

**LAPORAN KERJA PRAKTEK
CV. ZULIYAN TRIMULTI PACIFIC**

**PERANCANGAN BACK-END APLIKASI ABSENSI DAN
PENJADWALAN KARYAWAN DI CV ZULIYAN TRIMULTI
PACIFIC**

MUHAMMAD NOVAL HIDAYAT

6304191167



**PROGRAM STUDI REKAYASA PERANGKAT LUNAK
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BENGKALIS
BENGKALIS**

2023

LAPORAN KERJA PRAKTEK
CV. ZULIYAN TRIMULTI PACIFIC

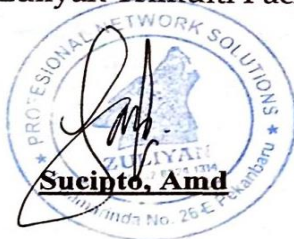
Ditulis sebagai salah satu syarat untuk menyelesaikan Kerja Praktek

MUHAMMAD NOVAL HIDAYAT

NIM. 6304191167

Bengkalis, 15 Juli 2023

Direktur
CV. Zuliyon Trimulti Pacific



Dosen Pembimbing Program Studi
Rekayasa Perangkat Lunak

A handwritten signature in blue ink, appearing to be "Depandi Enda".

Depandi Enda, M.Kom
NIP. 199005222019031010

Disetujui
Ka. Prodi Rekayasa Perangkat Lunak
Politeknik Negeri Bengkalis



Fajri Profesio Putra, M.Cs
NIP. 198805072015041003

KATA PENGANTAR

Alhamdulillahirobbil Alamin. Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan karunia-Nya sehingga mampu menyelesaikan Laporan Kerja Praktek ini. Sehingga dalam kesempatan ini, tak lupa juga ucapan terima kasih kepada kedua orang tua yang telah banyak memberikan dorongan berupa financial serta semangat yang diberikan dari awal hingga selesainya laporan ini. Selanjutnya tidak lupa pula ucapan terima kasih kepada pihak-pihak yang mendukung dalam penyelesaian laporan kerja praktek ini antara lain:

1. Bapak Johny Custer,ST.,MT selaku Direktur Politeknik Negeri Bengkalis.
2. Bapak Kasmawi, M.Kom Selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Bengkalis.
3. Bapak Fajri Profesio Putra, M.Cs Selaku Ketua Program Studi Rekayasa Perangkat Lunak Politeknik Negeri Bengkalis.
4. Ibu Lidya wati, M.Kom Selaku Koordinator Kerja Praktek dari Prodi Rekayasa Perangkat Lunak.
5. Bapak Depandi Enda, M.Kom selaku Dosen Pembimbing Kerja Praktek.
6. Bapak Sucipto Sebagai direktur sekaligus pembimbing Kerja Praktek di CV.Zuliyani Trimulti Pacific.
7. Bapak Muhammad Iqbal Sebagai wakil direktur sekaligus pembimbing Kerja Praktek di CV.Zuliyani Trimulti Pacific.

Bengkalis, 15 Juli 2023



Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan dan Manfaat.....	2
1.3 Luaran Proyek	2
BAB II GAMBARAN UMUM PERUSAHAAN	3
2.1 Sejarah Singkat Perusahaan.....	3
2.2 Visi dan Misi Perusahaan	3
2.3 Struktur Organisasi.....	4
2.4 Ruang.....	5
BAB III BIDANG PEKERJAAN SELAMA KERJA PRAKTEK	6
3.1 Bidang Kerja atau Bagian Tempat Kerja Praktek	6
3.2 Tugas Umum	6
3.3 Pembuatan Aplikasi.....	9
BAB IV PERANCANGAN DAN IMPLEMENTASI	12
4.1 Prosedur Pembuatan Sistem Alat dan Solusi.	12
4.2 Perancangan dan Implementasi	15
4.3 Implementasi Sistem	23
4.4 Dampak Implementasi Sistem.....	39
BAB V PENUTUP	40
5.1 Kesimpulan.....	40
5.2 Saran	40
DAFTAR PUSTAKA	41

DAFTAR GAMBAR

Gambar 2. 1 Struktur Organisasi.....	4
Gambar 3. 1 Mempelajari CMS Wordpress.....	6
Gambar 3. 2 Mempelajari NestJs	7
Gambar 3. 3 Mempelajari Angular	8
Gambar 3. 4 merancang aplikasi dengan elementor wordpress	9
Gambar 3. 5 Merancang desain usecase diagram aplikasi absensi	10
Gambar 3. 6 Merancang desain database absensi karyawan.....	10
Gambar 3. 7 Pengkodean backend aplikasi absensi dengan NestJs	11
Gambar 4. 1 Prosedur dalam Pembuatan Sistem	12
Gambar 4. 2 Usecase Diagram.....	16
Gambar 4. 3 Activity Diagram Login	18
Gambar 4. 4 Activity Diagram Mengelola data karyawan	19
Gambar 4. 5 Activity diagram mengelola absensi karyawan.....	20
Gambar 4. 6 Activity diagram mengelola jadwal karyawan.....	21
Gambar 4. 7 Activity diagram mengelola data shift	22
Gambar 4. 8 Tampilan Generate Resource	23
Gambar 4. 9 membuat module karyawan	24
Gambar 4. 10 Menjalankan API dengan thunder client.....	28
Gambar 4. 11 Mengaktifkan Swagger	29
Gambar 4. 12 Tampilan Swagger module karyawan	30
Gambar 4. 13 Tampilan Swagger Module Auth	31
Gambar 4. 14 menambahkan authorization	32
Gambar 4. 15 Routing Auth	33
Gambar 4. 16 Routing User	34
Gambar 4. 17 Routing karyawan	35
Gambar 4. 18 Routing shift.....	37
Gambar 4. 19 Routing penjadwalan.....	38

DAFTAR TABEL

Tabel 4. 1 Jadwal Pelaksanaan.....	14
Tabel 4. 2 Fungsi-Fungsi Usecase	17

BAB I PENDAHULUAN

1.1 Latar Belakang

Kerja Praktek (KP) merupakan serangkaian kegiatan yang meliputi pemahaman teori/konsep ilmu pengetahuan yang diaplikasikan dalam pekerjaan sesuai profesi bidang studi. KP dapat menambah wawasan, pengetahuan dan skill mahasiswa, serta mampu menyelesaikan persoalan-persoalan ilmu pengetahuan sesuai dengan teori yang mereka peroleh di bangku kuliah. (Buku Panduan Kerja Praktek Politeknik Negeri Bengkalis, 2017)

Kerja Praktek merupakan salah satu program yang diperuntukkan bagi mahasiswa agar mengetahui dan memperoleh pengalaman di dunia kerja. Kerja Praktek ini juga merupakan salah satu syarat yang harus dipenuhi oleh mahasiswa khususnya Politeknik Negeri Bengkalis. Selain itu kerja praktek bertujuan agar mahasiswa dapat mengetahui profesi serta ruang lingkup pekerjaan sesuai dengan program studi.

Berhubungan dengan hal di atas, maka penulis melaksanakan kerja praktek di sebuah perusahaan yang berada di Kota Pekanbaru, yaitu di CV. Zuliyani Trimulti Pacific. Waktu kerja praktek yang dilaksanakan berlangsung selama 4 bulan di perusahaan ini. Hal ini berdasarkan ketentuan untuk prodi Rekayasa Perangkat Lunak yang mana pelaksanaan KP dilaksanakan minimal 16 minggu. Zuliyani merupakan sebuah perusahaan kontraktor IT, yang bergerak dibidang jasa pengadaan *software* berskala kecil maupun besar.

Selama melaksanakan kerja Praktek di CV. Zuliyani Trimulti Pacific ini, penulis mengerjakan beberapa tugas yang diberikan oleh pembimbing lapangan. Diantaranya yaitu merancang desain *user interface*, mempelajari cms wordpress, membuat website dengan elementor wordpress, mempelajari frontend angular dan *backend* NestJs, penerapan *framework* Angular, dll. Selain itu kami juga diberi kesempatan untuk membuat

membuat suatu proyek yaitu aplikasi absensi dan penjadwalan karyawan. Yang dimana di perusahaan tersebut, proses pelaksanaan absensi serta penjadwalan masih dilakukan secara konvensional hal ini dapat mempengaruhi kedisiplinan karyawan serta kurangnya kinerja di perusahaan tersebut. Solusi yang dapat dilakukan oleh penulis yakni melakukan analisa, perencanaan, dan perancangan sistem untuk menyelesaikan permasalahan yang terjadi tersebut.

1.2 Tujuan dan Manfaat

Tujuan dari pembuatan aplikasi absensi dan penjadwalan karyawan ini agar mempermudah HRD dalam merekap absensi serta membuat jadwal masuk dan keluar untuk karyawan.

1.3 Luaran Proyek

Berdasarkan rencana kegiatan yang telah disusun maka target luaran yang diharapkan setelah pelaksanaan Kerja Praktek adalah aplikasi absensi dan penjadwalan karyawan ini di implementasikan dan hasil dari kegiatan ini juga akan digunakan untuk menyusun luaran berupa seminar hasil dari KP (Kerja Praktek).

BAB II

GAMBARAN UMUM PERUSAHAAN

2.1 Sejarah Singkat Perusahaan

Zuliyen Profesional Network Solutions adalah perusahaan yang berdiri pada Mei 2017, dibawah naungan perusahaan CV. ZULIYAN TRIMULTI PACIFIC yang beralamat di JL. Samarinda No.26 E, Tangkerang Utara, Pekanbaru - Riau. Zuliyen bergerak di bidang jasa pengadaan *software* dan *hardware* dengan berbagai macam kebutuhan berskala kecil maupun besar, selain itu juga mengadakan perawatan sistem yang berjangka panjang. Adapun anggota yang dimiliki tersebut berjumlah 2 orang yaitu Sucipto dan Muhammad Iqbal.

2.2 Visi dan Misi Perusahaan

Visi dan Misi dari CV. ZULIYAN TRIMULTI PACIFIC adalah sebagai berikut:

2.2.1 Visi

Menjadi perusahaan IT Management terdepan dalam pengembangan dan kemajuan dunia teknologi di Indonesia.

