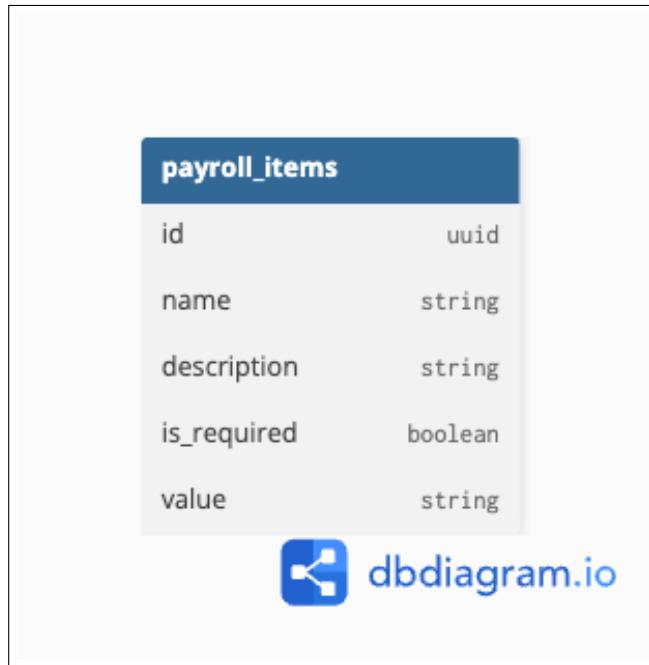
Gambar 3.17. Contoh *response* API *Delete Contract by ID*

3.5.2 Modul Payroll Item

Modul Payroll Item dibuat untuk mengelola berbagai komponen gaji yang akan digunakan dalam perhitungan gaji pegawai. Modul ini mencakup fitur pembuatan, pengelolaan, dan penghapusan item gaji seperti gaji pokok, tunjangan, dan potongan. Struktur basis data dirancang untuk menyimpan informasi terkait item gaji, termasuk jenis item, besaran, serta aturan penerapannya. Dengan adanya modul ini, perusahaan

A Diagram ERD Modul *Payroll Item*

Berikut merupakan *database diagram* untuk modul *Payroll Item* yang telah dibuat selama pelaksanaan kerja praktik. Gambar ?? menunjukkan struktur basis data untuk modul *Payroll Item*.



Gambar 3.18. Diagram ERD untuk modul *Payroll Item*

B *Payroll Item API Endpoints*

Berikut adalah daftar *endpoints* API yang telah dikembangkan untuk modul *Payroll Item*:

Tabel 3.3. Daftar *endpoints* API untuk modul *Payroll Item*

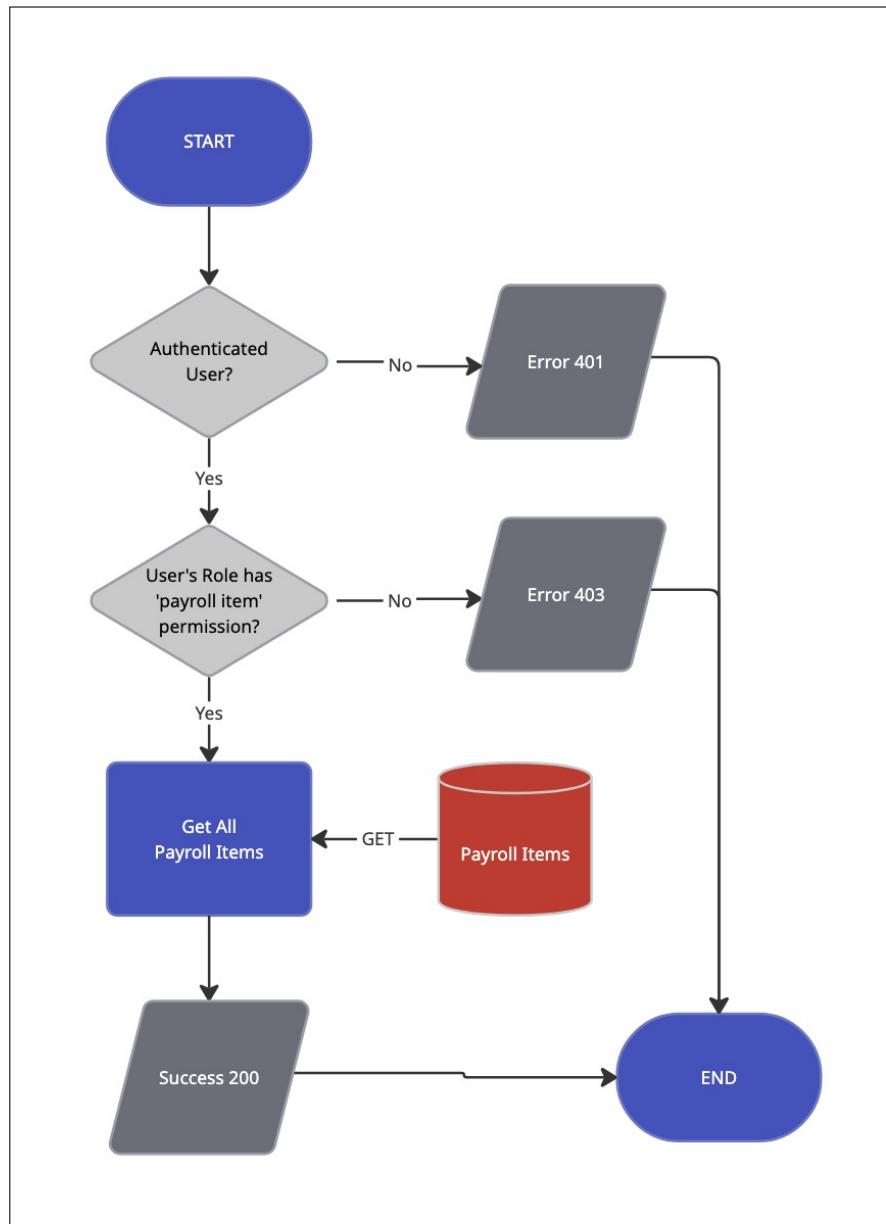
Nama API	Metode	Endpoint	Deskripsi
Get All Payroll Item	GET	/payroll-items	Mengambil semua komponen gaji (<i>Payroll Item</i>)
Get Payroll Item by ID	GET	/payroll-items/:id	Mengambil detail komponen gaji berdasarkan ID
Create Payroll Item	POST	/payroll-items	Membuat komponen gaji baru
Update Payroll Item	PUT	/payroll-items/:id	Memperbarui komponen gaji berdasarkan ID
Delete Payroll Item	DELETE	/payroll-items/:id	Menghapus komponen gaji berdasarkan ID

Berikut merupakan daftar *endpoints* API yang telah dikembangkan untuk modul *Payroll Item* seperti pada Tabel ???. Setiap *endpoint* memiliki fungsi spesifik untuk mengelola komponen gaji (*Payroll Item*). Selanjutnya, akan dijelaskan alur sistem untuk *endpoints* dalam pada modul ini.

Get All Payroll Item

Get All Payroll Item adalah *endpoint* API yang berfungsi untuk mengambil semua komponen gaji (*Payroll Item*) yang tersedia dalam sistem. API ini digunakan di sisi *frontend* untuk menampilkan daftar komponen gaji dalam page *Payroll Items*.

Berikut merupakan *flowchart* API *Get All Payroll Item*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error* 401. Jika autentikasi berhasil, sistem akan memeriksa *permission payroll item* untuk mengakses data komponen gaji. Jika pengguna tidak ada *permission payroll item*, sistem akan mengembalikan *error* 403. Setelah itu, sistem mengambil semua data komponen gaji dari basis data. Data komponen gaji yang diambil kemudian dikemas dalam format JSON dan dikirimkan kembali sebagai respons kepada pengguna.



Gambar 3.19. Flowchart alur sistem API Get All Payroll Item

Pada gambar ?? menunjukkan contoh *request* yang dikirimkan ke *endpoint* /payroll-items menggunakan metode HTTP *GET* dengan menyertakan *query parameters* untuk *pagination* seperti *page*, *row*, *sort*, dan *search*.

The screenshot shows a Swagger API documentation page for a 'GET /payroll-items' endpoint. At the top, there's a green button labeled 'GET /payroll-items' and a status indicator 'Released'. Below that, it says 'Shared' and provides metadata: 'Created August 6, 2025', 'Updated 4 hours ago', 'Updated by JoseAndreasLie', 'Creator JoseAndreasLie', 'Maintainer Not configured', 'Folder Payroll Items'. The main content area has a section titled 'Description' which states 'API to get status data for user attendance and standup'. Under 'Request', there's a subsection for 'Authorization' with instructions to provide a bearer token in the 'Authorization' header. An example is given as 'Authorization: Bearer *****'. The 'Query Params' section lists several parameters: 'pagination' (boolean, required), 'page' (string, required), 'row' (string, required), 'sort' (string, optional), and 'search' (string, optional). Examples for each are provided.

Gambar 3.20. Contoh *request* API *Get All Payroll Item*

Pada gambar ?? menunjukkan contoh *response* untuk API *Get All Payroll Item* yang diterima dari *server*, berisi daftar komponen gaji beserta informasi terkait seperti *payroll item ID*, nama komponen gaji, jenis komponen gaji, besaran komponen gaji, serta status komponen gaji (aktif atau tidak aktif).

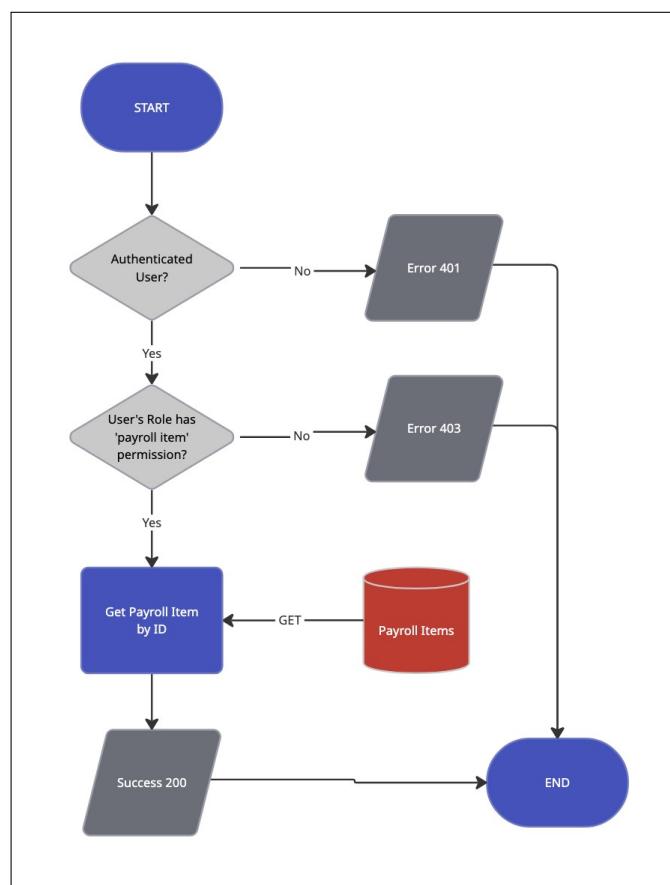
```
{
  "code": 200,
  "message": "Success",
  "data": {
    "count": 4,
    "rows": [
      {
        "id": "64ba20fd-596a-40f4-82cc-01afab381b71",
        "name": "Tunjangan Makan",
        "is_required": false,
        "value": null,
        "description": "Tunjangan yang diberikan untuk makan karyawan.",
        "created_at": "2025-12-17T04:26:52.630Z",
        "updated_at": "2025-12-17T04:26:52.630Z"
      },
      {
        "id": "aa5b9675-b608-487a-851c-4e9b55b170f7",
        "name": "Tunjangan BPJS",
        "is_required": false,
        "value": null,
        "description": "Tunjangan yang diberikan untuk BPJS karyawan.",
        "created_at": "2025-12-17T04:26:52.630Z",
        "updated_at": "2025-12-17T04:26:52.630Z"
      },
      {
        "id": "7cf74dbd-b758-4304-8c6c-1b46c52977e7",
        "name": "Tunjangan Kehormatan",
        "is_required": false,
        "value": null,
        "description": "Tunjangan yang diberikan untuk jabatan karyawan.",
        "created_at": "2025-12-17T04:26:52.630Z",
        "updated_at": "2025-12-17T04:26:52.630Z"
      },
      {
        "id": "6821d814-508c-436f-a232-423bae9997ed",
        "name": "Main Salary",
        "is_required": true,
        "value": "main salary",
        "description": "Gaji Pokok",
        "created_at": "2025-12-17T04:07:19.539Z",
        "updated_at": "2025-12-17T04:07:19.539Z"
      }
    ]
  }
}
```

Gambar 3.21. Contoh *response API Get All Payroll Item*

Get Payroll Item by ID

Get Payroll Item by ID adalah *endpoint API* yang berfungsi untuk mengambil detail komponen gaji (*Payroll Item*) berdasarkan *ID* komponen gaji yang diberikan. API ini digunakan di sisi *frontend* untuk menampilkan informasi rinci mengenai komponen gaji tertentu dalam page *Payroll Item Detail*.

Gambar ?? merupakan *flowchart* API *Get Payroll Item by ID*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan memeriksa *permission payroll item* untuk mengakses data komponen gaji. Jika pengguna tidak ada *permission payroll item*, sistem akan mengembalikan *error 403*. Setelah itu, sistem mengambil data komponen gaji berdasarkan *ID* yang diberikan dari *path parameters*, *payroll_item_id*. Data komponen gaji yang diambil kemudian dikemas dalam format JSON dan dikirimkan kembali sebagai respons kepada pengguna.



Gambar 3.22. *Flowchart* alur sistem API *Get Payroll Item by ID*

Pada gambar ?? menunjukkan contoh *request* yang dikirimkan ke *endpoint* /*payroll-items*/*:id* menggunakan metode HTTP *GET* dengan menyertakan *path parameters*, *payroll_item_id*, untuk menentukan komponen gaji yang ingin diambil detailnya.

The screenshot shows the API documentation for the 'Get Payroll Item by Id' endpoint. It includes the method (GET), URL (/payroll-items/{id}), status (Released), sharing information (Shared), and metadata (Created August 7, 2025, Updated 4 months ago, Updated by JoseAndreasLie, Creator JoseAndreasLie, Maintainer Not configured, Folder Payroll Items). The 'Description' section states it's an API to get status data for user attendance and standup. The 'Request' section includes an 'Authorization' field (example: Authorization: Bearer *****) and a 'Path Params' section with an 'id' parameter (string, required, example: 7cf74dbd-b758-4304-8c6c-1b46c52977e7).

Gambar 3.23. Contoh *request* API Get Payroll Item by ID

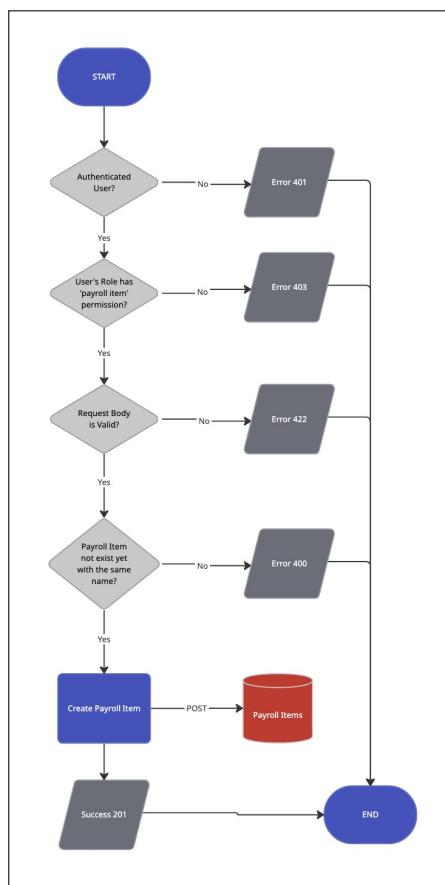
Pada gambar ?? menunjukkan contoh *response* yang diterima dari *server*, berisi informasi rinci mengenai komponen gaji tertentu, termasuk nama komponen gaji, jenis komponen gaji, besaran komponen gaji, serta status komponen gaji (aktif atau tidak aktif).

```
{
  "code": 200,
  "message": "Success",
  "data": {
    "id": "7cf74dbd-b758-4304-8c6c-1b46c52977e7",
    "name": "Tunjangan Kehormatan",
    "is_required": false,
    "value": null,
    "description": "Tunjangan yang diberikan untuk jabatan karyawan.",
    "created_at": "2025-12-17T04:26:52.630Z",
    "updated_at": "2025-12-17T04:26:52.630Z",
    "deleted_at": null
  }
}
```

Gambar 3.24. Contoh *response* API Get Payroll Item by ID

Create Payroll Item

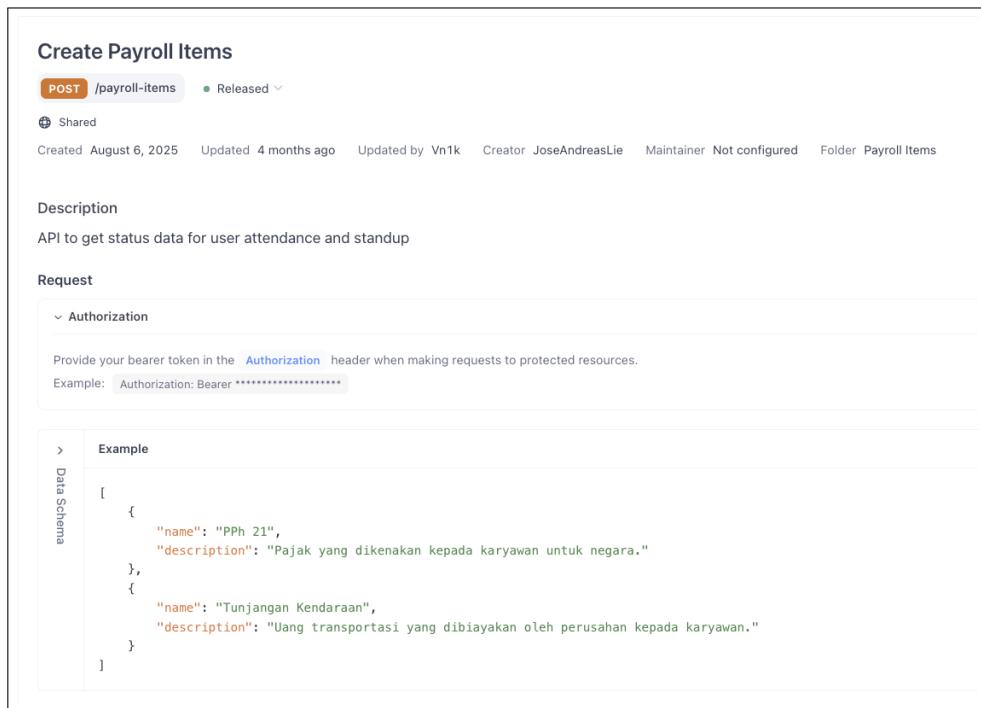
Gambar ?? merupakan *flowchart* API *Create Payroll Item*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan memeriksa *permission payroll item* untuk membuat komponen gaji baru. Jika pengguna tidak ada *permission payroll item*, sistem akan mengembalikan *error 403*. Setelah itu, sistem memeriksa apakah *request body* valid sesuai dengan ketentuan. Jika *request body* tidak valid, sistem akan mengembalikan *error 400*. Jika valid, sistem membuat data komponen gaji baru berdasarkan data yang diberikan dalam *request body*. Setelah data komponen gaji berhasil dibuat, sistem mengembalikan respons (*response*) kepada pengguna yang berisi informasi mengenai komponen gaji yang telah dibuat.