2.2.2 Misi

- Membangun Management kerja perusahaan yang efisien, terpercaya, dan bisa diandalkan.
- Membangun kerja sama dan melakukan sinergi saling menguntungkan dengan partner bisnis
- Mengembangkan Sistem Informasi dan Perangkat Teknologi ketingkat yang lebih baik dan dapat dipercaya sebagai salah satu solusi dalam management suatu instansi berskala kecil maupun besar
- Membangun jaringan seluas-luasnya dengan personal maupun perusahaan lain khususnya dalam bidang teknologi

2.3 Struktur Organisasi



Gambar 2. 1 Struktur Organisasi

Struktur organisasi CV. Zuliyon Trimulti Pacific Dibentuk atas kesepakatan Bersama yang dibentuk pada tahun 2017, berikut uraian dari jabatan organisasi tersebut:

1. Konsultan

Konsultan adalah penyedia tempat untuk CV.Zuliyon Trimulti Pacific yang bertugas sebagai penasihat di CV. Zuliyon tersebut.

2. CEO dan Programmer

CEO atau Chief Executive Officer sekaligus menjadi programmer CV. Zuliyon yang bertugas mengurus bagian internal perusahaan yaitu melayani client dan sekaligus programmer.

3. CMO dan Dokumentasi

CMO atau Chief Marketing Officer mempunyai tugas untuk memasarkan jasa dan sekaligus yang mengatur bagian external perusahaan yang berkaitan dengan surat menyurat atau pengurusan dokumen lainnya.

2.4 Ruang

CV. Zuliyani Trimulti Pacific bergerak dalam pembuatan jasa web, aplikasi, dan server yang bergerak dalam bidang Pendidikan, perusahaan kecil atau pun besar. Adapun beberapa produk yang dijalankan pada CV.Zuliyani Trimulti Pacific ini sebagai berikut :

1. **Business & Management System** : Untuk pengelolaan usaha retail, bengkel, kuliner, dan café (*hardware* dan *software*).
2. **Sistem Antrian Integrasi** : Sistem antrian bagi perusahaan-perusahaan atau layanan pemerintahan yang melayani public (*hardware* dan *software*).
3. **Medical Education System** : Sistem pengelolaan akademik perkuliahan dibidang kesehatan yang mencakup pengelolaan Blok, Osce, CBT, Bank Soal, E-Skripsi, E-Office, Rotasi, Face Identification, Absensi, dan Penjadwalan (*hardware* dan *software*).

BAB III BIDANG PEKERJAAN SELAMA KERJA PRAKTEK

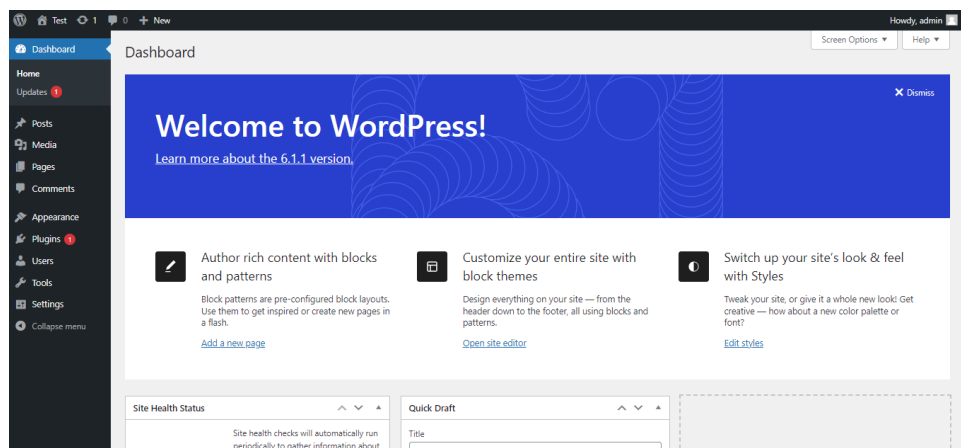
3.1 Bidang Kerja atau Bagian Tempat Kerja Praktek

Pelaksanaan Kerja Praktek di CV. Zuliyon Trimulti Pacific ditempatkan pada bagian program. Tujuan penempatan pada bagian program untuk mengenal situasi di perusahaan, sehingga dapat mempelajari program atau sistem yang dipakai perusahaan serta menyelesaikan masalah yang ada dengan sistem tersebut.

3.2 Tugas Umum

Tugas umum adalah tugas yang diberikan oleh tempat kerja praktek kepada seluruh mahasiswa kerja praktek. Selama melaksanakan kerja praktek di CV. Zuliyon Trimulti Pacific mahasiswa diberikan beberapa materi yang berkaitan dengan pembuatan *website*, Adapun materitugas yang diberikan yaitu :

3.3.1 Mempelajari CMS Wordpress

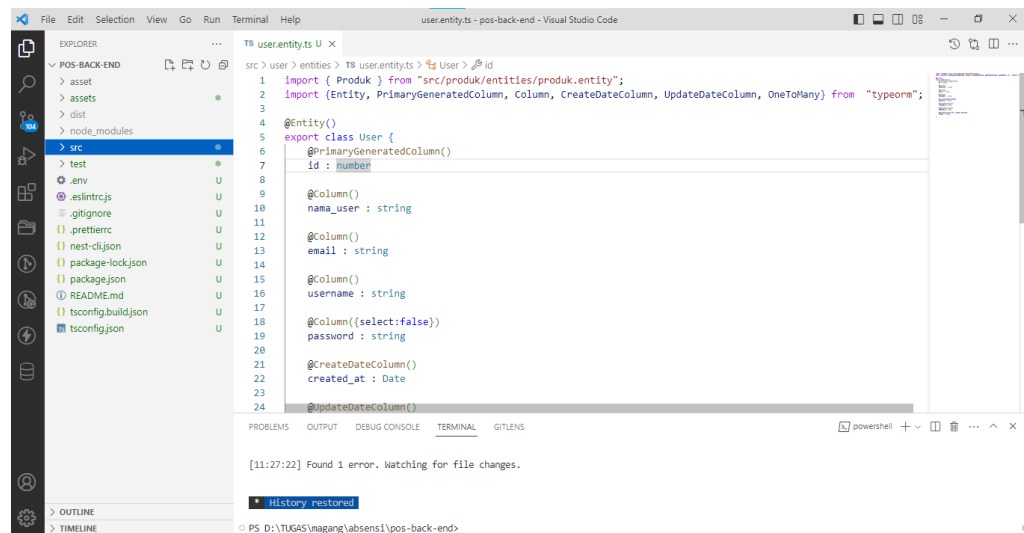


Gambar 3. 1 Mempelajari CMS Wordpress

WordPress adalah sebuah sistem manajemen konten (*Content Management System* atau CMS) yang populer dan *open-source*. CMS ini dibangun menggunakan bahasa pemrograman PHP dan didukung oleh basis data MySQL. WordPress awalnya dikembangkan sebagai platform blogging, namun seiring waktu, ia telah berkembang menjadi sebuah CMS

yang fleksibel yang memungkinkan pengguna untuk membuat dan mengelola berbagai jenis situs web, termasuk blog, situs berita, situs perusahaan, toko online, dan banyak lagi.

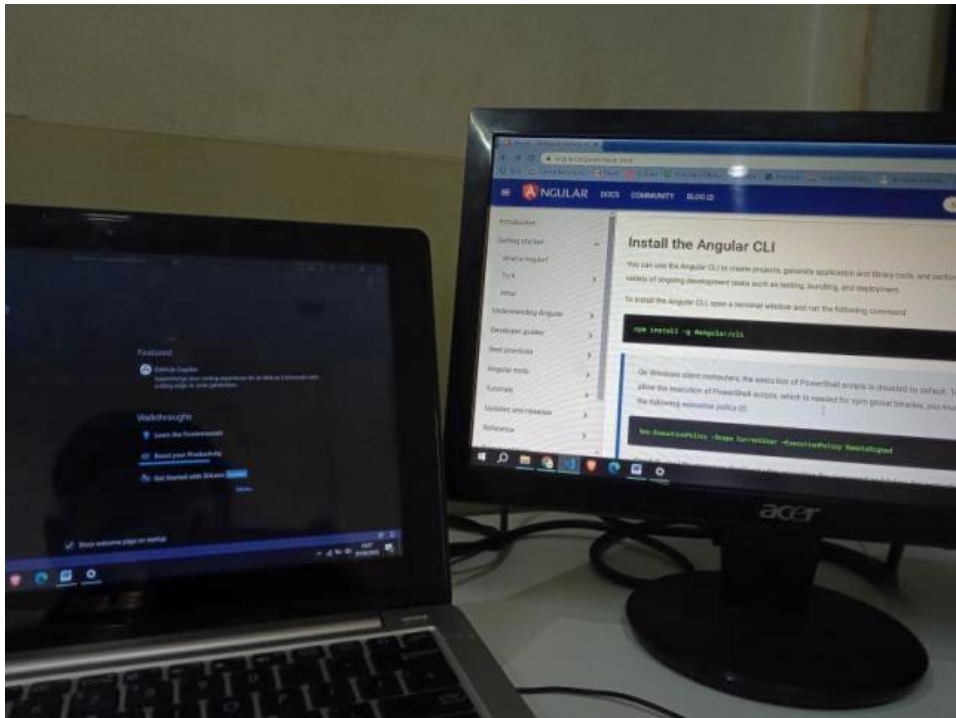
3.3.2 Mempelajari NestJs



Gambar 3. 2 Mempelajari NestJs

NestJS adalah sebuah *framework* server-side open-source yang digunakan untuk mengembangkan aplikasi berbasis Node.js. Dibangun menggunakan TypeScript, NestJS memungkinkan pengembang untuk memanfaatkan fitur-fitur dari bahasa tersebut dan menyediakan struktur yang terorganisir untuk mengelola proyek dengan skala besar. *Framework* ini mengadopsi arsitektur berbasis modul, yang memungkinkan pemisahan logika aplikasi menjadi komponen-komponen terpisah untuk meningkatkan skalabilitas dan pemeliharaan kode program.

3.2.3 Mempelajari Angular



Gambar 3. 3 Mempelajari Angular

Angular adalah *platform* pengembangan aplikasi web *open-source* yang dikembangkan oleh Google. Awalnya dikenal sebagai AngularJS, platform ini menggunakan bahasa TypeScript dan dirancang untuk memudahkan pengembangan aplikasi web berbasis klien yang kompleks dan interaktif. Angular menyediakan seperangkat alat dan fitur, termasuk sistem templating, pengikatan data dua arah, manajemen routing, dan dukungan untuk pembuatan komponen yang memudahkan para pengembang dalam membangun aplikasi web yang dinamis dan responsive.

3.3 Pembuatan Aplikasi

Berikut adalah proses pembuatan aplikasi :

3.3.1 Merancang aplikasi dengan elementor wordpress

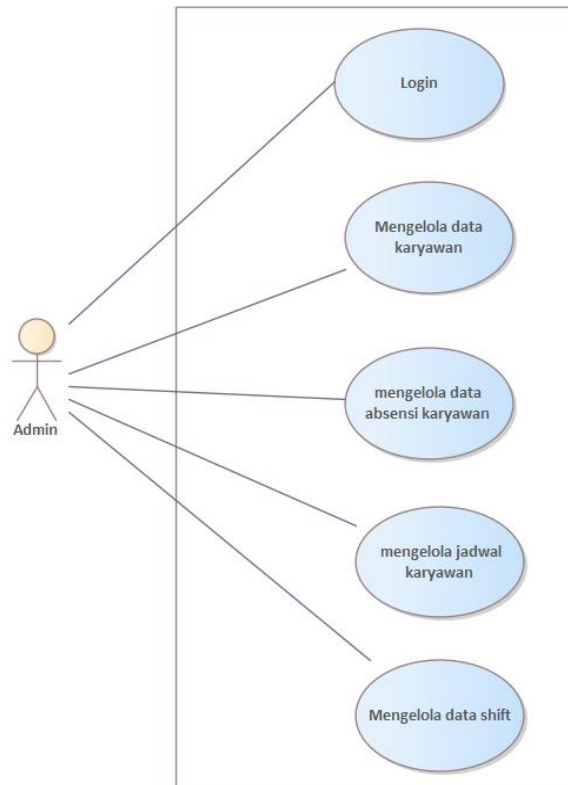


Gambar 3. 4 merancang aplikasi dengan elementor wordpress

Elementor adalah sebuah *page builder* yang beroperasi sebagai plugin di platform CMS WordPress. Dirancang sebagai editor drag-and-drop, Elementor memungkinkan pengguna untuk dengan mudah membuat dan mengatur tata letak halaman web tanpa harus menulis kode. Dengan memanfaatkan elementor wordpress ini, mahasiswa merancang aplikasi website profil perusahaan (*company profile*) yang mana aplikasi ini digunakan media penyampaian informasi bagi pihak perusahaan serta agar pihak lainnya bisa mengetahui tentang perusahaan zuliyon.

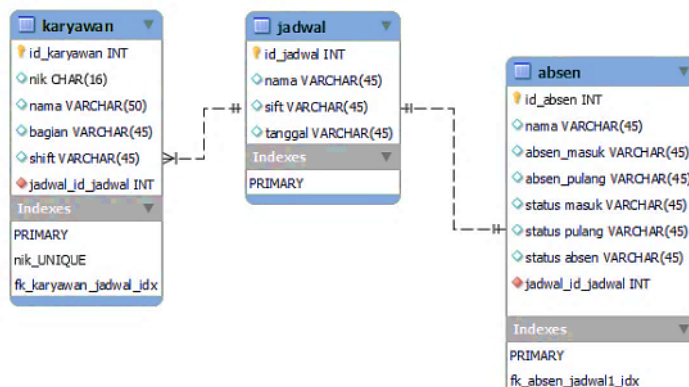
3.3.2 Merancang desain usecase diagram aplikasi absensi

Sebelum merancang aplikasi terlebih dahulu merancang desain use case aplikasi absensi, yang dimana usecase aplikasi absensi untuk admin.



Gambar 3. 5 Merancang desain usecase diagram aplikasi absensi

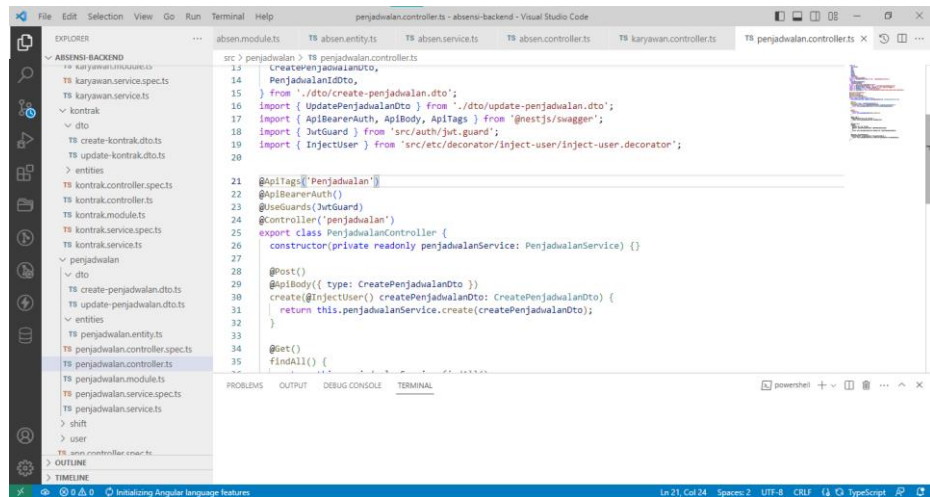
3.3.3 Merancang desain database absensi karyawan



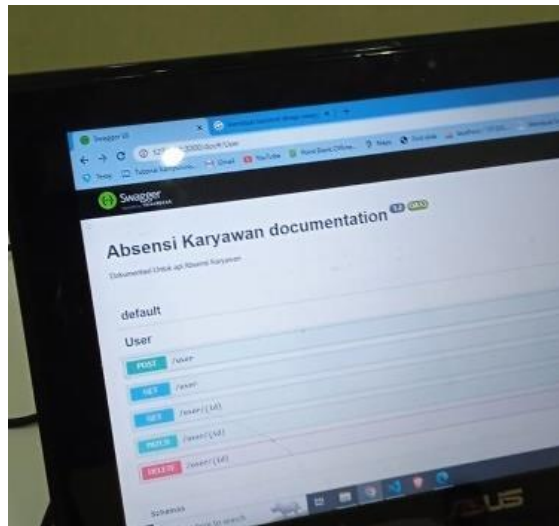
Gambar 3. 6 Merancang desain database absensi karyawan

Selain perancangan *usecase diagram*, juga merancang desain database absensi karyawan yang dimana desain database ini digunakan untuk menyimpan data berupa data karyawan, data absensi, jadwal karyawan dan lain-lain.

3.3.3 Pengkodean backend aplikasi absensi dengan NestJs



```
13 createPenjadwalanDto,
14   PenjadwalanIDto,
15   } from './dto/create-penjadwalan.dto';
16   import { UpdatePenjadwalanDto } from './dto/update-penjadwalan.dto';
17   import { ApiBearerAuth, ApiBody, ApiTags } from '@nestjs/swagger';
18   import { JwtGuard } from 'src/auth/jwt.guard';
19   import { InjectUser } from 'src/etc/decorator/inject-user/inject-user.decorator';
20
21   @ApiTags('Penjadwalan')
22   @ApiBearerAuth()
23   @UseGuards(JwtGuard)
24   @Controller('penjadwalan')
25   export class PenjadwalanController {
26     constructor(private readonly penjadwalanService: PenjadwalanService) {}
27
28     @Post()
29     @ApiBody({ type: CreatePenjadwalanDto })
30     create(@InjectUser() createPenjadwalanDto: CreatePenjadwalanDto) {
31       return this.penjadwalanService.create(createPenjadwalanDto);
32     }
33
34     @Get()
35     findAll() {
```



Gambar 3. 7 Pengkodean backend aplikasi absensi dengan NestJs

Membuat sistem bagian *backend* untuk aplikasi absensi dan penjadwalan karyawan yang sudah dianalisa dari proses pengumpulan data sampai desain. Pengerjaan sistem *backend* aplikasi absensi dan penjadwalan karyawan ini menggunakan framework dari Javascript yaitu Nestjs. Pengkodean back-end aplikasi absensi ini meliputi auth, karyawan, absensi, jadwal, shift dan juga kontrak. Dalam pengerjaan tersebut menggunakan visual studio code serta memerlukan extensi thunder client dan database client untuk melihat database. Selain itu juga menggunakan swagger ui untuk tampilan dokumentasi API.

BAB IV

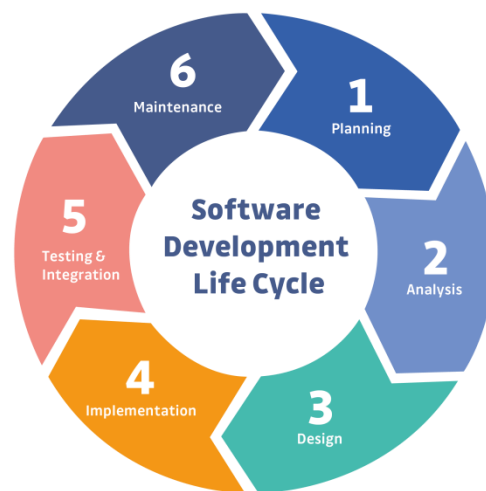
PERANCANGAN DAN IMPLEMENTASI

4.1 Prosedur Pembuatan Sistem Alat dan Solusi.

Penulis menentukan metodologi dalam penyelesaian masalah pembuatan sistem, yang dibutuhkan pengguna berdasarkan masalah yang ada serta alat dan solusi dalam pembuatan sistem ini yaitu sebagai berikut :

4.1.1 Prosedur dalam Pembuatan Sistem

Pada pembuatan Aplikasi Absensi dan Penjadwalan karyawan ini, Penulis menggunakan metode pengembangan SDLC (*Software Development Life Cycle*). Metode SDLC, atau yang dikenal sebagai Siklus Hidup Pengembangan Perangkat Lunak, merujuk pada serangkaian proses esensial yang membentuk kerangka kerja dalam pembuatan dan transformasi sistem perangkat lunak. SDLC merupakan siklus hidup pengembangan perangkat lunak yang terdiri dari analisa sistem, spesifikasi kebutuhan sistem, perencanaan system, pengembangan system, pengujian serta pemeliharaan sistem (Hasanah & Indriawan, 2021).