Gambar 3.25. *Flowchart* alur sistem API *Create Payroll Item*

Pada gambar ?? menunjukkan contoh *request* untuk API *Create Payroll Item* yang dikirimkan ke *endpoint* /*payroll-items* menggunakan metode HTTP *POST*

dengan menyertakan *request body* yang berisi data komponen gaji baru yang akan dibuat.



Gambar 3.26. Contoh *request API Create Payroll Item*

Pada gambar ?? menunjukkan contoh *response* untuk API *Create Payroll Item* yang diterima dari *server*, berisi informasi mengenai komponen gaji yang telah dibuat, termasuk nama komponen gaji, jenis komponen gaji, besaran komponen gaji, serta status komponen gaji (aktif atau tidak aktif).

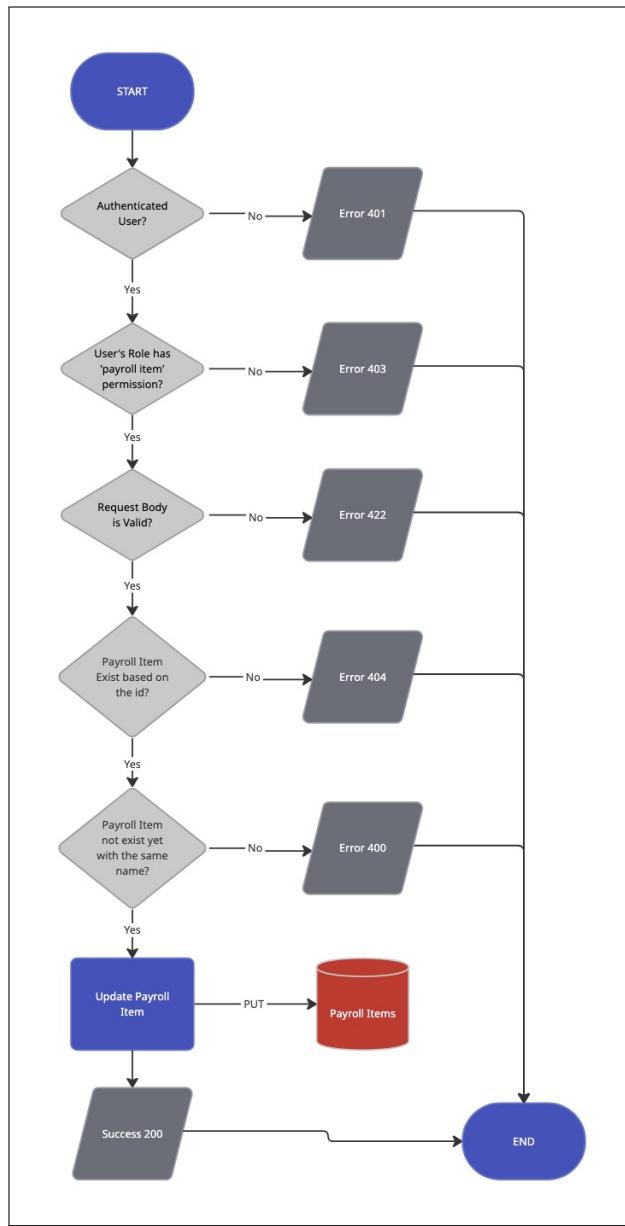
```
{
  "code": 201,
  "message": "Success",
  "data": [
    {
      "id": "5a347071-1985-47a2-bce8-adf509b732d6",
      "is_required": false,
      "name": "PPh 21",
      "description": "Pajak yang dikenakan kepada karyawan untuk negara.",
      "value": "pph 21",
      "created_at": "2025-12-18T08:07:11.178Z",
      "updated_at": "2025-12-18T08:07:11.178Z",
      "deleted_at": null
    },
    {
      "id": "58af5134-a28f-440e-acc2-275cc8cdbeec",
      "is_required": false,
      "name": "Tunjangan Kendaraan",
      "description": "Uang transportasi yang dibayarkan oleh perusahaan kepada karyawan.",
      "value": "tunjangan kendaraan",
      "created_at": "2025-12-18T08:07:11.178Z",
      "updated_at": "2025-12-18T08:07:11.178Z",
      "deleted_at": null
    }
  ]
}
```

Gambar 3.27. Contoh *response API Create Payroll Item*

Update Payroll Item

Update Payroll Item adalah *endpoint API* yang berfungsi untuk memperbarui komponen gaji (*Payroll Item*) berdasarkan *ID* komponen gaji yang diberikan. API ini digunakan di sisi *frontend* untuk mengubah informasi komponen gaji dalam page *Payroll Item Detail*.

Gambar ?? merupakan *flowchart API Update Payroll Item*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan memeriksa *permission payroll item* untuk memperbarui komponen gaji. Jika pengguna tidak ada *permission payroll item*, sistem akan mengembalikan *error 403*. Setelah itu, sistem memeriksa apakah *request body* valid sesuai dengan ketentuan. Jika *request body* tidak valid, sistem akan mengembalikan *error 400*. Jika valid, sistem mencari data komponen gaji dengan menggunakan *ID* yang diberikan dari *Path Parameter* serta memperbarui data komponen gaji data tersebut dengan data yang diberikan dalam *request body*. Setelah data komponen gaji berhasil diperbarui, sistem mengembalikan respons (*response*) kepada pengguna yang berisi informasi mengenai komponen gaji yang telah diperbarui.



Gambar 3.28. Flowchart alur sistem API Update Payroll Item

Pada gambar ?? menunjukkan contoh *request* untuk API *Update Payroll Item* yang dikirimkan ke *endpoint* /payroll-items/:id menggunakan metode HTTP *PUT* dengan menyertakan *path parameters*, payroll_item_id, untuk menentukan komponen gaji yang ingin diperbarui serta *request body* yang berisi data komponen gaji yang akan diperbarui.

Gambar 3.29. Contoh *request* API *Update Payroll Item*

Pada gambar ?? menunjukkan contoh *response* untuk API *Update Payroll Item* yang diterima dari *server*, berisi informasi mengenai komponen gaji yang telah diperbarui, termasuk nama komponen gaji, jenis komponen gaji, besaran komponen gaji, serta status komponen gaji (aktif atau tidak aktif).

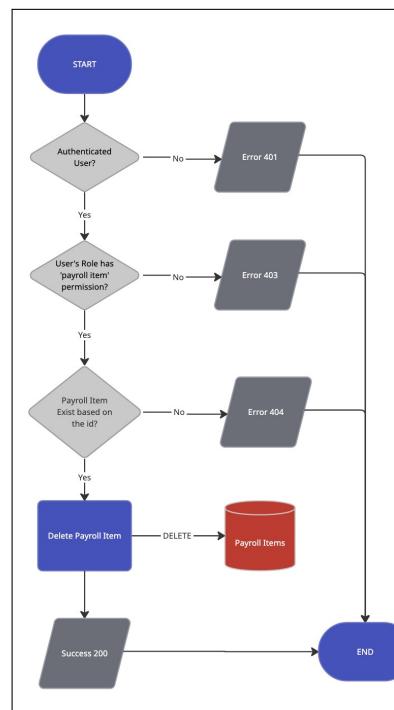
```
{
  "code": 200,
  "message": "Success",
  "data": {
    "id": "5a347071-1985-47a2-bce8-adf509b732d6",
    "name": "PPh 21 (karyawan)",
    "is_required": false,
    "value": "pph 21 (karyawan)",
    "description": "Pajak yang dikenakan kepada karyawan untuk negara (PPh 21).",
    "created_at": "2025-12-18T08:07:11.178Z",
    "updated_at": "2025-12-18T08:14:46.050Z",
    "deleted_at": null
  }
}
```

Gambar 3.30. Contoh *response* API *Update Payroll Item*

Delete Payroll Item

Delete Payroll Item adalah *endpoint API* yang berfungsi untuk menghapus komponen gaji (*Payroll Item*) berdasarkan *ID* komponen gaji yang diberikan. API ini digunakan di sisi *frontend* untuk menghapus data komponen gaji dalam page *Payroll Item Detail*.

Gambar ?? merupakan *flowchart* API *Delete Payroll Item*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan memeriksa *permission payroll item* untuk menghapus komponen gaji. Jika pengguna tidak ada *permission payroll item*, sistem akan mengembalikan *error 403*. Setelah itu, sistem mencari data komponen gaji berdasarkan *ID* yang diberikan dari *path parameters*, *payroll_item_id*. Jika komponen gaji tidak ditemukan, sistem akan mengembalikan *error 404*. Setelah itu, sistem menghapus data komponen gaji berdasarkan *ID* yang diberikan dari *path parameters*, *payroll_item_id*. Setelah data komponen gaji berhasil dihapus, sistem mengembalikan respons (*response*) kepada pengguna yang berisi informasi mengenai keberhasilan penghapusan komponen gaji.



Gambar 3.31. *Flowchart* alur sistem API *Delete Payroll Item*

Pada gambar ?? menunjukkan contoh *request* untuk API *Delete Payroll*

Item yang dikirimkan ke *endpoint* /payroll-items/:id menggunakan metode HTTP *DELETE* dengan menyertakan *path parameters*, payroll_item_id, untuk menentukan komponen gaji yang ingin dihapus.

The screenshot shows the 'Delete Payroll Item' endpoint. It includes a red 'DELETE' button next to the URL /payroll-items/{id}, a status indicator 'Released', and a note that it is 'Shared'. Below this, it shows creation date (August 7, 2025), update history (12 hours ago by JoseAndreasLie), and maintainers (Not configured). The endpoint is categorized under 'Payroll Items'. The 'Description' section states: 'API to get status data for user attendance and standup'. The 'Request' section contains an 'Authorization' field with instructions to provide a bearer token and an example: 'Authorization: Bearer *****'. The 'Path Params' section shows a required parameter 'id' of type string with the value 7cf74dbd-b758-4304-8c6c-1b46c52977e7.

Gambar 3.32. Contoh *request* API *Delete Payroll Item*

Pada gambar ?? menunjukkan contoh *response* untuk API *Delete Payroll Item* yang diterima dari *server*, berisi *status code* 200 dengan pesan sukses untuk mengindikasikan bahwa penghapusan komponen gaji berhasil dilakukan.



Gambar 3.33. Contoh *response* API *Delete Payroll Item*

3.5.3 Modul Payslip

Modul *Payslip* dibuat untuk mengelola dan menghasilkan slip gaji bagi pegawai berdasarkan kontrak yang telah disepakati, dan beberapa. Modul ini mencakup fitur perhitungan gaji, potongan, dan tunjangan sesuai dengan ketentuan yang berlaku dalam kontrak kerja. Struktur basis data dirancang untuk menyimpan informasi terkait gaji, termasuk rincian pembayaran, periode gaji, serta catatan historis. Modul ini juga menyediakan akses untuk menampilkan slip gaji kepada pegawai secara transparan. Modul Payslip juga merupakan langkah terakhir dari sistem penggajian (*Payroll*) yang terintegrasi dengan modul *Contract* dan modul lainnya.

A Diagram ERD Modul *Payslip*

Berikut merupakan *database diagram* untuk modul *Payslip* yang telah dibuat selama pelaksanaan kerja praktik. Gambar ?? menunjukkan struktur basis data untuk modul *Payslip*.



Gambar 3.34. Diagram ERD untuk modul *Payslip*

B Payslip API Endpoints

Berikut adalah daftar *endpoints* API yang telah dikembangkan untuk modul *Payslip*:

Tabel 3.4. Daftar *endpoints* API untuk modul *Payslip*

Nama API	Metode	Endpoint	Deskripsi
Get Payslip Detail	GET	/payslip/date/payslip_date_id /contract/contract_id	Mengambil detail slip gaji <i>user</i>
Save Payslip Detail	PATCH	/payslip/date/payslip_date_id /contract/contract_id	Mengambil detail slip gaji <i>user</i> berdasarkan ID
Get User Payslips List	GET	/payslip/user	Mengambil semua slip gaji milik pengguna
Get User Payslip Detail by ID	GET	/payslip/user/:id	Mengambil detail slip gaji <i>user</i> berdasarkan ID

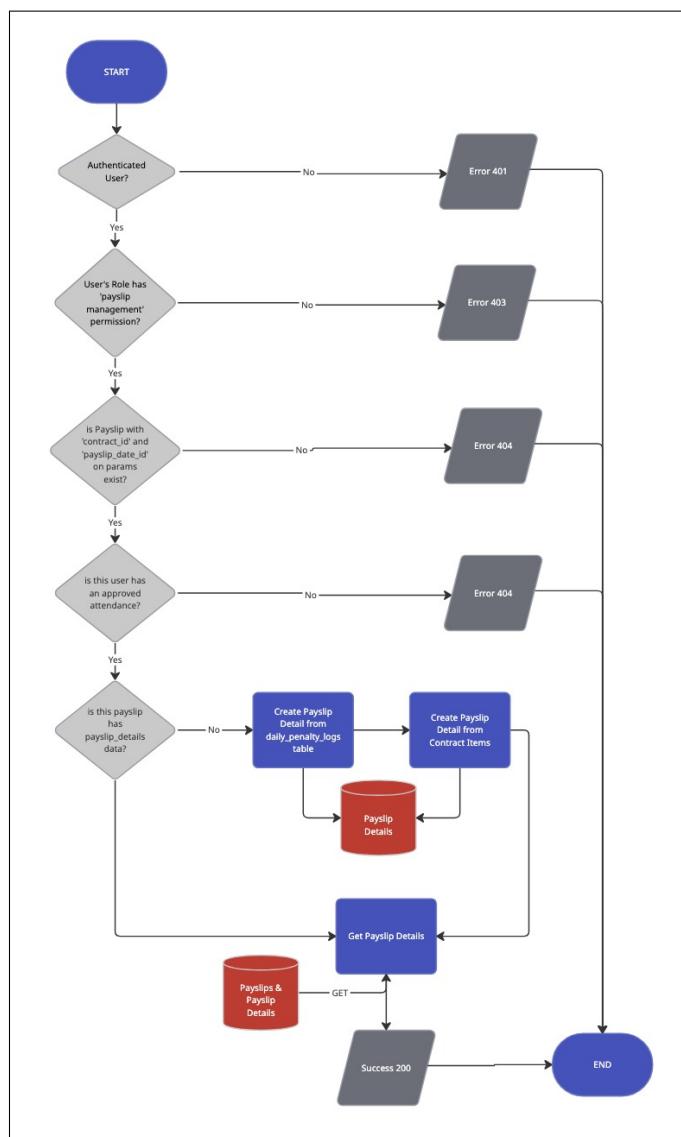
Berikut merupakan daftar *endpoints* API yang telah dikembangkan untuk modul *Payslip* seperti pada Tabel ???. Setiap *endpoint* memiliki fungsi spesifik dalam mengelola slip gaji pegawai, mulai dari pengambilan data slip gaji, hingga penyimpanan detail slip gaji. Selanjutnya, akan dijelaskan alur sistem untuk *endpoints* dalam pada modul ini.

Get Payslip Detail

Get Payslip Detail adalah *endpoint* API yang berfungsi untuk mengambil detail slip gaji (*Payslip Detail*) berdasarkan *payslip_date_id* dan *contract_id* yang diberikan. API ini digunakan di sisi *frontend* untuk menampilkan informasi rinci mengenai slip gaji tertentu dalam page *Payslip Detail*.

Gambar ?? merupakan *flowchart* API *Get Payslip Detail*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error* 401. Jika autentikasi berhasil, sistem akan memeriksa *permission payslip management* untuk mengakses data slip gaji. Jika pengguna tidak ada *permission payslip management*, sistem akan mengembalikan *error*

403. Setelah itu, sistem mengambil data slip gaji berdasarkan *payslip_date_id* dan *contract_id* yang diberikan dari *path parameters*. Jika data slip gaji tidak ditemukan, sistem akan mengembalikan *error 404*. Setelah itu, sistem mengambil data *payslip_details* yang berelasi dengan *payslip* tersebut. Jika data *payslip_details* tidak ditemukan, maka sistem akan membuat data *payslip_details* baru berdasarkan *contract_id* dan *payslip_date_id* yang diberikan. Data akan diambil dari *contract items* untuk komponen gaji yang ada pada kontrak tersebut, dan juga dari *daily_penalty_logs* untuk potongan denda harian yang terjadi pada periode slip gaji tersebut. Setelah data berhasil diambil, sistem mengembalikan respons (*response*) kepada pengguna yang berisi detail slip gaji yang diminta.



Gambar 3.35. Flowchart alur sistem API Get Payslip Detail

Pada gambar ?? menunjukkan contoh *request* untuk API *Get Payslip Detail* yang dikirimkan ke *endpoint* /payslip/date/payslip_date_id/contract/contract_id menggunakan metode HTTP *GET* dengan menyertakan *path parameters*, payslip_date_id dan contract_id, untuk menentukan slip gaji yang ingin diambil detailnya.