Gambar 4. 1 Prosedur dalam Pembuatan Sistem

Adapun tahapan dalam metode SDLC ini adalah sebagai berikut:

a. Tahap perencanaan (*planning*)

Tahap perencanaan merupakan tahap awal untuk menentukan kebutuhan pengguna. Sesuai dengan tools atau kakas bantu yang digunakan oleh penulis yaitu UML.

b. Tahap Analisis (*analysis*)

Tahap analisis merupakan tahap menganalisis lebih lanjut kebutuhan pengguna yang telah ditetapkan dengan merancang usecase, merancang objek maupun hubungan antara objek yang lain.

c. Tahap Perencanaan (*design*)

Pada tahap ini, yaitu membuat atau merancang desain dari tahapan analisis yang telah didapatkan sebelumnya.

d. Tahap Implementasi (*implementation*)

Pada tahap ini, penulis mengimplemantasikan atau mengeksekusi system yang telah dirancang sebelumnya menjadi sebuah aplikasi dengan melakukan proses pengkodean atau coding.

e. Tahap Pengujian (*testing*)

Pada tahap ini dilakukan pengujian terhadap sistem atau aplikasi yang telah dibuat apakah sudah sesuai dengan kebutuhan pengguna atau belum. Jika belum, proses selanjutnya bersifat iteraif, artinya kembali ke tahap sebelumnya. Tujuan dari pengujian ini adalah mengurangi cacat program.

f. Tahap Pemeliharaan

Tahap Pemeliharaan atau perawatan ini merupakan tahap yang ke enam, yaitu dimana mulai pengoperasian sistem dan perbaikan bug-bug kecil. Untuk pengembangan aplikasi ini pengembang tidak sampai pada tahap pemeliharaan ini hanya sampai di pengujian.

4.1.2 Metodologi Pengumpulan data

Metodologi pengumpulan data yang digunakan penulis yaitu menggunakan teknik wawancara. Teknik wawancara yaitu menanyakan langsung kepada narasumber perihal masalah yang ditemui di lapangan. Berikut ini merupakan proses wawancara yang dilakukan.

1. Mengumpulkan data berdasarkan masalah-masalah yang telah dijelaskan oleh narasumber
2. Mengevaluasi hal-hal yang berkaitan dengan sistem.
3. Menganalisis data yang telah didapatkan dari narasumber tersebut kemudian diolah menjadi perencanaan sistem

4.1.3 Tahapan dan Jadwal Pelaksanaan

Tabel 4. 1 Jadwal Pelaksanaan

No	Uraian Kegiatan	Bulan																
		Bulan 1				Bulan 2				Bulan 3				Bulan 4				
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
1	<i>Planning</i>																	
2	<i>Analysis</i>																	
3	Perancangan																	
4	<i>Implementation</i>																	
5	<i>Testing</i>																	

4.2 Perancangan dan Implementasi

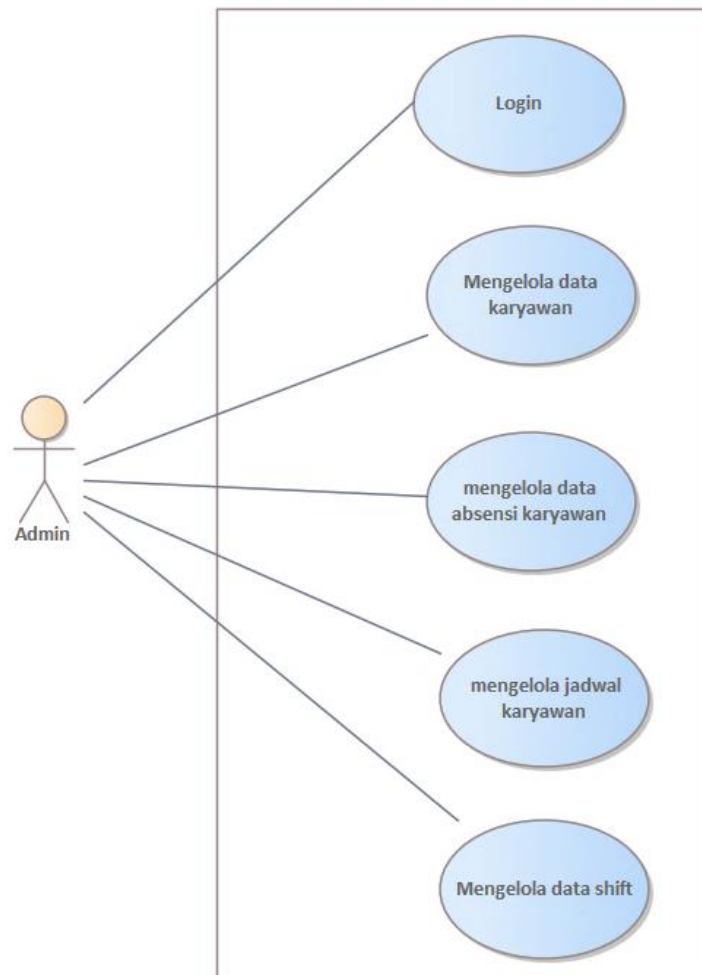
4.2.1 Analisis Data

Saat melakukan analisis data terdapat beberapa teknik analisis atau pengumpulan data sebelum melakukan perancangan sistem. Dalam hal ini penulis melakukan analisis data dengan teknik wawancara, yaitu dengan tatap muka serta Tanya jawab secara langsung bersama narasumber dari perusahaan, hal ini bertujuan agar informasi didapatkan secara langsung sehingga kebutuhan dari system yang akan dibangun lebih jelas.

4.2.2 Rancangan Sistem/Alat/Solusi

Rancangan ini dibuat guna menguraikan hasil rancangan sistem yang telah dibuat. Rancangan sistem disajikan dalam bentuk *usecase* dan *activity diagram*. Hasil dari rancangan tersebut sebagai berikut :

4.2.2.1 Usecase Diagram Aplikasi Absensi dan Penjadwalan Karyawan



Gambar 4. 2 Usecase Diagram

Pada gambar diatas yaitu menggambarkan *usecase diagram* Aplikasi Absensi dan Penjadwalan Karyawan. Aplikasi ini memiliki satu aktor yaitu admin. admin bisa melakukan login untuk masuk ke sistem, mengelola data karyawan, mengelola data absensi karyawan, mengelola jadwal karyawan, dan mengelola data shift. Fungsi *usecase* dapat dilihat pada tabel berikut ini.

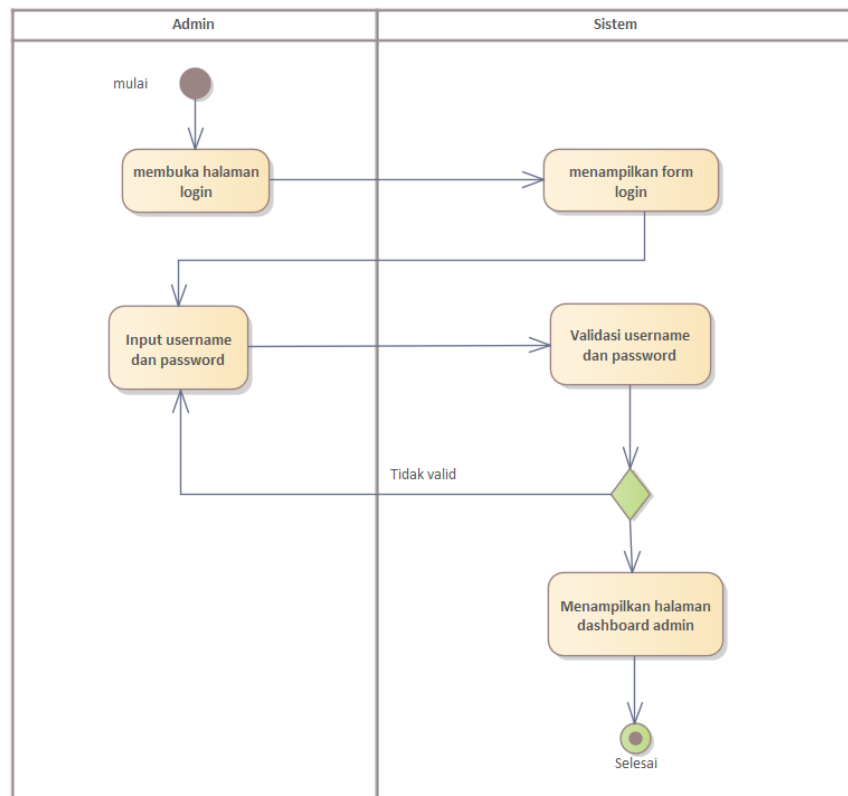
Tabel 4. 2 Fungsi-Fungsi Usecase

No	Usecase	Fungsi
1.	Login	Admin melakukan login agar bisa masuk ke sistem
2.	Mengelola data karyawan	Admin bisa menambah, edit serta hapus data karyawan
3.	Mengelola data absensi karyawan	Admin bisa menambah, edit serta hapus data absensi karyawan
4.	Mengelola data jadwal	Admin bisa menambah, edit, serta hapus data jadwal kerja karyawan
5.	Mengelola data shift	Admin bisa menambah, edit, serta hapus data shift/pembagian jadwal kerja karyawan

4.2.2.2 Activity Diagram

Adapun rancangan *activity diagram* yang dibuat berdasarkan *usecase* sebagai berikut :

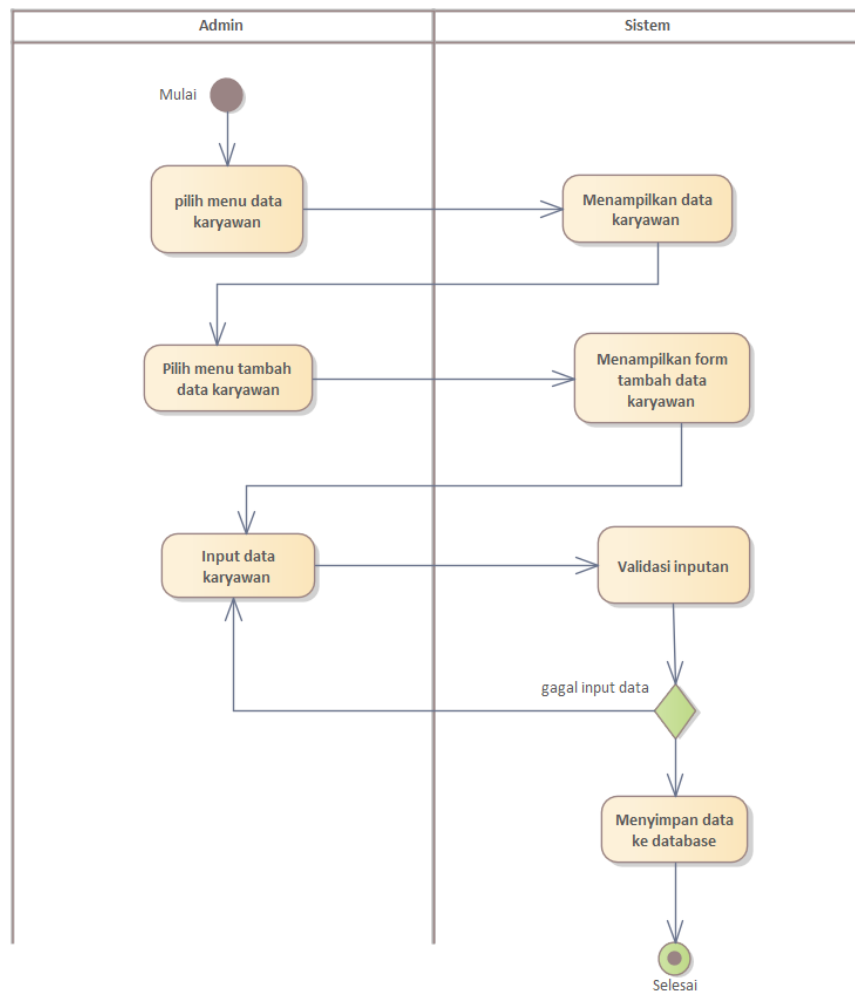
1. Login



Gambar 4. 3 Activity Diagram Login

Pada diagram *activity* diatas, pertama pengguna akan mengakses halaman login, selanjutnya sistem akan menampilkan form login. Admin menginputkan data berupa username dan password, setelah user menginputkan data tersebut, selanjutnya sisitem akan memvalidasi inputan Admin jika berhasil maka sistem akan mengarahkan ke halaman dashboard sistem, jika gagal maka akan kembali ke halaman login.

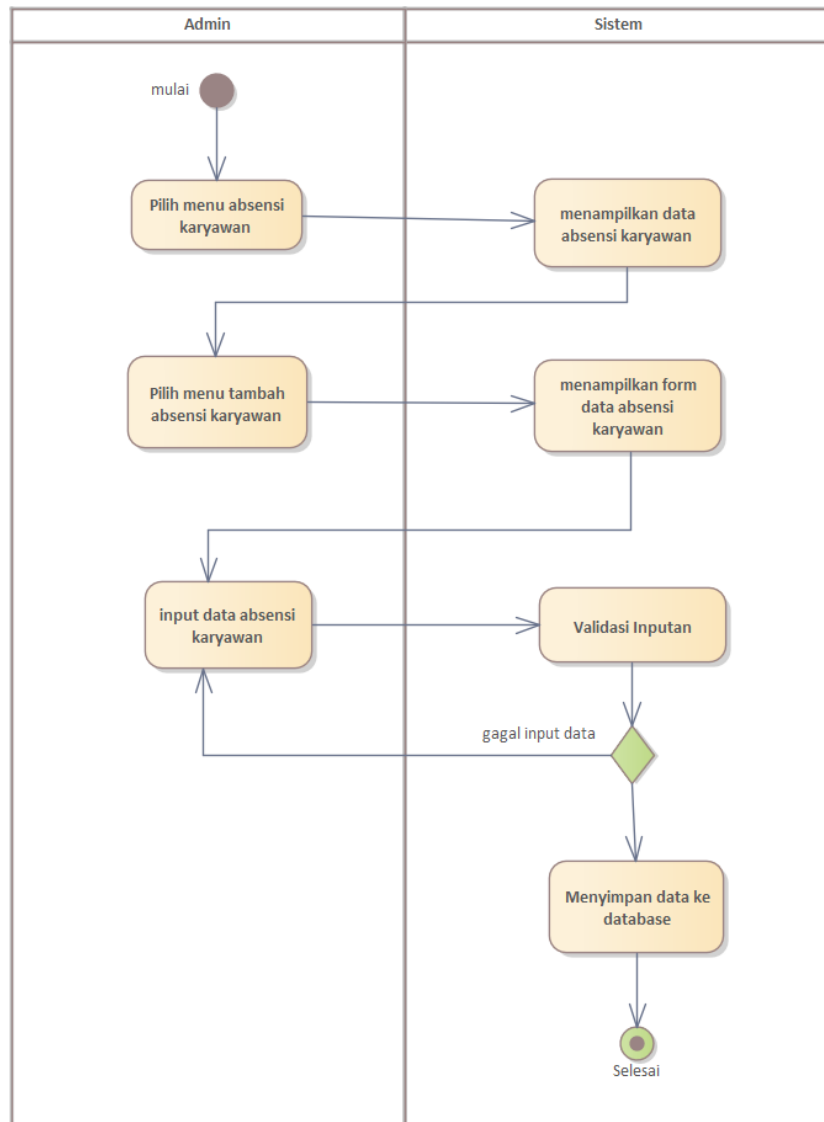
2. Mengelola data karyawan



Gambar 4. 4 Activity Diagram Mengelola data karyawan

Pertama admin memilih halaman data karyawan, selanjutnya sistem akan menampilkan halaman data karyawan, admin dapat menambah data karyawan dengan memilih menu tambah karyawan, selanjutnya sistem akan menampilkan form karyawan dan akan diisi oleh admin, setelah diisi oleh admin sistem akan memvalidasi inputan oleh admin untuk selanjutnya disimpan ke dalam database.

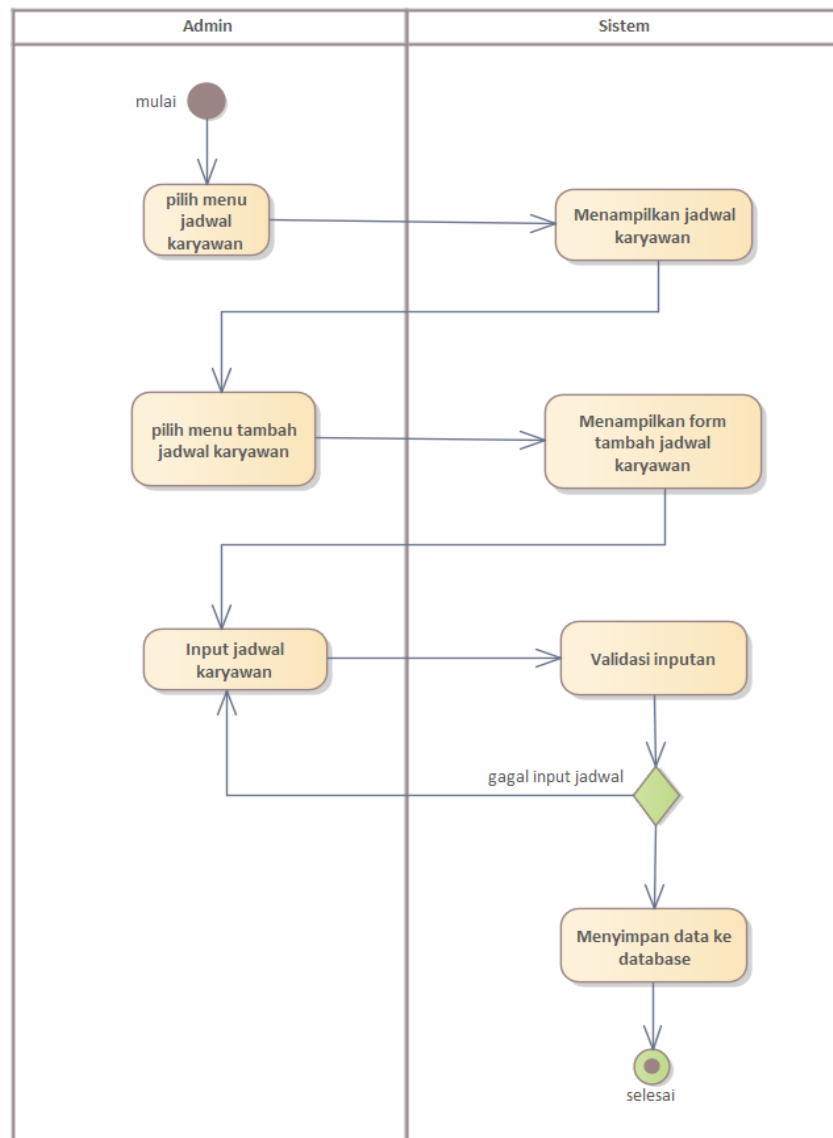
3. Mengelola absensi karyawan



Gambar 4. 5 Activity diagram mengelola absensi karyawan

Pada *activity diagram* diatas, pertama admin memilih menu absensi, selanjutnya sistem akan menampilkan data absensi, admin memilih menu tambah absensi, sistem akan menampilkan halaman form absensi yang akan diisi, selanjutnya admin mengisi data absensi. Setelah terinput maka sistem akan memvalidasi data inputan oleh admin. Jika berhasil maka data akan tersimpan di database.