The screenshot shows a Swagger API documentation page for the 'Get Payslip Detail' endpoint. At the top, there is a 'GET' button and the URL /payslip/date/{payslip_date_id}/contract/{contract_id}. Below the URL, it says 'Developing' with a dropdown arrow. There is also a 'Shared' icon and some metadata: 'Created September 8, 2025', 'Updated a month ago', 'Updated by JoseAndreasLie', 'Creator JoseAndreasLie', 'Maintainer Not configured', and 'Folder Payslip'. The main content area has sections for 'Description' and 'Request'. The 'Description' section contains the text 'Get Payslip Detail with payslip_date_id and contract_id'. The 'Request' section has a 'Authorization' field with the note 'Provide your bearer token in the Authorization header when making requests to protected resources.' and an example 'Authorization: Bearer *****'. It also lists 'Path Params' with two entries: 'payslip_date_id' (string, required) and 'contract_id' (string, required). Examples for each are provided: 'payslip_date_id' is 'Dafe7fc5-fd7b-4db3-96bc-6b39629ccf51' and 'contract_id' is '9cd78dbb-794c-4fa9-bc8a-8ec09777961b8'.

Gambar 3.36. Contoh *request* API *Get Payslip Detail*

Pada gambar ?? menunjukkan contoh *response* untuk API *Get Payslip Detail* yang diterima dari *server*, berisi informasi rinci mengenai slip gaji tertentu, termasuk rincian komponen gaji, potongan, tunjangan, serta total gaji yang diterima.

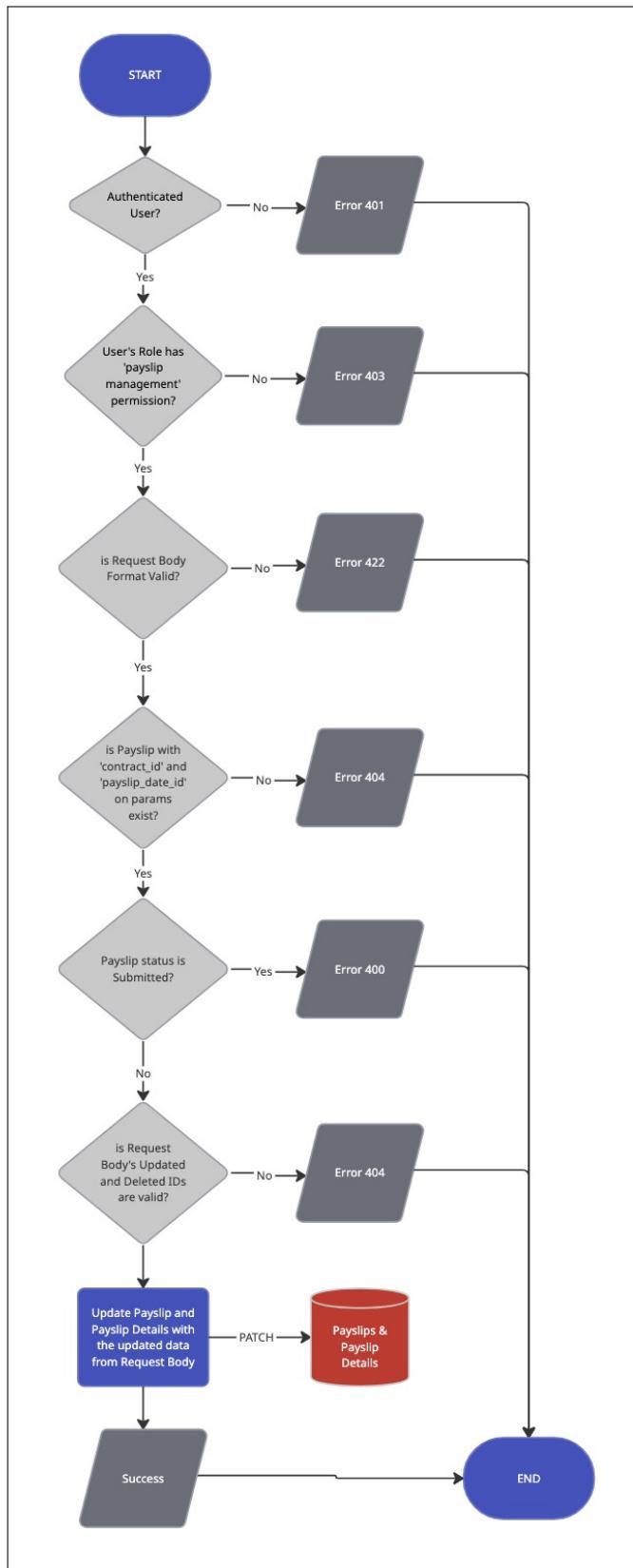
```
{
  "code": 200,
  "message": "Success",
  "data": {
    "start_date": "2025-11-30T17:00:00.000Z",
    "end_date": "2025-12-02T16:59:59.999Z",
    "status": "DRAFTED",
    "user_data": {
      "name": "HoHR 1",
      "nik": "112233445566778899",
      "contact": "08123456789",
      "job_title": "Director",
      "employment_status": "Full Time",
      "attendance": 2,
      "workdays": 2
    },
    "income": [
      {
        "id": "643e35cb-2dbc-4706-8928-c6a31dde119",
        "name": "Tunjangan BPJS",
        "amount": "35000",
        "description": "Tunjangan yang diberikan untuk BPJS karyawan.",
        "source": "SYSTEM"
      },
      {
        "id": "71ced7fd-e977-4651-aedd-096184447e2f",
        "name": "Tunjangan Kehormatan",
        "amount": "5000000",
        "description": "Tunjangan yang diberikan untuk jabatan karyawan.",
        "source": "SYSTEM"
      },
      {
        "id": "3a3b02a0-cb41-4884-812f-4f24df6e3129",
        "name": "Main Salary",
        "amount": "10000000",
        "description": "Gaji Pokok",
        "source": "SYSTEM"
      }
    ],
    "deduction": [
      {
        "id": "643e35cb-2dbc-4706-8928-c6a31dde119",
        "name": "Tunjangan BPJS",
        "amount": "35000",
        "description": "Tunjangan yang diberikan untuk BPJS karyawan.",
        "source": "SYSTEM"
      }
    ],
    "take_home_pay": 15000000,
    "bank_info": {
      "bank_name": "Bank Central Asia",
      "bank_account": "11223344556677",
      "bank_account_name": "HoHR 1"
    }
  }
}
```

Gambar 3.37. Contoh *response API Get Payslip Detail*

Save Payslip Detail

Save Payslip Detail adalah *endpoint API* yang berfungsi untuk menyimpan atau memperbarui detail slip gaji (*Payslip Detail*) berdasarkan *payslip_date_id* dan *contract_id* yang diberikan. API ini digunakan di sisi *frontend* untuk menyimpan perubahan atau pembaruan pada detail slip gaji dalam page *Payslip Detail*.

Gambar ?? merupakan *flowchart* API *Save Payslip Detail*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan memeriksa *permission payslip* untuk memperbarui detail slip gaji. Jika pengguna tidak ada *permission payslip*, sistem akan mengembalikan *error 403*. Setelah itu, sistem memeriksa apakah *request body* valid sesuai dengan ketentuan. Jika *request body* tidak valid, sistem akan mengembalikan *error 422*. Jika valid, sistem akan memeriksa apakah *payslip* yang dicari itu ada pada sistem. Jika tidak ada, sistem akan mengembalikan *error 404*. Setelah itu, sistem akan memeriksa kembali status dari *payslip* yang dicari, apakah status *payslip* tersebut merupakan *Submitted*. Bila iya, sistem akan mengembalikan *error 400*. Bila tidak, sistem memperbarui data detail slip gaji berdasarkan *payslip_date_id* dan *contract_id* yang diberikan dari *path parameters* dengan data yang diberikan dalam *request body*. Setelah data detail slip gaji berhasil diperbarui, sistem mengembalikan respons (*response*) kepada pengguna yang berisi informasi mengenai detail slip gaji yang telah diperbarui.



Gambar 3.38. Flowchart alur sistem API Save Payslip Detail

Pada gambar ?? menunjukkan contoh *request* untuk API *Save Payslip Detail* yang dikirimkan ke *endpoint* /payslip/date/payslip_date_id/contract/contract_id menggunakan metode HTTP *PATCH* dengan menyertakan *path parameters*, payslip_date_id dan contract_id, untuk menentukan slip gaji yang ingin diperbarui serta *request body* yang berisi data detail slip gaji yang akan diperbarui.

The screenshot shows a Swagger UI interface for a 'Save Payslip Detail' endpoint. At the top, it indicates the method is PATCH and the URL is /payslip/date/{payslip_date_id}/contract/{contract_id}. The status is 'Developing'. Below this, there's a note that it's 'Shared' and was created on September 9, 2025. The 'Description' section states: 'Save Payslip Detail with payslip_date_id and contract_id'. The 'Request' section includes an 'Authorization' field with instructions to provide a bearer token in the header and an example: 'Authorization: Bearer *****'. The 'Path Params' section defines two required parameters: 'payslip_date_id' (string) and 'contract_id' (string). The 'Data Schema' section shows a JSON schema for the request body:

```

{
  "income": {
    "create": [
      {
        "name": "Tambahans",
        "amount": "2000000",
        "description": "Tambahans untuk anak dan istri.",
        "type": "ADDITION"
      }
    ],
    "update": [
      {
        "id": "97652567-c661-4275-a0c6-5e00e0c6ae3b",
        "name": "THR",
        "amount": "4000000",
        "description": "Silakan digunakan untuk liburan spesial kamu."
      }
    ],
    "delete": []
  },
  "deduction": {
    "create": [
      {
        "name": "Pantry Fee",
        "amount": "300000",
        "description": "Menghabiskan semua makanan yang ada di pantry dalam 10 menit.",
        "type": "DEDUCTION"
      }
    ],
    "update": [],
    "delete": [
      {
        "id": "613f8164-b356-4e0b-9cbc-73925354ee8d"
      }
    ]
  }
}

```

Gambar 3.39. Contoh *request* API *Save Payslip Detail*

Pada gambar ?? menunjukkan contoh *response* untuk API *Save Payslip Detail* yang diterima dari *server*, berisi informasi mengenai detail slip gaji yang telah diperbarui, termasuk rincian komponen gaji, potongan, tunjangan, serta total gaji yang diterima.

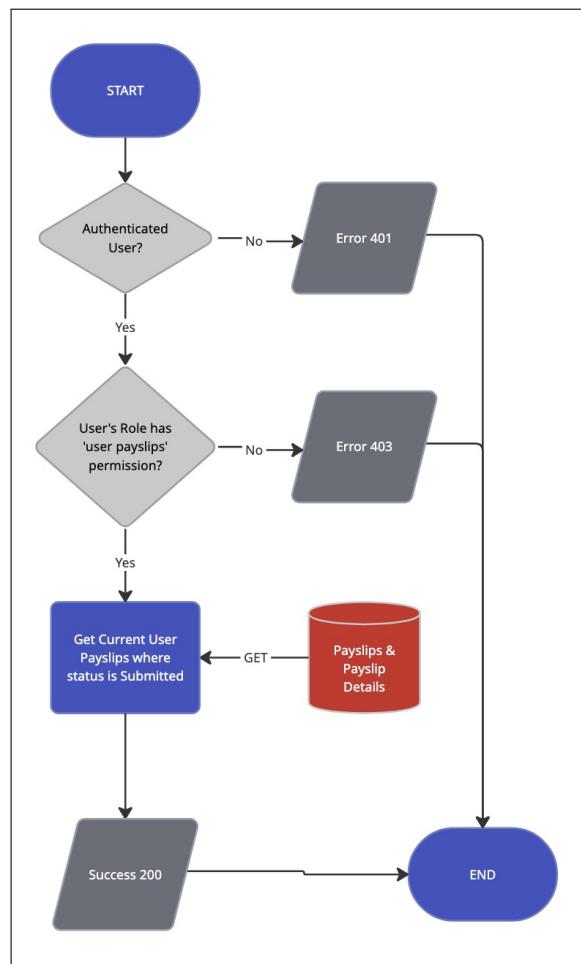
```
{
  "code": 200,
  "message": "Success",
  "data": {
    "payslip_id": "312e9e36-9140-4254-8c2b-72965b4e0586",
    "income": [
      {
        "id": "218b5cfcc-7ad4-43e9-885e-f9cd82f3de35",
        "payslip_id": "312e9e36-9140-4254-8c2b-72965b4e0586",
        "penalty_id": null,
        "type": "ADDITION",
        "name": "THR",
        "description": "Silakan digunakan untuk merayakan hari yang spesial!",
        "amount": "4000000",
        "source": "MANUAL",
        "source_id": null,
        "source_model": null,
        "created_at": "2025-12-20T06:54:56.457Z",
        "updated_at": "2025-12-20T06:56:20.034Z",
        "deleted_at": null
      },
      {
        "id": "643e35cb-2dbc-4706-8928-c6a31dde119",
        "payslip_id": "312e9e36-9140-4254-8c2b-72965b4e0586",
        "penalty_id": null,
        "type": "BENEFIT",
        "name": "Tunjangan BPJS",
        "description": "Tunjangan yang diberikan untuk BPJS karyawan.",
        "amount": "35000",
        "source": "SYSTEM",
        "source_id": null,
        "source_model": null,
        "created_at": "2025-12-20T06:49:02.754Z",
        "updated_at": "2025-12-20T06:49:02.754Z",
        "deleted_at": null
      },
      {
        "id": "71ced7fd-e977-4651-aedd-096184447e2f",
        "payslip_id": "312e9e36-9140-4254-8c2b-72965b4e0586",
        "penalty_id": null,
        "type": "ADDITION",
        "name": "Tunjangan Kehormatan",
        "description": "Tunjangan yang diberikan untuk jabatan karyawan.",
        "amount": "5000000",
        "source": "SYSTEM",
        "source_id": null,
        "source_model": null,
        "created_at": "2025-12-20T06:49:02.754Z",
        "updated_at": "2025-12-20T06:49:02.754Z",
        "deleted_at": null
      },
      {
        "id": "3a3b02a0-cb41-4884-812f-4f24df6e3129",
        "payslip_id": "312e9e36-9140-4254-8c2b-72965b4e0586",
        "penalty_id": null,
        "type": "ADDITION",
        "name": "Main Salary",
        "description": "Gaji Pokok",
        "amount": "10000000",
        "source": "SYSTEM",
        "source_id": null,
        "source_model": null,
        "created_at": "2025-12-20T06:49:02.754Z",
        "updated_at": "2025-12-20T06:49:02.754Z",
        "deleted_at": null
      }
    ],
    "deduction": [
      ...
    ]
  }
}
```

Gambar 3.40. Contoh *response* API *Save Payslip Detail*

Get User Payslips List

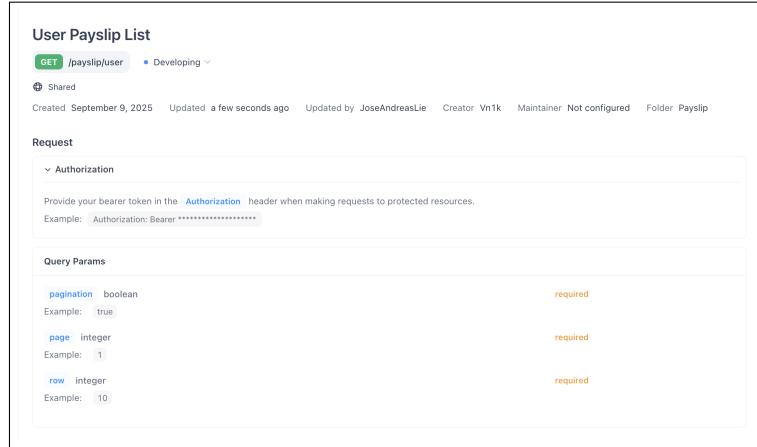
Get User Payslips List adalah *endpoint API* yang berfungsi untuk mengambil semua slip gaji (*Payslips*) milik pengguna yang sedang melakukan autentikasi. API ini digunakan di sisi *frontend* untuk menampilkan daftar slip gaji milik pengguna dalam page *Payslips*.

Gambar ?? merupakan *flowchart* API *Get User Payslips List*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan memeriksa *permission payslip* untuk mengakses data slip gaji. Jika pengguna tidak ada *permission payslip*, sistem akan mengembalikan *error 403*. Setelah itu, sistem mengambil data semua slip gaji milik pengguna yang sedang melakukan autentikasi. Setelah data berhasil diambil, sistem mengembalikan respons (*response*) kepada pengguna yang berisi daftar slip gaji milik pengguna.



Gambar 3.41. *Flowchart* alur sistem API *Get User Payslips List*

Pada gambar ?? menunjukkan contoh *request* untuk API *Get User Payslips List* yang dikirimkan ke *endpoint* /payslips/user menggunakan metode HTTP *GET* dengan menyertakan beberapa *query parameters* untuk kebutuhan *pagination*, seperti *page* dan *row*.



Gambar 3.42. Contoh *request* API *Get User Payslips List*

Pada gambar ?? menunjukkan contoh *response* untuk API *Get User Payslips List* yang diterima dari *server*, berisi daftar slip gaji milik pengguna yang sedang melakukan autentikasi, termasuk informasi seperti *payslip ID*, tanggal slip gaji, *job title*, dan *employment status*.

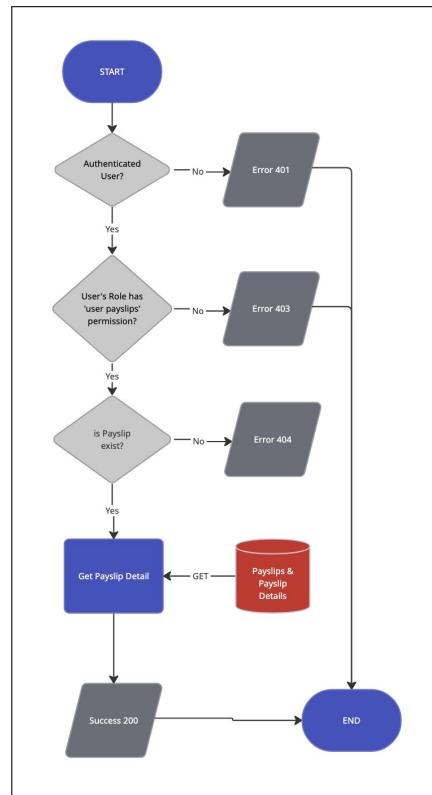
```
{
  "code": 200,
  "message": "Success",
  "data": {
    "count": 1,
    "rows": [
      {
        "id": "312e9e36-9140-4254-8c2b-72965b4e0586",
        "start_date": "2025-11-30T17:00:00.000Z",
        "end_date": "2025-12-02T16:59:59.999Z",
        "name": "HoHR 1",
        "job_title": "Director",
        "employment_status": "Full Time",
        "created_at": "2025-12-02T16:59:59.999Z",
        "updated_at": "2025-12-20T07:03:56.312Z"
      }
    ]
  }
}
```

Gambar 3.43. Contoh *response* API *Get User Payslips List*

Get User Payslip Detail by ID

Get User Payslip Detail by ID adalah *endpoint API* yang berfungsi untuk mengambil detail slip gaji (*Payslip Detail*) milik pengguna berdasarkan *ID* slip gaji yang diberikan. API ini digunakan di sisi *frontend* untuk menampilkan informasi rinci mengenai slip gaji tertentu milik pengguna dalam page *Payslip Detail*.