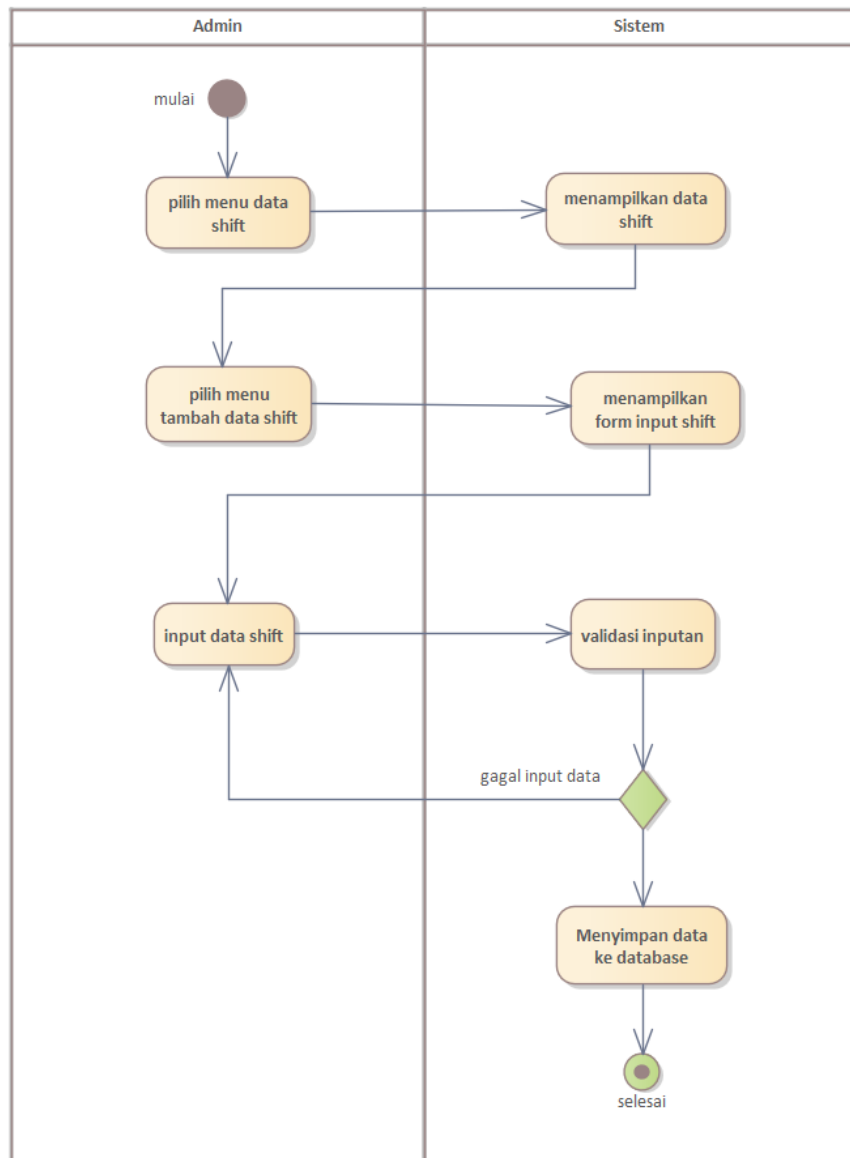
4. Mengelola jadwal karyawan



Gambar 4. 6 Activity diagram mengelola jadwal karyawan

Pada diagram *activity diagram* diatas merupakan diagram untuk mengelola penjadwalan karyawan pertama admin memilih halaman data karyawan, selanjutnya sistem akan menampilkan halaman penjadwalan. Admin dapat menambah data penjadwalaan dengan memilih menu tambah data penjadwalan, setelah data penjadwalan di input sistem akan memvalidasi inputan, jika berhasil maka sistem akan menyimpan inputan tersebut kedalam database.

5. Mengelola data shift



Gambar 4. 7 Activity diagram mengelola data shift

Pada *activity diagram* diatas ada untuk mengelola data shift karyawan, pertama admin memilih halaman data shift, selanjutnya sistem akan menampilkan halaman data shift, selanjutnya admin menambah datashift dengan memilih menu tambah data shift, sistem selanjutnya akan memvalidasi data shift yang sebelumnya ditambahkan admin, jika berhasil maka data yang sebelumnya ditambahkan akan di simpan ke database.

4.3 Implementasi Sistem

Pada implementasi sistem, terdapat beberapa hasil dokumentasi dari API aplikasi absensi dan penjadwalan karyawan dengan menggunakan swagger UI dari NestJS. Yang mana *routing* url API ini nantinya akan digunakan untuk integrasi dengan *front-end*. Langkah-langkah pembuatan dokumentasi API dengan Swagger UI yaitu sebagai berikut.

1. Langkah 1: Persiapan Awal

Pastikan memiliki Node.js dan npm diinstal pada sistem. Jika belum, instal Node.js dari situs resmi dan npm akan terpasang bersamanya.

2. Langkah 2: Instalasi NestJS CLI

Jika belum memiliki NestJS CLI, instal dengan menjalankan perintah berikut melalui terminal atau command prompt:

```
npm install -g @nestjs/cli
```

3. Langkah 3: Membuat Proyek Baru

Buat proyek baru dengan perintah berikut:

```
nest new nama-proyek
```

Jika di aplikasi ini berarti :

```
nest new absensi backend
```

4. Langkah 4: Membuat Module Karyawan

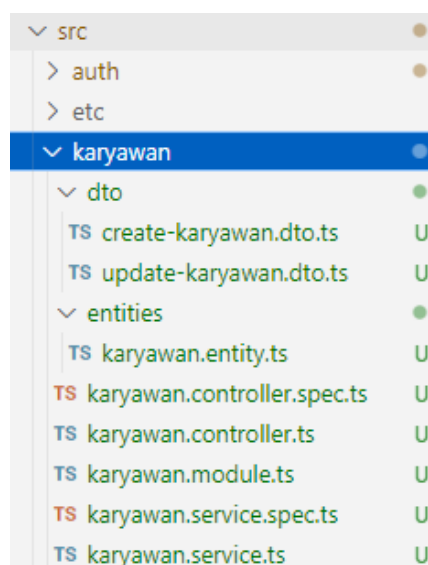
Buat modul untuk membuat module karyawan dengan melakukan generate resource. Perintahnya sebagai berikut:

```
nest g res karyawan
```

```
C:\pos-back-end>nest g res karyawan
? What transport layer do you use? REST API
? Would you like to generate CRUD entry points? Yes
CREATE src/karyawan/karyawan.controller.ts (967 bytes)
CREATE src/karyawan/karyawan.controller.spec.ts (596 bytes)
CREATE src/karyawan/karyawan.module.ts (268 bytes)
CREATE src/karyawan/karyawan.service.ts (663 bytes)
CREATE src/karyawan/karyawan.service.spec.ts (474 bytes)
CREATE src/karyawan/dto/create-karyawan.dto.ts (34 bytes)
CREATE src/karyawan/dto/update-karyawan.dto.ts (180 bytes)
CREATE src/karyawan/entities/karyawan.entity.ts (25 bytes)
UPDATE src/app.module.ts (1206 bytes)
```

Gambar 4. 8 Tampilan Generate Resource

Maka pada folder `absensi_backend/src` akan ada folder baru yang berisi module karyawan yang telah lengkap didalamnya ada entities, dto, controller, module dan service. entities dalam konteks NestJS adalah kelas-kelas yang mewakili struktur data atau model dalam aplikasi. dto adalah kelas yang digunakan untuk mengatur bagaimana data dikirimkan antara client (misalnya, *frontend*) dan server (NestJS). Controller yaitu bagian yang bertanggung jawab untuk menerima permintaan dari klien (seperti permintaan HTTP dari browser atau API) dan memberikan respons yang sesuai. Controller ini bertindak sebagai perantara antara permintaan masuk dan logika bisnis yang sesungguhnya di dalam layanan (service). Module untuk mengatur komponen-komponen aplikasi Anda. Ini membantu dalam mengelompokkan fungsionalitas terkait bersama-sama, membagi aplikasi menjadi bagian-bagian yang terpisah dan mudah dikelola serta service adalah seperti membuat "penolong" untuk mengurus pekerjaan khusus di aplikasi. Service membantu menjaga agar urusan-urusan bisnis tidak tercampur dengan bagian yang menangani permintaan dari pengguna atau tampilan. berikut tampilan dari folder karyawan :



Gambar 4. 9 membuat module karyawan

5. Langkah 5: Mengisi entity karyawan

Didalam *entity* karyawan, set *field* apa saja yang terdapat didalam tabel karyawan. Berikut *source code* dari aplikasi absensi dan penjadwalan karyawan ini :

```
@Entity()
export class KaryawanDto {
  @ApiModelProperty()
  @PrimaryGenerateColumn()
  id_karyawan: number;

  nik: number;

  @IsString()
  nama: string;

  @IsOptional()
  status: number;

  @IsString()
  jabatan: string;
}
```

Setelah mengisi data karyawan, coba jalankan aplikasi untuk membuat tabel yang telah dibuat dengan menjalankan perintah sebagai berikut :

```
npm run start:dev
```

6. Langkah 6: setting dto

Secara default, struktur dto hampir sama dengan entity. Buka file `createKaryawanDto` yang ada di dalam folder `dto`, kemudian buatlah field-field apa saja yang terdapat dalam tabel. Berikut *source code* dari file `create-karyawan.dto.ts` :

```
export class CreateKaryawanDto {
  id_karyawan?: number;
  nik: number;
  nama: string;
  status: number;
  jabatan: string;
}
```

7. Langkah 7: mendaftarkan entity ke module user

Untuk mendaftarkan entity, pertama buka file `karyawan.module.ts` kemudian import `TypeOrmModule` dan panggil entity `karyawan`. Untuk *source code* nya sebagai berikut :

```
@Module({
  imports: [TypeOrmModule.forFeature([karyawan])],
  controllers: [KaryawanController],
  providers: [KaryawanService],
})
```

8. Langkah 8: Registerkan repository karyawan

Untuk meregisterkan repository `karyawan` di service, buka file `karyawan.service.ts` kemudian buat *constructor* untuk meregisterkan repository. Untuk *source code* nya sebagai berikut :

```
export class KaryawanService {
  constructor(
    @InjectRepository(Karyawan) private karyawanRepo:
    Repository<Karyawan>,
  ) {}
  create(createKaryawanDto: CreateKaryawanDto) {
    console.log(createKaryawanDto)
    return this.karyawanRepo.save(createKaryawanDto);
  }
  findAll() {
    return this.karyawanRepo.find();
  }
  findOne(id: number) {
    return this.karyawanRepo.findOne({ where: { id_karyawan:
    id } });
  }
  update(id: number, updateKaryawanDto: UpdateKaryawanDto) {
    updateKaryawanDto.id_karyawan = id;
    return this.karyawanRepo.save(updateKaryawanDto);
  }
  async remove(id: number) {
    let karyawan = await this.karyawanRepo.findOne({
      where: { id_karyawan: id },
    });
  }
}
```



```

    });
    return this.karyawanRepo.remove(karyawan);
  }
}

```

9. Langkah 9: Definisikan Endpoints

Buka file `karyawan.controller.ts` yang telah dihasilkan dan tentukan endpoint-endpoint yang diinginkan dengan dekorator seperti `@Get()`, `@Post()`, dll.