Gambar ?? merupakan *flowchart* API *Get User Payslip Detail by ID*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan memeriksa *permission payslip* untuk mengakses data slip gaji. Jika pengguna tidak ada *permission payslip*, sistem akan mengembalikan *error 403*. Setelah itu, sistem mengambil data slip gaji milik pengguna berdasarkan *ID* slip gaji yang diberikan dari *path parameters*. Jika data slip gaji tidak ditemukan, sistem akan mengembalikan *error 404*. Setelah itu, sistem mengambil data *payslip_details* yang berelasi dengan *payslip* tersebut. Setelah data berhasil diambil, sistem mengembalikan respons (*response*) kepada pengguna yang berisi detail slip gaji yang diminta.



Gambar 3.44. *Flowchart* alur sistem API *Get User Payslip Detail by ID*

Pada gambar ?? menunjukkan contoh *request* untuk API *Get User Payslip Detail by ID* yang dikirimkan ke *endpoint* /payslip/user/:id menggunakan metode HTTP *GET* dengan menyertakan *path parameters*, id, untuk menentukan slip gaji yang ingin diambil detailnya.

The screenshot shows a Swagger UI interface for an API endpoint. At the top left is a green button labeled "GET" next to the URL "/payslip/user/{payslip_id}". To its right is a status indicator "Developing" with a dropdown arrow. Below the URL are two status indicators: "Shared" with a user icon and "Not configured" with a folder icon. To the right of these are "Created September 9, 2025", "Updated 3 months ago", "Updated by JoseAndreasLie", and "Creator JoseAndreasLie". Below this section is a "Description" field containing the text "Only this user's submitted payslip." Under the "Request" section, there is a collapsed "Authorization" field with the note "Provide your bearer token in the Authorization header when making requests to protected resources." and an example "Example: Authorization: Bearer *****". Below this is a "Path Params" field containing a parameter "payslip_id" of type "string" with the note "required" and an example "Example: 312e9e36-9140-4254-8c2b-72965b4e0586".

Gambar 3.45. Contoh *request* API *Get User Payslip Detail by ID*

Pada gambar ?? menunjukkan contoh *response* untuk API *Get User Payslip Detail by ID* yang diterima dari *server*, berisi informasi rinci mengenai slip gaji tertentu milik pengguna, termasuk rincian komponen gaji, potongan, tunjangan, serta total gaji yang diterima.

```
{
    "code": 200,
    "message": "Success",
    "data": {
        "start_date": "2025-11-30T17:00:00.000Z",
        "end_date": "2025-12-02T16:59:59.999Z",
        "status": "SUBMITTED",
        "user_data": {
            "name": "HoHR 1",
            "nik": "112233445566778899",
            "contact": "08123456789",
            "job_title": "Director",
            "employment_status": "Full Time",
            "attendance": 2
        },
        "income": [
            {
                "id": "218b5cfc-7ad4-43e9-885e-f9cd82f3de35",
                "name": "THR",
                "amount": "4000000",
                "description": "Silakan digunakan untuk merayakan hari yang spesial!"
            },
            {
                "id": "643e35cb-2dbc-4706-8928-c6a31dde119",
                "name": "Tunjangan BPJS",
                "amount": "35000",
                "description": "Tunjangan yang diberikan untuk BPJS karyawan."
            },
            {
                "id": "71ced7fd-e977-4651-aedd-096184447e2f",
                "name": "Tunjangan Kehormatan",
                "amount": "5000000",
                "description": "Tunjangan yang diberikan untuk jabatan karyawan."
            },
            {
                "id": "3a3b02a0-cb41-4884-812f-4f24df6e3129",
                "name": "Main Salary",
                "amount": "10000000",
                "description": "Gaji Pokok"
            }
        ],
        "deduction": [
            {
                "id": "7ac04829-0956-462b-9eff-e13051c98b20",
                "name": "Kursi Gaming Kantor",
                "amount": "650000",
                "description": "Merusak Properti Kantor"
            },
            {
                "id": "643e35cb-2dbc-4706-8928-c6a31dde119",
                "name": "Tunjangan BPJS",
                "amount": "35000",
                "description": "Tunjangan yang diberikan untuk BPJS karyawan."
            }
        ],
        "take_home_pay": 18350000,
        "bank_info": {
            "bank_name": "Bank Central Asia",
            "bank_account": "11223344556677",
            "bank_account_name": "HoHR 1"
        }
    }
}
```

Gambar 3.46. Contoh *response API Get User Payslip Detail by ID*

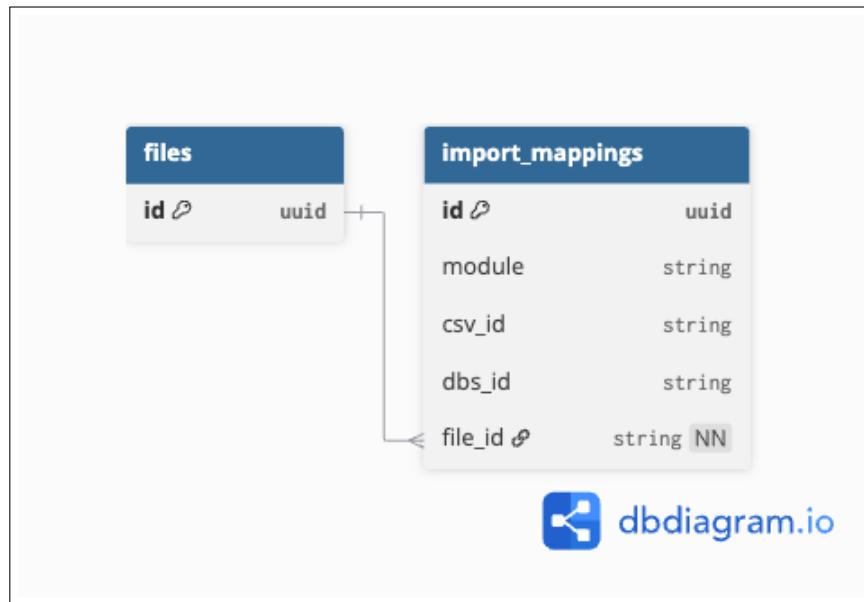
3.5.4 Modul On Boarding

Modul On Boarding dikembangkan untuk mempermudah proses penambahan data klien baru ke dalam sistem, dikembangkan modul On Boarding yang memungkinkan pengguna mengimpor data melalui *file CSV* dengan *template* yang sudah disiapkan. Modul ini mencakup fitur pemetaan kolom data dari berkas CSV ke struktur basis data yang sesuai, serta validasi data untuk memastikan integritas dan konsistensi informasi yang diimpor. Dengan adanya modul ini, proses penambahan klien baru menjadi lebih efisien sehingga mempercepat waktu implementasi sistem bagi pengguna baru.

Alur utama dari modul *On Boarding* dimulai dari klien mengisi data utama seperti lokasi perusahaan (*office location*), jam kerja (*schedule*), peranan (*role*), pengguna (*user*), jabatan (*job title*), komponen gaji (*payroll item*), jenis cuti (*leave type*), dan aturan denda (*penalty rule*) di *template sheets* yang sudah disediakan. Setelah klien sudah mengisi data di *template*, klien tinggal mengunggah dari setiap sheetsnya sebagai berkas CSV. Setelah data utama diunggah, klien dapat mengunduh *template* baru, kontrak (*contract*) dan komponen gaji lainnya (*contract item*) untuk diisi dengan data yang sebelumnya sudah di siapkan. Setelah mengisi data pada *template* yang baru, klien dapat mengunduh berkas menjadi format CSV tersebut lalu mengunggahnya ke dalam sistem. Sistem kemudian memproses berkas yang diunggah dengan melakukan validasi data dan menyimpan data ke dalam basis data (tabel *Import Mappings* dan tabel modul masing-masing). Jika terdapat kesalahan pada data yang diunggah, sistem akan memberikan notifikasi kepada klien untuk memperbaiki data tersebut. Setelah data utama dan data kedua berhasil, sesi *On Boarding* sudah selesai dan sistem siap mulai digunakan.

A Diagram ERD Modul On Boarding

Berikut merupakan *database diagram* untuk modul *On Boarding* yang telah dibuat selama pelaksanaan kerja praktik.



Gambar 3.47. Diagram ERD untuk modul *On Boarding*

Gambar ?? menunjukkan struktur basis data untuk modul *On Boarding*. Disini menggunakan table `import_mappings` untuk mencatat riwayat impor data yang dilakukan oleh pengguna. Setiap entri dalam tabel ini menyimpan informasi tentang pengguna yang melakukan impor, nama berkas (*file*) yang diunggah, serta pemetaan kolom yang digunakan selama proses impor. Hal ini memungkinkan pelacakan dan audit terhadap aktivitas impor data, serta memudahkan pengguna untuk mengelola dan meninjau pemetaan kolom yang telah mereka buat sebelumnya.

B *On Boarding API Endpoints*

Berikut adalah daftar *endpoints* API yang telah dikembangkan untuk modul *On Boarding*:

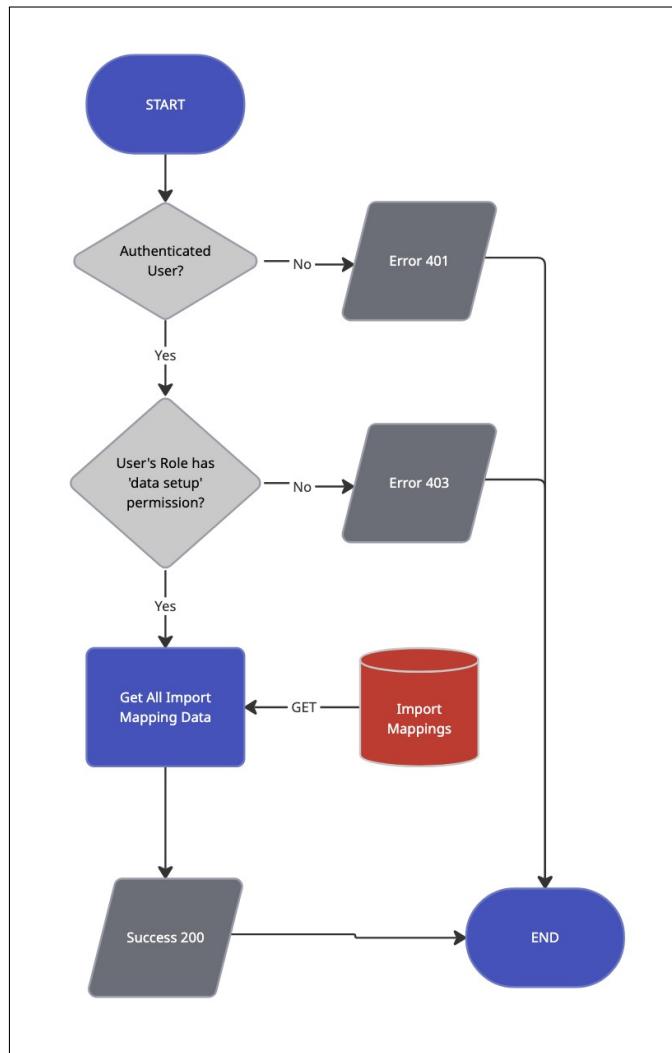
Tabel 3.5. Daftar *endpoints* API untuk modul *On Boarding*

Nama API	Metode	Endpoint	Deskripsi
Get On Boarding	GET	/on-boarding	Mengambil semua data impor
On Boarding Import	POST	/on-boarding/import-csv	Mengimpor data klien baru dari berkas CSV
Download Second Batch Template	GET	/on-boarding/download-template	Mengunduh <i>template</i> Excel untuk impor batch kedua

Berikut merupakan daftar *endpoints* API yang telah dikembangkan untuk modul *On Boarding* seperti pada Tabel ???. Setiap *endpoint* memiliki fungsi spesifik dalam mengelola proses impor data klien baru, mulai dari pengambilan data impor, hingga proses impor data dari berkas CSV. Selanjutnya, akan dijelaskan alur sistem untuk *endpoints* dalam pada modul ini.

Get On Boarding

Gambar ?? merupakan *flowchart* API *Get On Boarding*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error* 401. Jika autentikasi berhasil, sistem akan memeriksa *permission data setup* untuk mengakses data impor. Jika pengguna tidak ada *permission on boarding*, sistem akan mengembalikan *error* 403. Setelah itu, sistem mengambil data semua riwayat impor data klien baru. Setelah data berhasil diambil, sistem mengembalikan respons (*response*) kepada pengguna yang berisi daftar riwayat impor data.



Gambar 3.48. Flowchart alur sistem API *Get On Boarding*

Pada gambar ?? merupakan contoh *request* untuk API *Get On Boarding* yang dikirimkan ke *endpoint* /on-boarding menggunakan metode HTTP *GET*.

Get On Boarding

GET /on-boarding • Developing

Shared
Created November 25, 2025 Updated a month ago Updated by JoseAndreasLie Creator JoseAndreasLie Maintainer Not configured Folder On Boarding

Request

Authorization

Provide your bearer token in the `Authorization` header when making requests to protected resources.
Example: `Authorization: Bearer *****`

Gambar 3.49. Contoh *request* API *Get On Boarding*

Pada gambar ?? merupakan contoh *response* untuk API *Get On Boarding*

yang diterima dari *server*, berisi daftar riwayat impor data klien baru, termasuk informasi seperti *ID*, nama berkas (*file name*), dan modul yang diimpor.

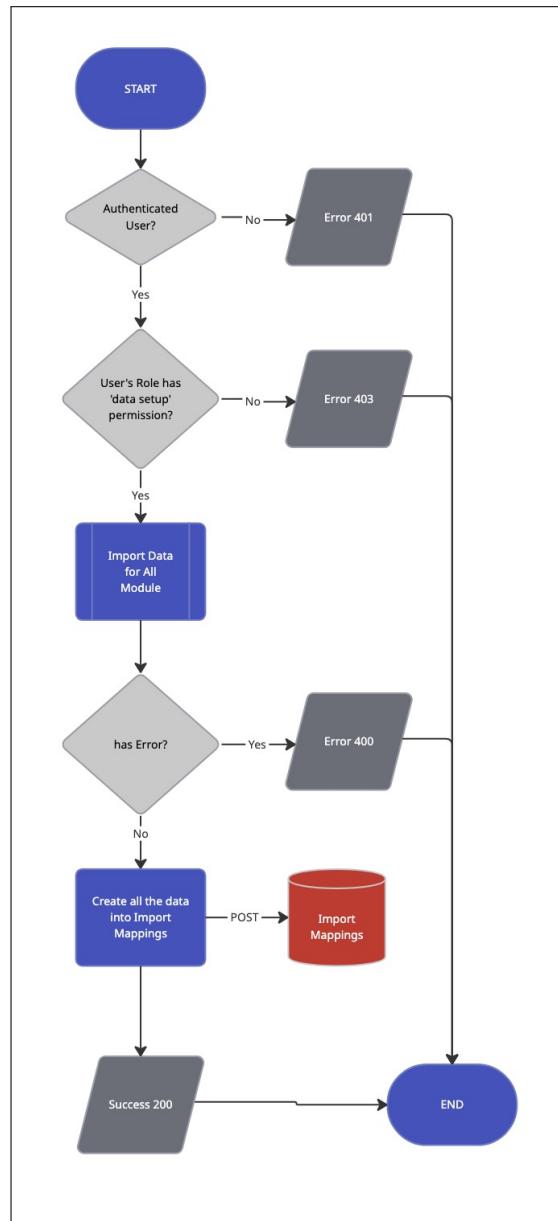
```
{
  "code": 200,
  "message": "On Boarding fetched successfully",
  "data": [
    {
      "id": "2d7287cc-599b-4e11-9867-46f7803ff699",
      "module": "leave_type",
      "file_id": "06cd0a4c-074f-40d5-9cab-b630a1486c8c",
      "name": "levtp.csv",
      "created_at": "2025-11-27T12:18:35.450Z",
      "updated_at": "2025-11-27T12:18:35.450Z"
    },
    {
      "id": "6b2623e6-49f2-4385-b126-f07706360e5c",
      "module": "office_location",
      "file_id": "1ddcf5d9-1800-4812-a219-82c724161ebe",
      "name": "ofloc.csv",
      "created_at": "2025-11-27T12:18:35.397Z",
      "updated_at": "2025-11-27T12:18:35.397Z"
    },
    {
      "id": "25145f0f-be91-4674-8870-a378d74e6d04",
      "module": "role",
      "file_id": "20547a38-f510-490d-9e09-e6211e1b7bc1",
      "name": "role.csv",
      "created_at": "2025-11-27T12:18:35.228Z",
      "updated_at": "2025-11-27T12:18:35.228Z"
    }
  ]
}
```

Gambar 3.50. Contoh *response API Get On Boarding*

On Boarding Import

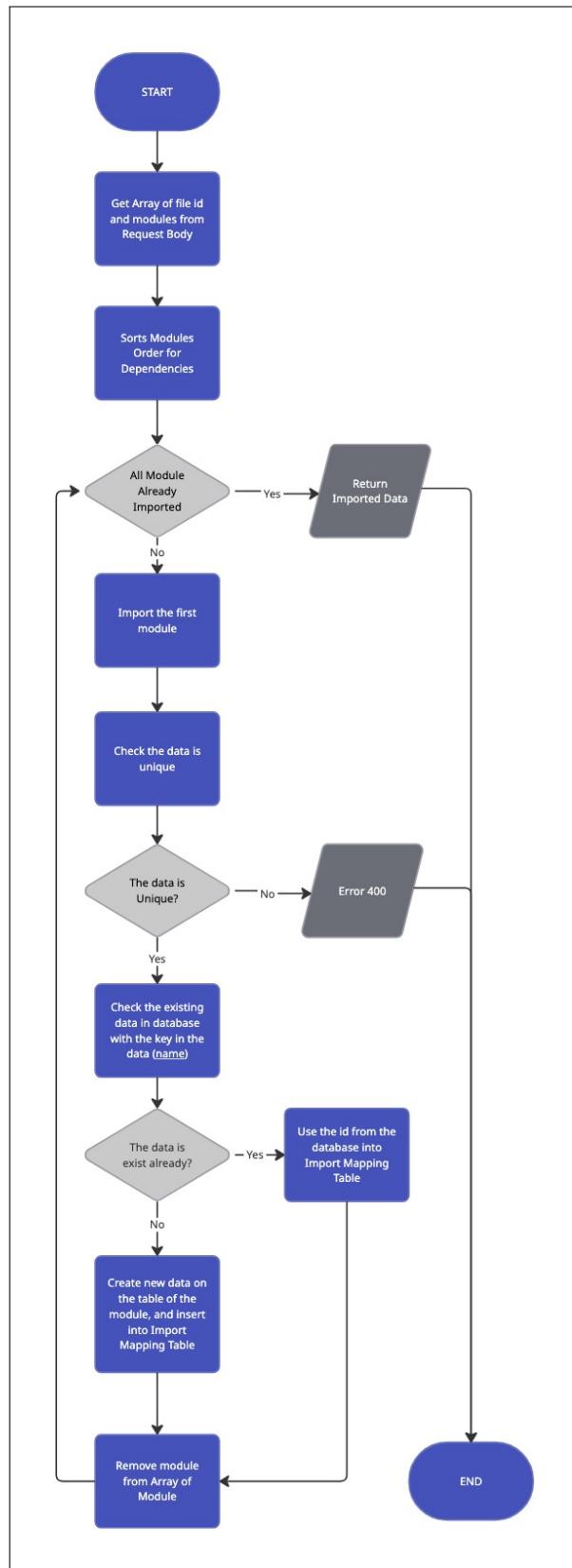
On Boarding Import adalah *endpoint API* yang berfungsi untuk mengimpor data klien baru dari berkas CSV yang diunggah oleh pengguna. API ini digunakan di sisi *frontend* untuk memproses impor data klien baru ke dalam sistem berdasarkan data yang terdapat pada berkas CSV. Gambar ?? dan Gambar ?? merupakan *flowchart API On Boarding Import*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error* 401. Jika autentikasi berhasil, sistem akan memeriksa *permission data setup* untuk mengimpor data klien baru. Jika pengguna tidak ada *permission on boarding*, sistem akan mengembalikan *error* 403. Setelah itu, sistem memproses data dari *request body* untuk mengimpor data yang sudah diunggah ke dalam table modul masing-masing. Setelah data berhasil diimpor, sistem mengembalikan respons

(response) kepada pengguna yang berisi informasi mengenai hasil impor data.



Gambar 3.51. Flowchart alur sistem API On Boarding Import

Pada gambar ?? merupakan subproses dari *flowchart API On Boarding Import* yang bernama *Import Data for All Module* yang menjelaskan proses untuk memastikan tidak ada data yang duplikat dan memproses memasukkan data dari masing-masing modul.



Gambar 3.52. Flowchart alur sistem API On Boarding Import (Lanjutan)

Pada gambar ?? menunjukkan contoh *request* untuk API *On Boarding Import* yang dikirimkan ke *endpoint* /on-boarding/import-csv menggunakan metode HTTP *POST* dengan menyertakan *request body* yang berisi data dari berkas CSV yang diunggah oleh pengguna.

The screenshot shows the configuration for a POST request to the endpoint /on-boarding/import-csv. The status is 'Developing'. The request body is defined under the 'Data Schema' tab, showing a JSON schema for the 'fileIds' array:

```
{
  "fileIds": [
    [
      {
        "59feb015-e816-439e-906b-ce243510bfc1",
        "contract_item"
      },
      [
        {
          "f8eff7be-168f-4749-bbed-15d3a14fd974",
          "contract"
        },
        [
          {
            "bf94f0c2-4836-48ec-a2da-1b4afcc17c9b",
            "job_title"
          },
          [
            {
              "13a3b3e9-52d6-4f09-bb7a-5f9f1cd5173d",
              "leave_type"
            },
            [
              {
                "f82074a6-c8cb-4d41-848c-885ae038333a",
                "office_location"
              },
              [
                {
                  "d9d7b42e-e4fa-45e8-b833-abfd8495f46d",
                  "payroll_item"
                },
                [
                  {
                    "19648deb-ab90-4cdb-86ea-b3f3d63417c0",
                    "role"
                  },
                  [
                    {
                      "6b4e17b4-b97e-4582-9762-6dd6204d7a4e",
                      "schedule"
                    },
                    [
                      {
                        "5457794c-6090-4afd-9963-61ea24e4e275",
                        "user"
                      }
                    ]
                  ]
                ]
              ]
            ]
          ]
        ]
      ]
    ]
  }
}
```

Gambar 3.53. Contoh *request* API *On Boarding Import*

Pada gambar ?? menunjukkan contoh *response* untuk API *On Boarding Import* yang diterima dari *server*, berisi informasi mengenai hasil impor data klien baru, termasuk jumlah data yang berhasil diimpor dan jumlah data yang gagal diimpor beserta alasan kegalannya.

```
{
    "code": 200,
    "message": "Success",
    "data": {
        "success": [
            {
                "module": "role",
                "count": 4
            },
            {
                "module": "user",
                "count": 7
            },
            {
                "module": "job_title",
                "count": 4
            },
            {
                "module": "office_location",
                "count": 1
            },
            {
                "module": "schedule",
                "count": 61
            },
            {
                "module": "payroll_item",
                "count": 3
            },
            {
                "module": "penalty_rule",
                "count": 4
            }
        ],
        "failed": [],
        "mappings": {
            "leave_type": {
                "Annual": "deec7b77-ef46-4fc9-a4f1-6a5a42e21028",
                "Force Major": "df95fb8e-8483-43fa-ab69-5a126a4a55f4",
                "WFH": "cf3368cc-7d08-438a-8300-8a40e6f03b0d",
                "WFH Director": "263c91a1-ed77-469c-9287-cebb5197e4b8"
            },
            "role": {
                "Supervisor": "b6278a11-3666-4cc6-9c4c-f816742e79d0",
                "Staff": "5f896215-7103-4f22-92bb-e800de0c3d07",
                "Human Resources": "bb447b00-4187-4ff4-b89c-83c0a89aba8b",
                "Head of Human Resources": "f66a4300-42d4-4bf2-b74d-0df88e1e2250"
            },
            "user": {
                "Staff 1": "720d357a-306f-48f5-937c-9b9d8d15cf39",
                "Supervisor 1": "0721dd2a-c7b0-4da6-9f11-25891d1c756f",
                "HR 1": "abd3bd7f-54f0-4579-ac48-533e64c0diec",
                "HR 2": "fe73cfe3-4501-4a66-bd73-731b10d48e41",
                "HoHR 1": "8163393b-10bb-4af1-9ce1-73fd6748d816",
                "Supervisor 2": "0a24d480-e4d4-4c91-a15a-642d583f81ab",
                "Staff 2": "3df396f2-6ce3-49ed-8d51-48d11a798197"
            },
            "job_title": {
                "Human Resource": "8cafee98-22de-4d60-bb8b-5fdb3d45373b",
                "Director": "199c12ae-8622-45fd-a57d-08496fe8c226",
                "Security": "32938e45-2a6b-4687-8fc4-af00d712cc2b",
                "Marketing": "53f41a6a-0c8a-4e02-bade-7379d95526b7"
            },
            "office_location": {
                "Tangerang Office @GOP": "26cd0609-8cb9-4856-a964-a6ee7920078e"
            },
            "schedule": {
                "Semin - Jumat (09:00 - 18:00)": "fba23a2c-cb1b-4fb4-9435-22450159d26c",
                "Sabtu (09:00 - 16:00)": "5fd60ee6-5214-490e-b80f-d3ff724fe9e5",
                "Lunch Time (12:00 - 13:00)": "2232f42f-19ea-43e0-85db-6783411f243a",
                "3 Shift": "5e6b4039-2afe-4b12-941f-2b9a4cd8cd7e",
                "2 Shift": "cf690dc8-3c6e-4966-bbe6-f345eb71ef6d",
                "Senin - Rabu (08:00 - 17:00)": "157abccb-17e3-4eb8-a63a-381bad3693f8",
                "Senin - Minggu (08:00 - 16:00)": "029436f8-a003-4d70-9f1c-7eb8d61d0961"
            }
        }
    }
}
```

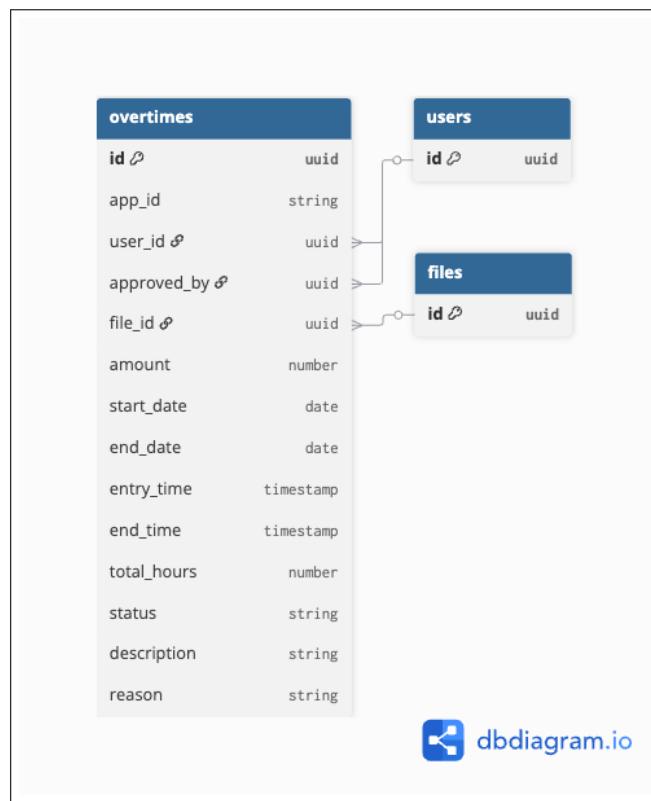
Gambar 3.54. Contoh *response API On Boarding Import*

3.5.5 Modul Overtime

Modul Overtime dikembangkan untuk mengelola dan mencatat jam kerja lembur pegawai secara efisien. Modul ini mencakup fitur pengajuan lembur, persetujuan oleh atasan, serta memasukkan kompensasi lembur sesuai dengan kebijakan perusahaan. Struktur basis data dirancang untuk menyimpan informasi terkait jam lembur, termasuk tanggal, durasi, dan status persetujuan. Dengan adanya modul ini, perusahaan dapat memantau dan mengelola jam kerja lembur secara transparan dan akurat.

A Diagram ERD Modul *Overtime*

Berikut merupakan *database diagram* untuk modul *Overtime* yang telah dibuat selama pelaksanaan kerja praktik.



Gambar 3.55. Diagram ERD untuk modul Overtime

Gambar ?? menunjukkan struktur basis data untuk modul Overtime.

B *Overtime API Endpoints*

Berikut adalah daftar *endpoints* API yang telah dikembangkan untuk modul *Overtime*:

Tabel 3.6. Daftar *endpoints* API untuk modul *Overtime*

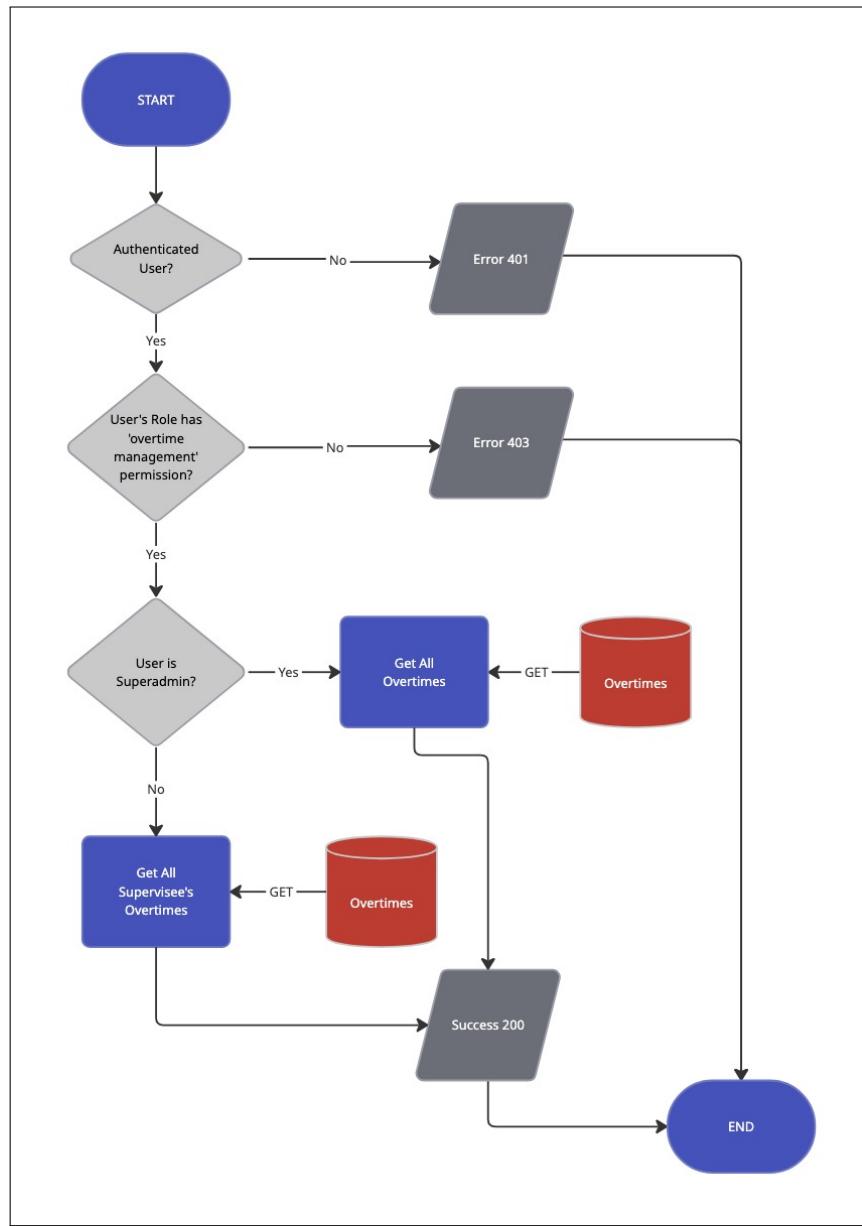
Nama API	Metode	Endpoint	Deskripsi
Get Overtime List	GET	/overtime	Mengambil semua permintaan lembur
Get User Overtime List	GET	/overtime/user	Mengambil permintaan lembur berdasarkan pengguna
Get Overtime Detail by ID	GET	/overtime/:id	Mengambil detail permintaan lembur berdasarkan ID
Delete Overtime by ID	DELETE	/overtime/:id	Menghapus permintaan lembur berdasarkan ID

Berikut merupakan daftar *endpoints* API yang telah dikembangkan untuk modul *Overtime* seperti pada Tabel ???. Setiap *endpoint* memiliki fungsi spesifik dalam mengelola permintaan lembur pegawai, mulai dari pengambilan data lembur, hingga penghapusan permintaan lembur. Selanjutnya, akan dijelaskan alur sistem untuk *endpoints* dalam pada modul ini.

Get Overtime List

Get Overtime List adalah *endpoint* API yang berfungsi untuk mengambil semua permintaan lembur (*Overtime*) yang ada di dalam sistem. API ini digunakan

di sisi *frontend* untuk menampilkan daftar permintaan lembur dalam page *Overtime*. Gambar ?? merupakan *flowchart* API *Get Overtime List*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan memeriksa *permission overtime* untuk mengakses data permintaan lembur. Jika pengguna tidak ada *permission overtime*, sistem akan mengembalikan *error 403*. Setelah itu, sistem mengambil data semua permintaan lembur. Setelah data berhasil diambil, sistem mengembalikan respons (*response*) kepada pengguna yang berisi daftar permintaan lembur.



Gambar 3.56. Flowchart alur sistem API *Get Overtime List*

Pada gambar ?? menunjukkan contoh *request* untuk API *Get Overtime List* yang dikirimkan ke *endpoint* /overtime menggunakan metode HTTP *GET* dengan menyertakan beberapa *query parameters* untuk kebutuhan *pagination*, seperti *page* dan *row*.

The screenshot shows a detailed API endpoint for retrieving overtime data. At the top, it indicates a **GET** method to **/overtime/**, with a status of **Released**. Below this, it shows the API is **Shared** and was created on **November 6, 2025**. The endpoint is listed under the **Overtime** folder.

Request

- Authorization**: Provides instructions to include a bearer token in the `Authorization` header.
- Query Params**: Lists various parameters:
 - pagination**: boolean, required, Example: `true`.
 - page**: string, required, Example: `1`.
 - row**: string, required, Example: `10`.
 - search**: string, optional, Example: `Calista Belva`.
 - sort**: enum<string>, optional, Allowed values: `start_date:ASC` `start_date:DESC` `end_date:ASC` `end_date:DESC` `total_hours:ASC` `total_hours:DESC`
Example: `total_hours:DESC`.
 - filter[status]**: string, optional, Example: `Pending`.

Gambar 3.57. Contoh *request API Get Overtime List*

Pada gambar ?? menunjukkan contoh *response* untuk API *Get Overtime List* yang diterima dari *server*, berisi daftar permintaan lembur, termasuk informasi seperti *overtime ID*, nama pengguna, tanggal lembur, durasi, dan status persetujuan.