Berikut *source code* endpoint dari `karyawanController`

```

export class KaryawanController {
  constructor(private readonly karyawanService:
KaryawanService) {}

  @Post()
  create(@Body() createKaryawanDto: CreateKaryawanDto) {
    return this.karyawanService.create(createKaryawanDto);
  }

  @Get()
  findAll() {
    return this.karyawanService.findAll();
  }

  @Get('/:id')
  findOne(@Param('id') id: string) {
    return this.karyawanService.findOne(+id);
  }

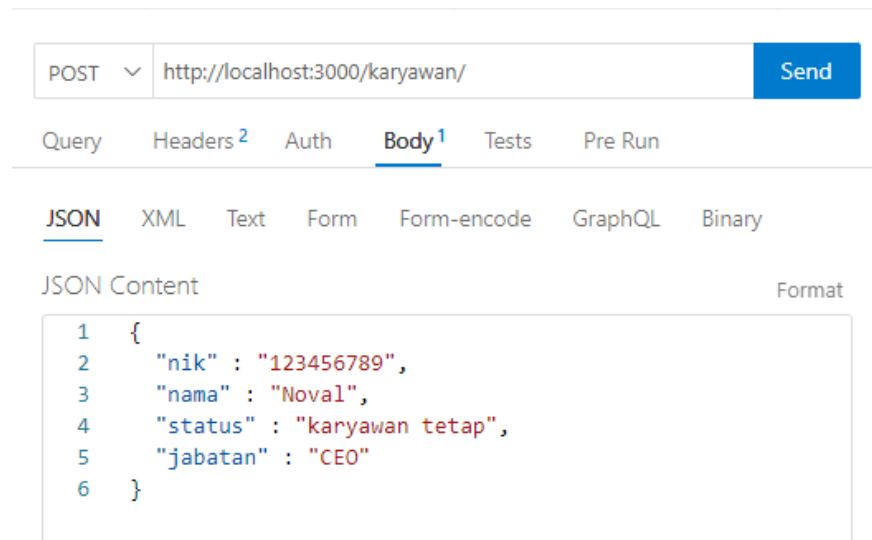
  @Patch('/:id')
  update(
    @Param('id') id: string,
    @Body() updateKaryawanDto: UpdateKaryawanDto,
  ) {
    return this.karyawanService.update(+id,
updateKaryawanDto);
  }

  @Delete('/:id_karyawan')
  remove(@Param() id_karyawan: KaryawanIdDto) {
    return
this.karyawanService.remove(id_karyawan.id_karyawan);
  }
}

```

```
}  
}
```

Setelah menentukan endpoint coba jalankan dengan extensi thunder client yang bisa di instal jika menggunakan code editor vs code.



Gambar 4. 10 Menjalankan API dengan thunder client

10. Langkah 10: Mengaktifkan Swagger

Swagger digunakan untuk pengujian serta dokumentasi API. Untuk instalasi swagger yaitu harus menginstal paket @nestjs/swagger untuk integrasi dengan Swagger UI:

```
npm install --save @nestjs/swagger swagger-ui-express
```

11. Langkah 11: Mengaktifkan Swagger

Untuk mengaktifkan swagger, pertama-tama buka file main.ts kemudian tambahkan kode untuk config swagger. Untuk *source code* nya dapat dilihat pada dibawah ini :

```
async function bootstrap() {  
  const app = await NestFactory.create(AppModule);  
  .setTitle('Absensi Karyawan documentation')  
  .setDescription('Dokumentasi Untuk api Absensi Karyawan')  
  .setVersion('1.2')  
  .build();
```

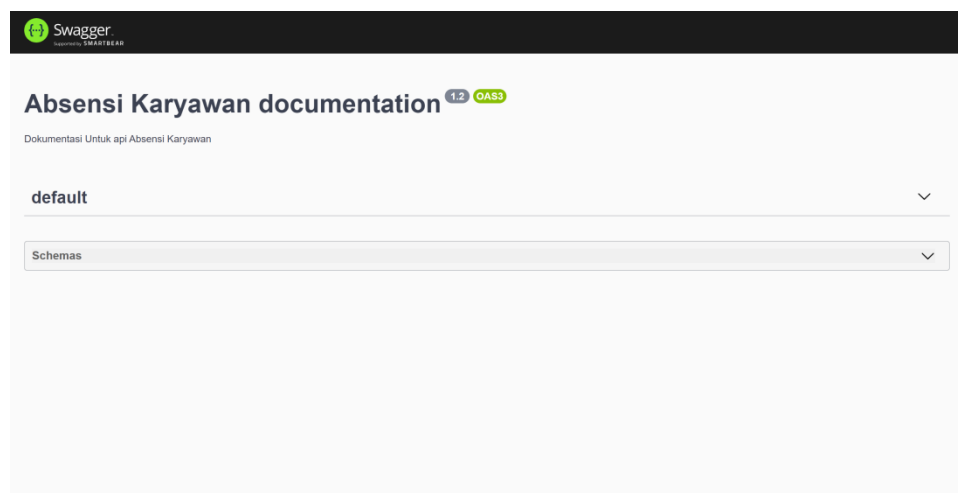
```

const configCustomSwagger: SwaggerCustomOptions = {
  swaggerOptions: { docExpansion: 'none' },
};

const doc = SwaggerModule.createDocument(app,
configSwagger);
SwaggerModule.setup('doc', app, doc, configCustomSwagger);

await app.listen(3000);
}

```



Gambar 4. 11 Mengaktifkan Swagger

12. Langkah 12: Menambahkan Anotasi pada Controller karyawan

Untuk menampilkan dokumentasi karyawan module pada swagger perlu ditambakkannya anotasi pada controller yaitu dengan cara membuka file karyawan.controller.ts lagi dan tambahkan dekorator Swagger pada setiap endpoint yang ingin di dokumentasikan. Berikut *source code* untuk penambahan anotasi pada controller karyawan dengan menambahkan @ApiTags.

```

@ApiTags('Karyawan')
@Controller('karyawan')
export class KaryawanController {
  constructor(private readonly karyawanService:
KaryawanService) {}
  @Post()
  create(@Body() createKaryawanDto: CreateKaryawanDto) {

```

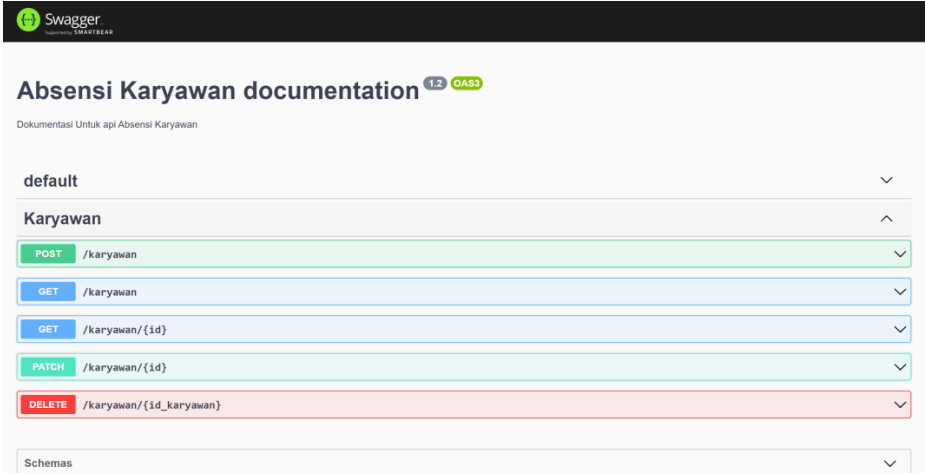
```
    return this.karyawanService.create(createKaryawanDto);
  }

  @Get()
  findAll() {
    return this.karyawanService.findAll();
  }

  @Get('/:id')
  findOne(@Param('id') id: string) {
    return this.karyawanService.findOne(+id);
  }

  @Patch('/:id')
  update(
    @Param('id') id: string,
    @Body() updateKaryawanDto: UpdateKaryawanDto,
  ) {
    return this.karyawanService.update(+id,
updateKaryawanDto);
  }

  @Delete('/:id_karyawan')
  remove(@Param() id_karyawan: KaryawanIdDto) {
    return
this.karyawanService.remove(id_karyawan.id_karyawan);
  }
}
```



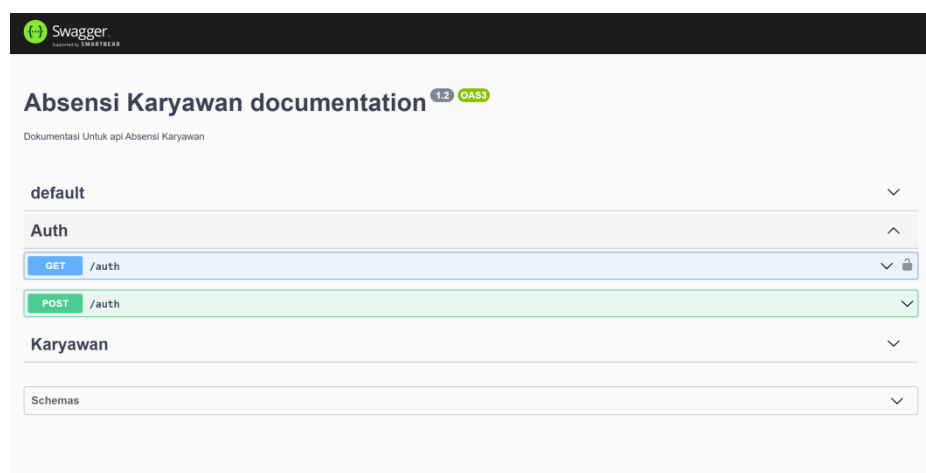
Gambar 4. 12 Tampilan Swagger module karyawan