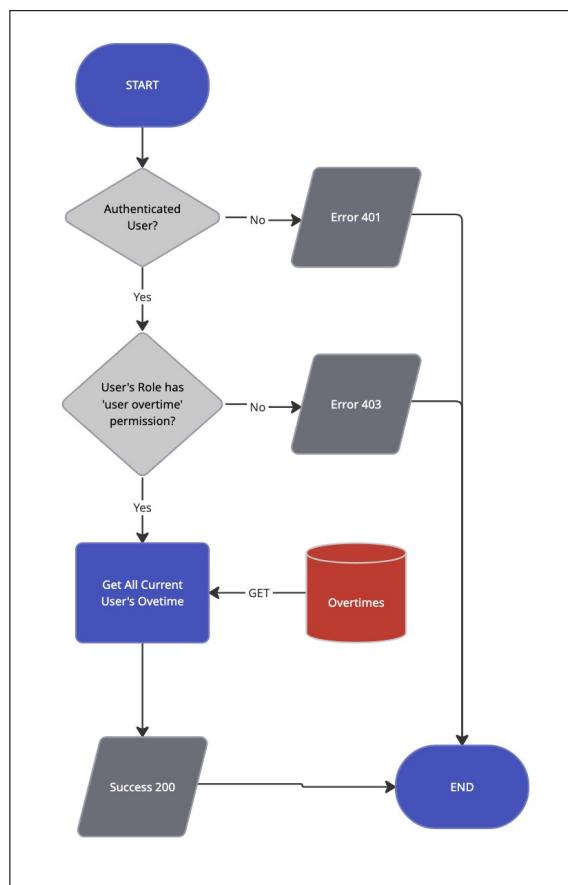
```
{
  "code": 200,
  "message": "Success",
  "data": {
    "count": 2,
    "rows": [
      {
        "id": "399b26ea-b659-4261-a660-385a9f2f719b",
        "app_id": "OT-202511-010",
        "name": "Calista Belva",
        "start_date": "2025-11-06T17:00:00.000Z",
        "end_date": "2025-11-06T17:00:00.000Z",
        "entry_time": "09:00:00",
        "end_time": "17:00:00",
        "total_hours": "8",
        "status": "Pending"
      },
      {
        "id": "069cb6c5-32fe-4cda-b1d4-c5deb6cc5258",
        "app_id": "OT-202511-009",
        "name": "Calista Belva",
        "start_date": "2025-11-05T17:00:00.000Z",
        "end_date": "2025-11-05T17:00:00.000Z",
        "entry_time": "09:00:00",
        "end_time": "17:00:00",
        "total_hours": "8",
        "status": "Pending"
      }
    ]
  }
}
```

Gambar 3.58. Contoh *response API Get Overtime List*

Get User Overtime List

Get User Overtime List adalah *endpoint API* yang berfungsi untuk mengambil semua permintaan lembur (*Overtime*) milik pengguna yang sedang melakukan autentikasi. API ini digunakan di sisi *frontend* untuk menampilkan daftar permintaan lembur milik pengguna dalam page *My Overtime*. Gambar ?? merupakan *flowchart API Get User Overtime List*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan

mengembalikan *error* 401. Jika autentikasi berhasil, sistem akan memeriksa *permission overtime* untuk mengakses data permintaan lembur. Jika pengguna tidak ada *permission overtime*, sistem akan mengembalikan *error* 403. Setelah itu, sistem mengambil data semua permintaan lembur milik pengguna yang sedang melakukan autentikasi. Setelah data berhasil diambil, sistem mengembalikan respons (*response*) kepada pengguna yang berisi daftar permintaan lembur milik pengguna.



Gambar 3.59. Flowchart alur sistem API *Get User Overtime List*

Pada gambar ?? menunjukkan contoh *request* untuk API *Get User Overtime List* yang dikirimkan ke *endpoint* /overtime/user menggunakan metode HTTP *GET* dengan menyertakan beberapa *query parameters* untuk kebutuhan *pagination*, seperti `page` dan `row`.

The screenshot shows a Swagger UI interface for a 'User Get All' endpoint. At the top, there is a green 'GET' button next to the URL '/overtime/user'. To the right of the button, it says 'Developing' with a dropdown arrow. Below the button, there is a 'Shared' icon and a timestamp 'Created November 6, 2025 Updated 2 minutes ago'. It also shows 'Updated by JoseAndreasLie', 'Creator JoseAndreasLie', 'Maintainer Not configured', and 'Folder Overtime'. The main area is titled 'Request' and contains sections for 'Authorization' (with a note to provide a bearer token) and 'Query Params'. The 'Query Params' section lists several parameters: 'pagination' (boolean, required), 'page' (number, required), 'row' (number, required), 'search' (string, optional), 'sort' (string, optional), and 'filter[status]' (string, optional). Examples for some parameters are provided.

Gambar 3.60. Contoh *request* API *Get User Overtime List*

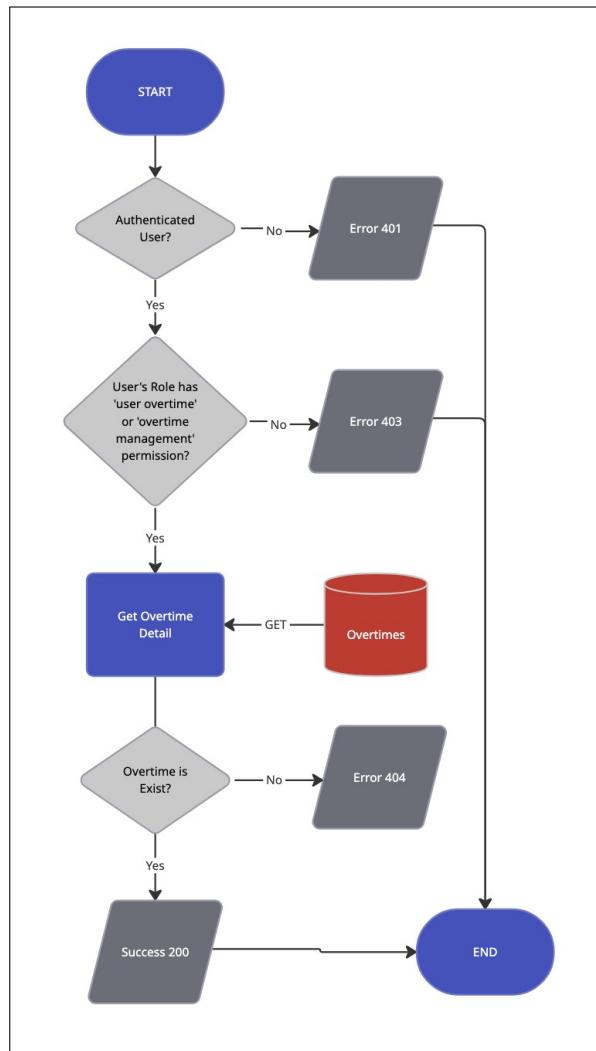
Pada gambar ?? menunjukkan contoh *response* untuk API *Get User Overtime List* yang diterima dari *server*, berisi daftar permintaan lembur milik pengguna yang sedang melakukan autentikasi, termasuk informasi seperti *overtime ID*, tanggal lembur, durasi, dan status persetujuan.

```
{
  "code": 200,
  "message": "Success",
  "data": {
    "count": 2,
    "rows": [
      {
        "id": "399b26ea-b659-4261-a660-385a0f2f719b",
        "app_id": "OT-202511-010",
        "name": "Calista Belva",
        "start_date": "2025-11-06T17:00:00.000Z",
        "end_date": "2025-11-06T17:00:00.000Z",
        "entry_time": "09:00:00",
        "end_time": "17:00:00",
        "total_hours": "8",
        "status": "Pending"
      },
      {
        "id": "069cb6c5-32fe-4cda-b1d4-c5deb6cc5258",
        "app_id": "OT-202511-009",
        "name": "Calista Belva",
        "start_date": "2025-11-05T17:00:00.000Z",
        "end_date": "2025-11-05T17:00:00.000Z",
        "entry_time": "09:00:00",
        "end_time": "17:00:00",
        "total_hours": "8",
        "status": "Pending"
      }
    ]
  }
}
```

Gambar 3.61. Contoh *response* API *Get User Overtime List*

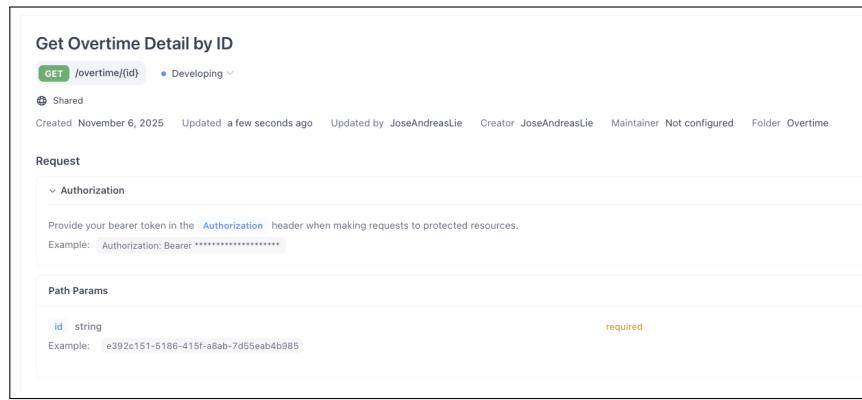
Get Overtime Detail by ID

Get Overtime Detail by ID adalah *endpoint* API yang berfungsi untuk mengambil detail permintaan lembur (*Overtime*) berdasarkan *ID* lembur yang diberikan. API ini digunakan di sisi *frontend* untuk menampilkan informasi rinci mengenai permintaan lembur tertentu dalam page *Overtime Detail*. Gambar ?? merupakan *flowchart* API *Get Overtime Detail by ID*. *Flowchart* dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error* 401. Jika autentikasi berhasil, sistem akan memeriksa *permission user overtime* atau *permission overtime management* untuk mengakses data permintaan lembur. Jika pengguna tidak ada *permission user overtime* atau *permission overtime management*, sistem akan mengembalikan *error* 403. Setelah itu, sistem mengambil data detail permintaan lembur berdasarkan *ID* lembur yang diberikan dari *path parameters*. Jika data permintaan lembur tidak ditemukan, sistem akan mengembalikan *error* 404. Setelah data berhasil diambil, sistem mengembalikan respons (*response*) kepada pengguna yang berisi detail permintaan lembur yang diminta.



Gambar 3.62. *Flowchart* alur sistem API *Get Overtime Detail by ID*

Pada gambar ?? menunjukkan contoh *request* untuk API *Get Overtime Detail by ID* yang dikirimkan ke *endpoint* /overtime/:id menggunakan metode HTTP *GET* dengan menyertakan *path parameters*, id, untuk menentukan permintaan lembur yang ingin diambil detailnya.



Gambar 3.63. Contoh *request* API *Get Overtime Detail by ID*

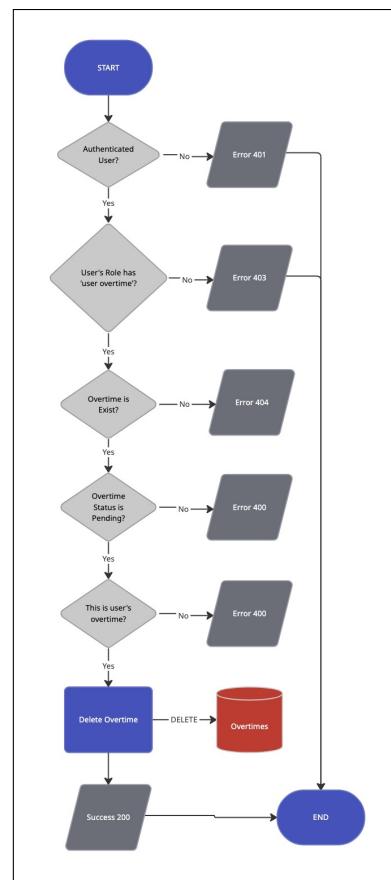
Pada gambar ?? menunjukkan contoh *response* untuk API *Get Overtime Detail by ID* yang diterima dari *server*, berisi informasi rinci mengenai permintaan lembur tertentu, termasuk rincian seperti nama pengguna, tanggal lembur, jam lembur, durasi, alasan lembur, status persetujuan, deskripsi lembur, dan lain-lain.

```
{
  "code": 200,
  "message": "Success",
  "data": {
    "id": "703de794-0a2b-431f-ba83-4566168fd1ac",
    "status": "approved",
    "name": "HoHR 1",
    "start_date": "2025-12-17",
    "end_date": "2025-12-17",
    "entry_time": "18:00",
    "end_time": "22:00",
    "total_hours": "3",
    "amount": "200000",
    "description": "Lembur karena disuruh Owner.",
    "reason": null,
    "approved_by": "Superadmin",
    "created_at": "2025-12-21T12:12:42.705Z",
    "updated_at": "2025-12-21T12:15:43.349Z",
    "user_id": "5a5ecd8d-e754-4e5c-8a6b-0e086a46dd6b",
    "file": {
      "id": "20cf480a-eb2f-4945-9fc6-7c0b7b0b0b8e",
      "name": "BuktiLembur001.jpg"
    }
  }
}
```

Gambar 3.64. Contoh *response* API *Get Overtime Detail by ID*

Delete Overtime by ID

Delete Overtime by ID adalah *endpoint API* yang berfungsi untuk menghapus permintaan lembur (*Overtime*) berdasarkan *ID* lembur yang diberikan. API ini digunakan di sisi *frontend* untuk menghapus permintaan lembur tertentu dalam page *Overtime Detail*. Gambar ?? merupakan *flowchart API Delete Overtime by ID*. Flowchart dimulai dari pengecekan autentikasi pengguna. Jika autentikasi gagal, sistem akan mengembalikan *error 401*. Jika autentikasi berhasil, sistem akan memeriksa *permission overtime management* untuk menghapus data permintaan lembur. Jika pengguna tidak ada *permission overtime management*, sistem akan mengembalikan *error 403*. Setelah itu, sistem menghapus data permintaan lembur berdasarkan *ID* lembur yang diberikan dari *path parameters*. Jika data permintaan lembur tidak ditemukan, sistem akan mengembalikan *error 404*. Setelah data permintaan lembur berhasil dihapus, sistem mengembalikan respons (*response*) kepada pengguna yang berisi informasi mengenai keberhasilan penghapusan permintaan lembur.



Gambar 3.65. Flowchart alur sistem API *Delete Overtime by ID*

Pada gambar ?? menunjukkan contoh *request* untuk API *Delete Overtime by ID* yang dikirimkan ke *endpoint* /overtime/:id menggunakan metode HTTP *DELETE* dengan menyertakan *path parameters*, id, untuk menentukan permintaan lembur yang ingin dihapus.

The screenshot shows a REST API documentation page for the 'Delete Overtime' endpoint. At the top, there is a red 'DELETE' button next to the URL '/overtime/{id}'. Below the button, it says 'Developing' with a dropdown arrow. There is also a 'Shared' icon. A timestamp 'Created November 6, 2025' and an update status 'Updated a month ago' are shown, along with details about the creator and maintainer. The main section is titled 'Request'. Under 'Authorization', it says 'Provide your bearer token in the Authorization header when making requests to protected resources.' and gives an example: 'Example: Authorization: Bearer *****'. Under 'Path Params', there is a table with one row. The column 'id' is described as 'string' and 'required'. An example value '454ba1f0-26d1-4509-afce-8f90c661bcd' is provided. The entire interface has a clean, modern design with a white background and light gray borders.

Gambar 3.66. Contoh *request* API *Delete Overtime by ID*

Pada gambar ?? menunjukkan contoh *response* untuk API *Delete Overtime by ID* yang diterima dari *server*, berisi *status code* 200 dengan pesan sukses mengindikasikan bahwa permintaan lembur telah berhasil dihapus.



Gambar 3.67. Contoh *response* API *Delete Overtime by ID*

3.6 Kendala dan Solusi yang Ditemukan

Selama pelaksanaan kerja praktik, terdapat beberapa kendala yang dihadapi dalam pengembangan sistem, antara lain:

- Beberapa *edge case* tidak teridentifikasi pada tahap pengembangan dan pengujian awal, salah satu contohnya adalah pada *Payslip Detail, penalty* yang dikenakan itu tidak sesuai dengan jangka waktu pada *Payslip* tersebut. Akibatnya, ketika sistem telah memasuki tahap produksi, masih ditemukan *bug* yang sebelumnya tidak terdeteksi. Hal ini berpotensi mengganggu stabilitas sistem dan pengalaman pengguna.
- Fokus pengembangan diarahkan pada pencapaian target agar klien dapat segera menggunakan modul utama, khususnya modul *Contract* dan *Payslip*. Kondisi ini menyebabkan keterbatasan waktu untuk melakukan *refactoring* kode lama serta perbaikan *bug* pada modul-modul yang telah dikembangkan sebelumnya.

Untuk mengatasi kendala-kendala tersebut, beberapa solusi yang diterapkan dan direkomendasikan adalah sebagai berikut:

- Pengembangan sistem dilengkapi dengan pengujian otomatis, seperti *unit testing*, untuk membantu mendeteksi *edge case* sejak tahap awal pengembangan. Dengan adanya pengujian otomatis, potensi *bug* dapat diidentifikasi lebih cepat sebelum sistem diterapkan ke lingkungan produksi.
- Untuk mengatasi keterbatasan waktu akibat prioritas pengembangan fitur utama, dilakukan perencanaan *refactoring* dan perbaikan *bug* secara bertahap. Pendekatan ini memungkinkan tim untuk tetap memenuhi target implementasi klien tanpa mengabaikan kualitas kode dan stabilitas modul yang telah ada.

BAB 4

SIMPULAN DAN SARAN

4.1 Simpulan

Pengembangan dari sisi *backend* pada sistem informasi kepegawaian Nine to Six di PT Visi Karya Nusantara telah berhasil dilakukan dan diimplementasikan sesuai dengan kebutuhan yang telah ditetapkan. Selama pengembangan, terdapat *framework* dan teknologi yang digunakan seperti *ExpressJs* untuk membangun *API*, *PostgreSQL* sebagai basis data, serta *Sequelize* sebagai *ORM* untuk memudahkan interaksi dengan basis data. Beberapa modul utama yang telah dikembangkan meliputi modul *Contract* untuk manajemen kontrak karyawan, modul *Payslip* untuk pengelolaan slip gaji, modul *Payroll Item* untuk membuat komponen gaji di kontrak karyawan, modul *On Boarding* untuk mempermudah klien baru saat baru menggunakan sistem, dan modul *Overtime* untuk membuat pengajuan lembur. Pengembangan ini juga melibatkan integrasi dengan sistem yang sudah ada, serta penerapan praktik terbaik dalam pengembangan perangkat lunak seperti penggunaan *version control* dengan Git dan penerapan *code review* untuk memastikan kualitas kode yang dihasilkan.

4.2 Saran

Ada beberapa saran yang dapat dipertimbangkan untuk pengembangan sistem informasi kepegawaian Nine to Six ke depannya:

1. Disarankan untuk menambahkan *unit testing* pada setiap fungsi atau modul yang dikembangkan. Hal ini akan memastikan bahwa setiap perubahan kode dapat langsung tervalidasi, sehingga meminimalkan risiko *bug* yang tidak terdeteksi.
2. Saling integrasi untuk modul-modul yang sudah ada seperti *Leave Permit*, *Overtime*, dan *Payslip*. Karena modul *Payslip* bisa mengambil data dari modul-modul tersebut seperti data lembur dari modul *Overtime* dan data cuti dari modul *Leave Permit*.

DAFTAR PUSTAKA

- [1] N. B. Moussa, “The impact of human resources information systems on individual innovation capability,” *Technology in Society*, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2444883418301992>
- [2] R. M. Shiferaw, “Digital technology and human resource practices,” *Heliyon*, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405844025003263>
- [3] R. N. Onuotu and W. A. Amaewhule, “Influence of automated accounting system on payroll preparation of small and medium scale enterprises in rivers state,” *Rivers State University Journal of Education (RSUJOE)*, vol. 27, no. 1, pp. 212–220, 2024. [Online]. Available: <https://rsujoe.com.ng/index.php/joe/article/download/217/186/583>
- [4] S. Meenugu, “The technical evolution of payroll systems: From manual processing to intelligent automation,” *Global Journal of Engineering and Technology Advances*, vol. 23, no. 1, pp. 201–208, 2025, review Article. [Online]. Available: <https://doi.org/10.30574/gjeta.2025.23.1.0103>
- [5] M. Fedushko and V. Batenko, “The link between knowledge management and the implementation of a working time recording system in the home office,” *Cogent Business Management*, vol. 10, no. 1, 2023. [Online]. Available: <https://doi.org/10.1080/23311886.2023.2281287>
- [6] S. Porkodi and A. Raman, “Success of cloud computing adoption over an era in human resource management systems: A comprehensive meta-analytic literature review,” *Management Review Quarterly*, 2025. [Online]. Available: <https://doi.org/10.1007/s11301-023-00401-0>
- [7] C. Kinyeki, “Adoption and implementation of integrated personnel and payroll database in kenya government ministries,” 2015. [Online]. Available: <https://erepository.uonbi.ac.ke/handle/11295/93334>
- [8] A. Ahmed, C. Mohammed, and A. Ahmad, “Web-based payroll management system: design, implementation, and evaluation,” *Journal of Electrical Systems and Infotech*, 2023. [Online]. Available: <https://doi.org/10.1186/s43067-023-00082-5>
- [9] S. S. Chowdhury and M. S. Haidear, “A review of human resource information system implementation approach in the corporate sector of bangladesh,” in *Proceedings of the 2nd International Conference on Business and Management (ICBM 2019)*. University of Dhaka, 2019, p. 559. [Online]. Available: <https://www.researchgate.net>

net/publication/336949303_A_Review_of_Human_Resource_Information_System_Implementation_Approach_in_the_Corporate_Sector_of_Bangladesh

- [10] D. Assimakopoulos, “An integrated web-based system for managing payrolls of regionally spread governmental offices,” in *Proceedings of the European Conference on E Government Eceg 2015*, 2015, pp. 489–498.
- [11] M. P. Bah, E. K. Duramany-Lakkoh, M. S. Kargbo, J. Kamara, and T. G. Bangura, “Assessing the effect of human resource information systems on the human resource strategies of commercial banks,” *European Journal of Business and Management Research*, vol. 7, no. 3, pp. 304–312, 2022. [Online]. Available: <https://www.ejbm.org/index.php/ejbm/article/view/1460>
- [12] J. A. Lie, “Transkrip wawancara bersama bapak raditya,” Dokumen Internal Laporan Magang, 2025, narasumber: Bapak Raditya (Software Engineer Manager).

Lampiran 1. PRO-STEP-01 Cover Letter

PRO-STEP-01 Cover Letter PRO-STEP Career Acceleration Program Track 2

Tangerang, December 29th 2025



No : 1/UMN/TI/Career Acceleration Program Track 2/IX/2025
Subject : **Student's Application for PRO-STEP Career Acceleration Program Track 2**

Dear. Head of Human Resource Department

PT Visi Karya Nusantara

Universitas Multimedia Nusantara's providing the PRO-STEP Career Acceleration Program Track 2, 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 Career Acceleration Program Track 2, in the company to learn solving problems based on knowledge that gained in campus, to link and match Career Acceleration Program Track 2 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 Career Acceleration Program Track 2 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 Career Acceleration Program Track 2 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)

Lampiran 2. PRO-STEP-02 Internship Card

PRO-STEP-02 PRO-STEP Career Acceleration Program Track 2 Card																					
PRO-STEP CAREER ACCELERATION PROGRAM TRACK 2 CARD																					
<table border="1"><tr><td>Name : Jose Andreas Lie</td><td>Student ID : 00000067097</td></tr><tr><td>Address : Taman Royal, Jl. Permata Raya No.1, RT.005/RW.002</td><td>Mobile Phone : 6285156119138</td></tr><tr><td colspan="2">PRO-STEP Career Acceleration Program Track 2 Acceptance Letter No: - Letter Date : 01-08-2025</td></tr><tr><td colspan="2">Advisor's Name : Fenina Adline Twince Tobing, M.Kom.</td></tr><tr><td colspan="2">Company Name : PT Visi Karya Nusantara</td></tr><tr><td>Company Address : Start Space Coworking Space Gading Serpong</td><td>City : TANGERANG Postal Code: 15810</td></tr><tr><td>Company Website : -</td><td>Company Phone : 0859106965884</td></tr><tr><td>Supervisor's Name : Atanasius Raditya Herkristito Supervisor's Phone : ??70859106965884?? Supervisor's Email : aftersixidn@gmail.com</td><td>Supervisor's Position : Senior Software Engineer Supervisor's Ext. : -</td></tr><tr><td>Department : Software Development Team</td><td>Position : Software Engineer</td></tr><tr><td colspan="2">Acceptance Date : 04-08-2025</td></tr></table>		Name : Jose Andreas Lie	Student ID : 00000067097	Address : Taman Royal, Jl. Permata Raya No.1, RT.005/RW.002	Mobile Phone : 6285156119138	PRO-STEP Career Acceleration Program Track 2 Acceptance Letter No: - Letter Date : 01-08-2025		Advisor's Name : Fenina Adline Twince Tobing, M.Kom.		Company Name : PT Visi Karya Nusantara		Company Address : Start Space Coworking Space Gading Serpong	City : TANGERANG Postal Code: 15810	Company Website : -	Company Phone : 0859106965884	Supervisor's Name : Atanasius Raditya Herkristito Supervisor's Phone : ??70859106965884?? Supervisor's Email : aftersixidn@gmail.com	Supervisor's Position : Senior Software Engineer Supervisor's Ext. : -	Department : Software Development Team	Position : Software Engineer	Acceptance Date : 04-08-2025	
Name : Jose Andreas Lie	Student ID : 00000067097																				
Address : Taman Royal, Jl. Permata Raya No.1, RT.005/RW.002	Mobile Phone : 6285156119138																				
PRO-STEP Career Acceleration Program Track 2 Acceptance Letter No: - Letter Date : 01-08-2025																					
Advisor's Name : Fenina Adline Twince Tobing, M.Kom.																					
Company Name : PT Visi Karya Nusantara																					
Company Address : Start Space Coworking Space Gading Serpong	City : TANGERANG Postal Code: 15810																				
Company Website : -	Company Phone : 0859106965884																				
Supervisor's Name : Atanasius Raditya Herkristito Supervisor's Phone : ??70859106965884?? Supervisor's Email : aftersixidn@gmail.com	Supervisor's Position : Senior Software Engineer Supervisor's Ext. : -																				
Department : Software Development Team	Position : Software Engineer																				
Acceptance Date : 04-08-2025																					
<p>This PRO-STEP Career Acceleration Program Track 2 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, December 16th 2025 Student's signature	 Jose Andreas Lie																				
																					
	Supervisor's signature & Company stamp																				
	 Atanasius Raditya Herkristito EST.																				
																					

Lampiran 3. PRO-STEP-03 Daily Task

PRO-STEP-03 Daily Task - Career Acceleration Program Track 2



Daily Task

STUDENT ID : 00000067097

STUDENT NAME : Jose Andreas Lie

COMPANY NAME : PT Visi Karya Nusantara

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
1	01/08/2025	13:01	20:01	On Boarding internship	Approved at 23 Oktober 2025 22:05
2	04/08/2025	13:01	20:01	Planning for Payroll Module	Approved at 23 Oktober 2025 22:05
3	05/08/2025	13:01	20:01	Making Contract, Payslip, Payroll Database Schemas	Approved at 23 Oktober 2025 22:05
4	06/08/2025	13:01	20:01	Making Contract, Payslip, Payroll Models Get Contract Detail API Create Payroll Items CRUD APIs	Approved at 23 Oktober 2025 22:05
5	07/08/2025	13:01	20:01	Payroll Setting Fix Contract Model Fix Payroll Setting API	Approved at 23 Oktober 2025 22:05
6	08/08/2025	13:01	20:01	Payroll Setting API (cont.) Fix seedings Contract API (cont.)	Approved at 23 Oktober 2025 22:05
7	11/08/2025	13:01	20:01	Contract APIs (cont.) Create seeding file on misc, penalty rules, and schedules table	Approved at 23 Oktober 2025 22:05
8	12/08/2025	13:01	20:01	Contract API (cont.) - Cancel Contract & Approval Contract	Approved at 23 Oktober 2025 22:05

Notes:

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

In witness whereof the company,



Atanasing Raditya Herkristito
Senior Software Engineer

Please sign along with the Company's stamp





Daily Task

STUDENT ID : 00000067097

STUDENT NAME : Jose Andreas Lie

COMPANY NAME : PT Visi Karya Nusantara

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
9	13/08/2025	13:01	20:01	Permission Check on Contract APIs Refactor Contract Controller on getAll	Approved at 23 Oktober 2025 22:05
10	14/08/2025	13:01	20:01	Contract APIs (cont.) - Delete Contract	Approved at 23 Oktober 2025 22:05
11	15/08/2025	13:01	20:01	Progress Update Team Retrospective	Approved at 23 Oktober 2025 22:05
12	18/08/2025	13:01	20:01	Fix Contract APIs	Approved at 23 Oktober 2025 22:05
13	19/08/2025	13:01	20:01	Fix Routing on Job Title New Create API for Job Title	Approved at 23 Oktober 2025 22:05
14	20/08/2025	13:01	20:01	Revamp Leave Permit	Approved at 23 Oktober 2025 22:05
15	21/08/2025	13:01	20:01	Revamp Leave Permit (cont.)	Approved at 23 Oktober 2025 22:05
16	22/08/2025	13:01	20:01	Reorder Penalty on Contract	Approved at 23 Oktober 2025 22:05

Notes:

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

In witness whereof the company,



Atanasius Raditya Herkristito
Senior Software Engineer

Please sign along with the Company's stamp



Daily Task

STUDENT ID : 00000067097

STUDENT NAME : Jose Andreas Lie

COMPANY NAME : PT Visi Karya Nusantara

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
17	25/08/2025	13:01	20:01	Revamp User Management	Approved at 23 Oktober 2025 22:05
18	26/08/2025	13:01	20:01	Revamp User Management (cont.)	Approved at 23 Oktober 2025 22:05
19	27/08/2025	13:01	20:01	Revamp User Management (cont.) Revamp auth validator, remove model association Removed unused codes	Approved at 23 Oktober 2025 22:05
20	28/08/2025	13:01	20:01	Revamp Leave Management Flow	Approved at 23 Oktober 2025 22:05
21	29/08/2025	13:01	20:01	Revamp Permit Detail and User Leave Revamp CountDay function	Approved at 23 Oktober 2025 22:05
22	01/09/2025	13:01	20:01	Revamp CountDay function (cont.) New Approve Leave API	Approved at 23 Oktober 2025 22:05
23	02/09/2025	13:01	20:01	Add a function on SuperDao Refactor user involved function using current active contract	Approved at 23 Oktober 2025 22:05
24	03/09/2025	13:01	20:01	Fix pagination default sort Set unique on Job Title	Approved at 23 Oktober 2025 22:05

Notes:

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

In witness whereof the company,



Atanasius Raditya Herkristito
Senior Software Engineer

Please sign along with the Company's stamp



Daily Task

STUDENT ID : 00000067097

STUDENT NAME : Jose Andreas Lie

COMPANY NAME : PT Visi Karya Nusantara

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
25	04/09/2025	13:01	20:01	Revamp pagination search option for nested models Removed job title attributes on include models	Approved at 23 Oktober 2025 22:05
26	05/09/2025	13:01	20:01	Update Attendances Fix Project Product and Job Title	Approved at 23 Oktober 2025 22:05
27	08/09/2025	13:01	20:01	Revamp user report and user contract dao Revamp People Report	Approved at 23 Oktober 2025 22:05
28	09/09/2025	13:01	20:01	Revamp Stand Up API Responses Revamp User Report on Stand Ups Remove Duppes Code	Approved at 23 Oktober 2025 22:05
29	10/09/2025	13:01	20:01	Add Description and Value to Contract and Payroll Item Revise getStartDate function Add Paydate on Miscellaneous	Approved at 23 Oktober 2025 22:05
30	11/09/2025	13:01	20:01	Add Required Associations on Models Payslip Details APIs	Approved at 23 Oktober 2025 22:05
31	12/09/2025	13:01	20:01	Add pagination on payroll items Refine and fix payroll item Create and Update API Revamp Leave Type	Approved at 23 Oktober 2025 22:05

Notes:

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

In witness whereof the company,



Atanasius Raditya Herkristito
Senior Software Engineer

Please sign along with the Company's stamp



Daily Task

STUDENT ID : 00000067097

STUDENT NAME : Jose Andreas Lie

COMPANY NAME : PT Visi Karya Nusantara

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
32	15/09/2025	13:01	20:01	Finalization Payslip API Fix People Report Bug	Approved at 23 Oktober 2025 22:05
33	16/09/2025	13:01	20:01	Add Pagination Sorts Fix sort Daily Attendances Add Pagination on Payslip	Approved at 23 Oktober 2025 22:05
34	17/09/2025	13:01	20:01	New Migration and models on Paysip Details Revamp Update Payslip Details Fix GetAll Payslip dates and detail	Approved at 23 Oktober 2025 22:05
35	18/09/2025	13:01	20:01	People Report Summary Fix	Approved at 23 Oktober 2025 22:05
36	19/09/2025	13:01	20:01	Revise Validator	Approved at 23 Oktober 2025 22:05
37	22/09/2025	13:01	20:01	Export XLSX, Remove date validation Revamp Create User API	Approved at 23 Oktober 2025 22:05
38	23/09/2025	13:01	20:01	Refine migration and seeding files	Approved at 23 Oktober 2025 22:05
39	24/09/2025	13:01	20:01	Refine migration and seeding files (cont.)	Approved at 23 Oktober 2025 22:05

Notes:

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

In witness whereof the company,



Atanasius Raditya Herkristito
Senior Software Engineer

Please sign along with the Company's stamp





Daily Task

STUDENT ID : 00000067097

STUDENT NAME : Jose Andreas Lie

COMPANY NAME : PT Visi Karya Nusantara

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
40	25/09/2025	13:01	20:01	Refine migration and seeding files (cont.)	Approved at 23 Oktober 2025 22:05
41	26/09/2025	13:01	20:01	New Permission on Every Modules Fix Register Holiday Permission	Approved at 23 Oktober 2025 22:05
42	29/09/2025	13:01	20:01	Fix Contract Route Leave Setting Update Fix Account Seeding for Bank	Approved at 23 Oktober 2025 22:05
43	30/09/2025	13:01	20:01	Create FindAllActiveContract Function Update summary to a specific time Add start_date and end_date to user report record attendance	Approved at 23 Oktober 2025 22:05
44	01/10/2025	13:01	20:01	Fix Registration Password Fix getAll User API	Approved at 23 Oktober 2025 22:05
45	02/10/2025	13:01	20:01	Set Leave to be Optional on Create Contract Change Constant on Project Product Restrains on Delete user with Active Contract	Approved at 23 Oktober 2025 22:05
46	03/10/2025	13:01	20:01	Planning for "On Boarding" Data input for client New CSV File Reader/Parser	Approved at 23 Oktober 2025 22:05

Notes:

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

In witness whereof the company,



Atanasius Raditya Herkristito
Senior Software Engineer

Please sign along with the Company's stamp



Daily Task

STUDENT ID : 00000067097

STUDENT NAME : Jose Andreas Lie

COMPANY NAME : PT Visi Karya Nusantara

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
47	06/10/2025	13:01	20:01	Planning for "On Boarding" Data input for client (cont.) New CSV File Reader/Parser (cont.)	Approved at 23 Oktober 2025 22:05
48	07/10/2025	13:01	20:01	Revamp FileController for On Boarding	Approved at 23 Oktober 2025 22:05
49	08/10/2025	13:01	20:01	Create Table Import Mapping Create Import Mapping function	Approved at 23 Oktober 2025 22:05
50	09/10/2025	13:01	20:01	Create Import Mapping function (cont.)	Approved at 23 Oktober 2025 22:05
51	10/10/2025	13:01	20:01	Create Import Mapping function (cont.)	Approved at 23 Oktober 2025 22:05
52	13/10/2025	13:01	20:01	Create On Boarding Service and make the Parsing on every modules function	Approved at 23 Oktober 2025 22:05
53	14/10/2025	13:01	20:01	Create On Boarding Service and make the Parsing on every modules function (cont.)	Approved at 23 Oktober 2025 22:05
54	15/10/2025	13:01	20:01	Create On Boarding Service and make the Parsing on every modules function (cont.)	Approved at 23 Oktober 2025 22:05

Notes:

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

In witness whereof the company,



Please sign along with the Company's stamp



Daily Task

STUDENT ID : 00000067097

STUDENT NAME : Jose Andreas Lie

COMPANY NAME : PT Visi Karya Nusantara

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
55	16/10/2025	13:01	20:01	Create On Boarding Service and make the Parsing on every modules function (cont.)	Approved at 23 Oktober 2025 22:05
56	17/10/2025	13:01	20:01	Create On Boarding Service and make the Parsing on every modules function (cont.)	Approved at 23 Oktober 2025 22:05
57	20/10/2025	13:01	20:01	Add Error Handling on On Boarding Function for CSV Files Uploaded	Approved at 23 Oktober 2025 22:05
58	21/10/2025	13:01	20:01	Add Error Handling on On Boarding Function for CSV Files Uploaded (cont.)	Approved at 23 Oktober 2025 22:05
59	22/10/2025	13:01	20:01	Add Error Handling on On Boarding Function for CSV Files Uploaded (cont.)	Approved at 23 Oktober 2025 22:05
60	23/10/2025	13:01	20:01	Enable user to upload and import sections of module first.	Approved at 23 Oktober 2025 22:05
61	24/10/2025	13:01	22:01	Fix On Boarding JSON Response Revamp Parsing CSV	Approved at 13 Desember 2025 21:25
62	27/10/2025	13:01	22:01	Revamp CSV Requirement from id to name	Approved at 13 Desember 2025 21:25

Notes:

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

In witness whereof the company,



Please sign along with the Company's stamp





Daily Task

STUDENT ID : 00000067097

STUDENT NAME : Jose Andreas Lie

COMPANY NAME : PT Visi Karya Nusantara

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
63	28/10/2025	13:01	22:01	Revamp CSV Requirement from id to name (cont.)	Approved at 13 Desember 2025 21:25
64	29/10/2025	13:01	22:01	Fix on Contract Item for Payslip	Approved at 13 Desember 2025 21:25
65	30/10/2025	13:01	22:01	On Boarding Service Fix on Schedule Module	Approved at 13 Desember 2025 21:25
66	31/10/2025	13:01	22:01	On Progress Meeting Retro	Approved at 13 Desember 2025 21:25
67	01/11/2025	13:01	22:01	Fix on boarding service on Schedule (cont.) Fix on boarding service on User	Approved at 13 Desember 2025 21:25
68	03/11/2025	13:01	22:01	Fix Seeding and Create Overtime Get APIs with rules	Approved at 13 Desember 2025 21:25
69	04/11/2025	13:01	22:01	Fix Seeding and Create Overtime Get APIs with rules (cont.)	Approved at 13 Desember 2025 21:25
70	05/11/2025	13:01	22:01	Fix Seeding and Create Overtime Get APIs with rules (cont.)	Approved at 13 Desember 2025 21:25

Notes:

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

In witness whereof the company,



Atanasis Raditya Herkristito
Senior Software Engineer

Please sign along with the Company's stamp





Daily Task

STUDENT ID : 00000067097

STUDENT NAME : Jose Andreas Lie

COMPANY NAME : PT Visi Karya Nusantara

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
71	06/11/2025	13:01	22:01	Add File on Overtime get by id	Approved at 13 Desember 2025 21:25
72	07/11/2025	13:01	22:01	Meeting Progress	Approved at 13 Desember 2025 21:25
73	10/11/2025	13:01	22:01	new pagination sort fields and searchable fields removed unused baseURL	Approved at 13 Desember 2025 21:25
74	11/11/2025	13:01	22:01	Truncate `HH:mm:ss` to `HH:mm` Research about how to connect Overtime to Payslip Module	Approved at 13 Desember 2025 21:25
75	12/11/2025	13:01	22:01	Research about how to connect Overtime to Payslip Module (cont.)	Approved at 13 Desember 2025 21:25
76	13/11/2025	13:01	22:01	Research about how to connect Overtime to Payslip Module (cont.)	Approved at 13 Desember 2025 21:25
77	14/11/2025	13:01	22:01	Progress Update and Meeting	Approved at 13 Desember 2025 21:25
78	17/11/2025	13:01	22:01	Overtime integrate with Payslip	Approved at 13 Desember 2025 21:25

Notes:

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

In witness whereof the company,


 Atanasius Raditya Herkristito
 Senior Software Engineer
 Please sign along with the Company's stamp



Daily Task

STUDENT ID : 00000067097

STUDENT NAME : Jose Andreas Lie

COMPANY NAME : PT Visi Karya Nusantara

No	Date	In	Out	Duties /Responsibilities	Supervisor's Sign
79	18/11/2025	13:01	22:01	Overtime integrate with Payslip (cont.)	Approved at 13 Desember 2025 21:25
80	19/11/2025	13:01	22:01	Payslip Service update for Overtime and Reimbursement (cont.) Adding 2 more APIs for get overtime and reimbursement data	Approved at 13 Desember 2025 21:25
81	20/11/2025	13:01	22:01	Meeting Progress	Approved at 13 Desember 2025 21:25
82	21/11/2025	13:01	22:01	Testing	Approved at 13 Desember 2025 21:25
83	24/11/2025	13:01	22:01	Listing Backlogs Testing	Approved at 13 Desember 2025 21:25
84	25/11/2025	13:01	22:01	Testing Create On Boarding API (Get & Delete)	Approved at 13 Desember 2025 21:25
85	26/11/2025	13:01	22:01	Testing Delete data inside csv when delete on Data Setup	Approved at 13 Desember 2025 21:25
86	27/11/2025	13:01	22:01	Testing Fix On Boarding Get and Delete APIs	Approved at 13 Desember 2025 21:25

Notes:

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

In witness where company,



Atanasis Raditya Herkristito
Senior Software Engineer

Please sign along with the Company's stamp



Lampiran 4. PRO-STEP-04 Verification Form

PRO-STEP-04 Verification Form of Internship Report PRO-STEP Career Acceleration Program
Track 2



VERIFICATION FORM OF INTERNSHIP REPORT PRO-STEP CAREER ACCELERATION PROGRAM TRACK 2

Student's Career Acceleration Program Track 2 Advisor

Name : Fenina Adline Twince Tobing, M.Kom.

I, who signed below

Name : Atanasius Raditya Herkristito

Position : Senior Software Engineer

Company : PT Visi Karya Nusantara

had received, read and approved the Internship Report from

Student ID : 00000067097

Student Name : Jose Andreas Lie

Period : January 2026

Report Title : Backend Enhancement of the HR Information System at PT Visi Karya Nusantara

Tangerang, 22 December 2025

A handwritten signature in black ink, appearing to read "Fenina Adline Twince Tobing, M.Kom.".

Fenina Adline Twince Tobing, M.Kom.
Student Advisor

Tangerang, 22 December 2025



Atanasius Raditya Herkristito
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. PRO-STEP-05 Surat Penerimaan



PT Visi Karya Nusantara
Start Space Coworking Space Gading Serpong,
Tangerang, Banten, 15810
aftersixidn@gmail.com | +859 106 965 884

Internship Offer Letter

Date: 01 August 2025

Dear Jose Andreas Lie,

We are pleased to offer you an internship position at PT Visi Karya Nusantara as a Software Engineer within our Software Development Team. This internship is designed to provide you with valuable hands-on experience and exposure to real-world software development processes, collaborative teamwork, and product development cycles.

Below are the key details of your internship:

- Start Date: 4 August 2025
- End Date: 4 January 2026
- Location: Remote
- Working Hours: 35 hours per week
- Stipend (if any): As specified in the attached internship contract

During your internship, you will report directly to Raditya, who will serve as your supervisor and provide guidance, support, and regular feedback throughout your internship period.

Please confirm your acceptance of this offer by signing and returning a copy of this letter no later than **4 August 2025**. If you have any questions regarding the terms of the internship, feel free to reach out.

We are excited to welcome you aboard and look forward to the valuable contributions you will make. We hope this internship will serve as a meaningful step in your professional and personal growth.

Warm regards,
Keshia Tiffany
Director
PT Visi Karya Nusantara

Acknowledged and Accepted by:

Signature: _____
Name: Jose Andreas Lie
Date: _____

Lampiran 6. Form Bimbingan

Form Bimbingan Internship Report Program Studi Informatika Semester Gasal 2025/2026

Nama : Jose Andreas Lie
NIM : 00000067097
Angkatan : 2022
Dosen Pembimbing : Fenina Adline Twince Tobing, M.Kom.

Meeting	Tanggal	Jam	Keterangan	Tanggal Approval
1	03 Oktober 2025	14:00	Info session mengenai Magang Track 2 Requirement untuk lulus sidang magang Gunakan template yang disediakan Untuk UTS, usahakan Bab 1 - 3	24 Desember 2025
2	21 Oktober 2025	10:00	Penilaian UTS Progress Update (Magang & Laporan)	24 Desember 2025
3	29 September 2025	10:00	Counseling Perdana dengan Dosen Pembimbing Running through gimana proses Magang di Track 2	24 Desember 2025
4	10 November 2025	15:00	Update Progress: 1. Remove all bolds (hanya judul di bold) 2. Remove all ambiguity (seperti kata 'sempurna') 3. Singkatan harus di define sebelum menggunakan singkatan nya saja	24 Desember 2025
5	24 November 2025	10:00	Ada beberapa minor revisi, sisanya sudah oke, lanjutkan dulu bab 3 nya	24 Desember 2025
7	15 Desember 2025	10:00	Sudah oke, segera cepat selesaikan Bab 3	24 Desember 2025
9	01 Desember 2025	10:00	Input untuk refactor struktur bab nya, cukup jelaskan ketentuan api nya gimana, lalu jelaskan setiap modul dan setiap api nya flowchart nya bagaimana dan erd setiap modul gimana.	24 Desember 2025
10	22 Desember 2025	10:00	- Update Judul - Finalisasi Laporan - Judul pada flowchart dihilangkan	24 Desember 2025



Lampiran 7. Hasil Pengecekan Similarity Turnitin

 turnitin Page 1 of 82 - Cover Page Submission ID trn:oid::1:3451387202

Umn Libtii TI 3

MBKM_JOSE_ANDREAS_LIE

 JOSE ANDREAS LIE
 2025 GANJIL - MAGANG (REGULER & MBKM) TEKNIK INFORMATIKA
 Universitas Multimedia Nusantara

Document Details

Submission ID	trn:oid::1:3451387202	78 Pages
Submission Date	Dec 29, 2025, 6:19 PM GMT+7	9,651 Words
Download Date	Dec 29, 2025, 6:24 PM GMT+7	62,290 Characters
File Name	MBKM_JOSE_ANDREAS_LIE.pdf	
File Size	5.5 MB	

 turnitin Page 1 of 82 - Cover Page Submission ID trn:oid::1:3451387202



17% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- ▶ Bibliography
- ▶ Quoted Text

Top Sources

17%	Internet sources
0%	Publications
0%	Submitted works (Student Papers)





Top Sources

17% Internet sources
0% Publications
0% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

Rank	Type	Source	Percentage
1	Internet	kc.umn.ac.id	16%
2	Internet	123dok.com	<1%
3	Internet	docplayer.info	<1%
4	Internet	repositori.kemdikbud.go.id	<1%
5	Internet	e-procurement.telin.co.id	<1%
6	Internet	pustikom.uinssc.ac.id	<1%
7	Internet	sis.binus.ac.id	<1%
8	Internet	blog.w3loker.com	<1%
9	Internet	falseidlepunk.com	<1%
10	Internet	fmipa-unpak.academia.edu	<1%
11	Internet	islamicmarkets.com	<1%



Lampiran 8. Formulir Penggunaan Perangkat Kecerdasan Artifisial (AI)

Nama : Jose Andreas Lie
 NIM : 00000067097
 Email : jose.andreas1@student.umn.ac.id
 Program Studi : Informatika

No	Nama Tool	Alamat Web/Url	Prompt	Tanggal Akses	Media Output
1	Gemini	gemini.google.com	Dalam sistem HRIS modern, jelaskan perbedaan antara data statis dan data transaksional. . .	01 Agustus 2025	<p>♦ Data Master HRIS (Human Resource Information System), penilaian antara data statis dan transaksi dalam sebuah sistem untuk memudahkan pengambilan keputusan yang efektif dan akurat.</p> <p>Berikut adalah penjelasan mendalam mengenai perbedaan serta peran imporsip:</p> <ul style="list-style-type: none"> 1. Perbedaan Data Statis vs Data Transaksi: Sistem HRIS yang menggunakan teknologi terpadu berfungsi untuk memudahkan pengambilan keputusan dalam operasi bisnis. Data statis merupakan data yang tidak berubah seiring waktu, sementara data transaksi adalah "data yang selalu berubah" di sepanjang waktu. <p>Data Master (Modul Kontrak Master Data): Data ini bersifat tetap atau arus yang berada dalam jangka waktu lama. Data ini memang referensi dalam sistem HRIS. <ul style="list-style-type: none"> - Data Diri: Data pribadi, kewarganegaraan, identitas, status page (IP/TP), durasi kontrak, dan rincian pembayaran. - Kontrak: Kontrak kerja, perjanjian kerja, surat perintah (seperti keranjang gigi), dan surat pengunduran diri. <p>Data Transaksi (Modul Lembur & Absensi): Data ini bersifat dinamis, berubah seiring hasil dari aktivitas surga besar. <ul style="list-style-type: none"> - Id Card: Nomor karyawan, jadwal shift, tindakan, pengalaman cuti, kinerjanya, dan pengalaman kerja. - Kinerjanya: Transaksi secara real time di sistem HRIS untuk input karyawannya, misalnya tanda tangan atau pengesahan. </p> <p>Analisis Pengaruh Payroll Automation terhadap Kepatuhan Sosial: Analisis ini menunjukkan bahwa penggunaan sistem payroll otomatis dapat meningkatkan keterwakilan gender dalam pengambilan keputusan dan meningkatkan keterwakilan gender dalam pengambilan keputusan.</p> <ul style="list-style-type: none"> - Standarisasi Perhitungan Sistem memudahkan semua karyawan di suatu organisasi mendapatkan penghasilan yang sama. - Peningkatan Akurasi: Autopilot kalkulator membantu menghindari kesalahan manusia dalam perhitungan. - Transaksi Data: Transaksi data yang digital yang mempermudah kewajiban para SA dan DPH dalam melaksanakan tugas mereka. <p>3 Risiko Hukum yang Dilemahkan Sistem Payroll Digital: Dengan bantuan sistem terpadu, perusahaan dapat mengeliminasi atau mereduk risiko hukum berikut:</p> </p>
2	Gemini	gemini.google.com	Analisis pengaruh payroll automation dalam meningkatkan kepatuhan terhadap regulasi . . .	01 Agustus 2025	<p>♦ Observasi pengaruh payroll automation dalam memberikan data untuk mempermudah pengambilan keputusan dalam sistem HRIS. Sistem ini memungkinkan perusahaan untuk memenuhi persyaratan berbagai sasaran kinerja kerja di Indonesia.</p> <p>Data kinerja reguler yang kompleks seperti UU Cipta Kerja, Peraturan Pemerintah (PP) No. 34/2021, dan peraturan perundang-undangan lainnya yang memerlukan faktor-faktor yang berpengaruh pada kinerja kerja.</p> <p>Analisis Pengaruh Payroll Automation terhadap Kepatuhan Sosial: Analisis ini menunjukkan bahwa penggunaan sistem payroll otomatis dapat meningkatkan keterwakilan gender dalam pengambilan keputusan dan meningkatkan keterwakilan gender dalam pengambilan keputusan.</p> <ul style="list-style-type: none"> - Standarisasi Perhitungan Sistem memudahkan semua karyawan di suatu organisasi mendapatkan penghasilan yang sama. - Peningkatan Akurasi: Autopilot kalkulator membantu menghindari kesalahan manusia dalam perhitungan. - Transaksi Data: Transaksi data yang digital yang mempermudah kewajiban para SA dan DPH dalam melaksanakan tugas mereka. <p>3 Risiko Hukum yang Dilemahkan Sistem Payroll Digital: Dengan bantuan sistem terpadu, perusahaan dapat mengeliminasi atau mereduk risiko hukum berikut:</p>
3	Gemini	gemini.google.com	Jelaskan peran kritis Modul Kontrak Karyawan sebagai data master dalam HRIS . . .	05 Agustus 2025	<p>♦ Sistem informasi HRIS (Modul Kontrak Karyawan) memiliki peran penting sebagai "Single Source of Truth" (Sumber tunggal kebenaran). Tersebut modul berfungsi yang baik, data di modul ini sangat akurat dan dapat dipercaya angka tersebut tidak berasa.</p> <p>Berikut adalah analisis peran kritis serta manfaat standarisasi dan sentralisasi data kontrak:</p> <ul style="list-style-type: none"> 1. Peran Kritis Modul Kontrak sebagai Data Master: Modul Kontrak Sistem terpadu merupakan dokumen digital (PDF), maklumat penting perusahaan yang mendukung operasi sistem HRIS secara keseluruhan. - Standarisasi Perhitungan Sistem memudahkan semua karyawan di suatu organisasi mendapatkan penghasilan yang sama. - Peningkatan Akurasi: Autopilot kalkulator membantu menghindari kesalahan manusia dalam perhitungan. - Transaksi Data: Transaksi data yang digital yang mempermudah kewajiban para SA dan DPH dalam melaksanakan tugas mereka. <p>2. Manfaat Sentralisasi dan Standarisasi Data Kontrak: Data sentralisasi dan standarisasi data kontrak memudahkan pengambilan keputusan dalam sistem HRIS.</p>
4	Gemini	gemini.google.com	Jelaskan dengan detail konsep "single source of truth" dalam HRIS . . .	05 Agustus 2025	<p>♦ Konsep Single Source of Truth (SSOT) dalam HRIS berarti data adalah sebuah arsitektur data yang mendukung operasi sistem HRIS secara keseluruhan. Data ini harus akurat dan relevan di seluruh sistem dan database untuk mendukung fungsi dan tujuan sistem.</p> <p>Data sentralisasi, data ini berfungsi real time dan terpusat di server penyedia layanan, sehingga tidak ada redundansi data yang berkaitan dengan department atau bidang.</p> <p>1. Bagaimana SSOT Mengintegrasikan Data Sistem? Data Sistem terdiri dari department HR yang memiliki data di departemen Finance, HR, Sales, dan Marketing. Data ini dimulai dari data dasar seperti nama, alamat, telepon, dan WhatsApp. Selain itu, data ini dimulai dari data dasar seperti nama, alamat, telepon, dan WhatsApp. Selain itu, data ini dimulai dari data dasar seperti nama, alamat, telepon, dan WhatsApp.</p> <ul style="list-style-type: none"> - Akurasi Multikanal: Setiap departemen HR, Finance, dan Marketing menggunakan database yang berbeda-beda, sehingga data mereka tidak selaras. - Sistem yang Fleksibel: Analisis dan pengambilan keputusan memerlukan data dari modul Employee Self-Service, data di modul payroll atau ledger yang tidak tersedia langsung pada perusahaan. - Integrasi Data: Tidak ada lagi pertumbuhan menggunakan "Data mana yang benar?" jika sistem mencantumkan data di seluruh sistem dan database. <p>2. Meningkatkan Akurasi Perhitungan Payroll Otomatis: Algoritma payroll otomatis berfungsi untuk kalkulasi total pembayaran input berdasarkan</p>