Selanjutnya untuk menambahkan dokumentasi module lain di Swagger caranya juga sama yaitu anotasikan di controllernya. Contoh disini menambahkan dokumentasi module auth.

```
@ApiTags('Auth')
@Controller('auth')
export class AuthController {
  constructor(private readonly authService: AuthService) {}

  @Get()
  @ApiBearerAuth()
  @UseGuards(JwtGuard)
  cekUser(@Request() req) {
    return req.user;
  }

  @Post()
  async login(@Body() authDto: AuthDto) {
    let user = await this.authService.cekUser(
      authDto.username,
      authDto.password,
    );
    return this.authService.generateToken({ id:
user.id_user });
  }
}
```



Gambar 4. 13 Tampilan Swagger Module Auth

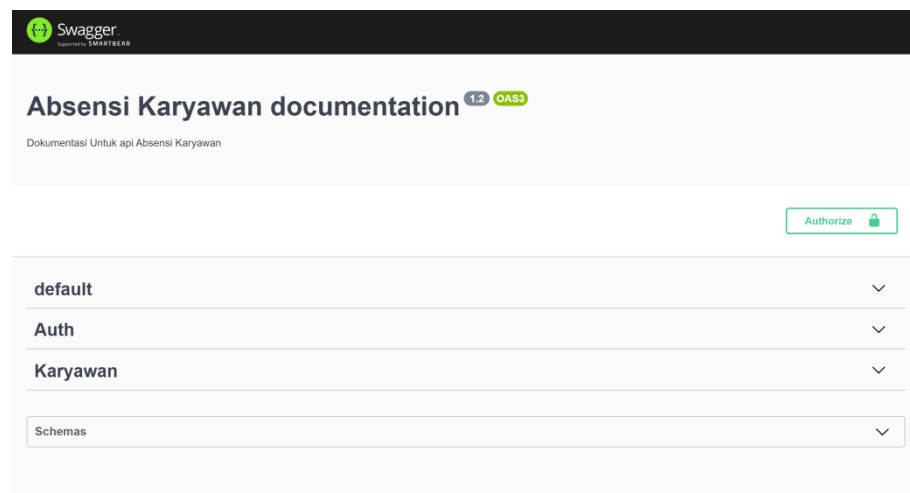
13. Langkah 13: Menambahkan bearer auth

Bearer auth digunakan untuk tempat memasukkan token ketika pengguna sudah login dari module auth agar pengguna bisa mengakses modul-modul yang terkunci. Langkahnya yaitu :

- Buka file main.ts lalu tambahkan `.addBearerAuth()` pada config swagger

```
const configSwagger = new DocumentBuilder()
  .setTitle('Absensi Karyawan documentation')
  .setDescription('Dokumentasi Untuk api Absensi Karyawan')
  .setVersion('1.2')
  .addBearerAuth()
  .build();
```

Maka untuk tampilannya akan ada *icon* gembok dengan tulisan *Authorize* dibagian atas dokumentasi API.



Gambar 4. 14 menambahkan authorization

Gambar diatas merupakan tampilan dari dokumentasi API setelah menambahkan bearer auth. Dengan begitu selesai sudah langkah-langkah dari pembuatan dokumentasi API dengan menerapkan swagger UI. Berikut ini adalah hasil implementasi keseluruhan dari aplikasi absensi dan penjadwalan karyawan ini.

1. Auth

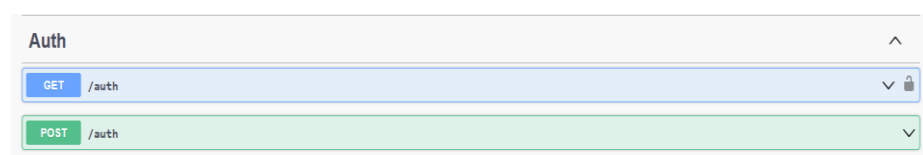
Auth digunakan untuk *authorhization* admin yang menggunakan *bearer* yang didapatkan dari proses login. Didalan *authorization* ini menyimpan *value* yang berisi token yang akan digunakan untuk login.

Source code:

```
export class AuthController {
  constructor(private readonly authService: AuthService) {}

  @Get()
  @ApiBearerAuth()
  @UseGuards(JwtGuard)
  cekUser(@Request() req) {
    return req.user;
  }

  @Post()
  async login(@Body() authDto: AuthDto) {
    let user = await this.authService.cekUser(
      authDto.username,
      authDto.password,
    );
    return this.authService.generateToken({ id: user.id_user
});
  }
}
```



Gambar 4. 15 Routing Auth

2. User

User digunakan untuk mendaftarkan atau membuat akun agar admin bisa mengakses aplikasi frontend nantinya.

Source code:

```
export class UserController {
  constructor(private readonly userService: UserService) {}

  @Post()
  create(@Body() createUserDto: CreateUserDto) {
```

```

    return this.userService.create(createUserDto);
  }

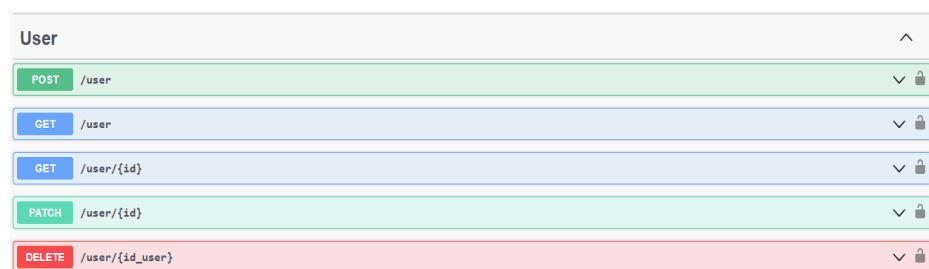
  @Get()
  findAll() {
    return this.userService.findAll();
  }

  @Get('/:id')
  findOne(@Param('id') id: string) {
    return this.userService.findOne(+id);
  }

  @Patch('/:id')
  update(@Param('id') id: string, @Body() updateUserDto:
  UpdateUserDto) {
    return this.userService.update(+id, updateUserDto);
  }

  @Delete('/:id_user')
  remove(@Param() id_user: UserIdDto) {
    return this.userService.remove(id_user.id_user);
  }
}

```



Gambar 4. 16 Routing User

3. Karyawan

Karyawan yang disini dimaksud adalah yaitu admin mendata karyawan yang bekerja di perusahaan. Setiap *routing* ini digunakan untuk melakukan tambah data, edit maupun hapus data karyawan yang telah diinput.

Source code:

```
export class KaryawanController {
  constructor(private readonly karyawanService:
KaryawanService) {}
  @Post()
  create(@Body() createKaryawanDto: CreateKaryawanDto) {
    return this.karyawanService.create(createKaryawanDto);
  }

  @Get()
  findAll() {
    return this.karyawanService.findAll();
  }

  @Get('/:id')
  findOne(@Param('id') id: string) {
    return this.karyawanService.findOne(+id);
  }

  @Patch('/:id')
  update(
    @Param('id') id: string,
    @Body() updateKaryawanDto: UpdateKaryawanDto,
  ) {
    return this.karyawanService.update(+id,
updateKaryawanDto);
  }

  @Delete('/:id_karyawan')
  remove(@Param() id_karyawan: KaryawanIdDto) {
    return
this.karyawanService.remove(id_karyawan.id_karyawan);
  }
}
```

Karyawan		^
POST	/karyawan	⌵ 🔒
GET	/karyawan	⌵ 🔒
GET	/karyawan/{id}	⌵ 🔒
PATCH	/karyawan/{id}	⌵ 🔒
DELETE	/karyawan/{id_karyawan}	⌵ 🔒

Gambar 4. 17 Routing karyawan

4. Shift

Shift digunakan untuk menambah atau input data shift yang nantinya data shift ini akan digunakan di form penjadwalan untuk menetapkan jadwal karyawan

Source code:

```
export class ShiftController {
  constructor(private readonly shiftService: ShiftService)
  {}
  @Post()
  @ApiBody({ type: CreateShiftDto })
  create(@InjectUser() createShiftDto: CreateShiftDto) {
    return this.shiftService.create(createShiftDto);
  }

  @Get()
  findAll() {
    return this.shiftService.findAll();
  }

  @Get('/:id')
  findOne(@Param('id') id: string) {
    return this.shiftService.findOne(+id);
  }

  @Patch('/:id')
  @ApiBody({ type: UpdateShiftDto })
  update(
    @Param('id') id: string,
    @InjectUser() updateShiftDto: UpdateShiftDto,
  ) {
    console.log(UpdateShiftDto);
    return this.shiftService.update(+id, updateShiftDto);
  }

  @Delete('/:id_shift')
  remove(@Param() id_shift: ShiftIdDto) {
    return this.shiftService.remove(id_shift.id_shift);
  }
}
```

Shift	
POST	/shift
GET	/shift
GET	/shift/{id}
PATCH	/shift/{id}
DELETE	/shift/{id_shift}

Gambar 4. 18 Routing shift

5. Penjadwalan

Penjadwalan digunakan admin agar bisa menjadwalkan waktu mulai serta berakhirnya jadwal seorang karyawan. Didalam data penjadwalan inilah akan ditetapkan seorang karawan masuk di shift mana.

Source code:

```

export class PenjadwalanController {
  constructor(private readonly penjadwalanService:
PenjadwalanService) {}

  @Post()
  @ApiBody({ type: CreatePenjadwalanDto })
  create(@InjectUser() createPenjadwalanDto:
CreatePenjadwalanDto) {
    return
this.penjadwalanService.create(createPenjadwalanDto);
  }

  @Get()
  findAll() {
    return this.penjadwalanService.findAll();
  }

  @Get('/:id')
  findOne(@Param('id') id: string) {
    return this.penjadwalanService.findOne(+id);
  }

  @Patch('/:id')
  update(
    @Param('id') id: string,
    @Body() updatePenjadwalanDto: UpdatePenjadwalanDto,
  ) {
    return this.penjadwalanService.update(+id,
updatePenjadwalanDto);
  }
}

```

```
}  
  
@Delete('/:id_penjadwalan')  
remove(@Param() id_penjadwalan: PenjadwalanIdDto) {  
    return  
    this.penjadwalanService.remove(id_penjadwalan.id_penjadwalan  
);  
}  
}
```

Penjadwalan		^
POST	/penjadwalan	⌵ 🔒
GET	/penjadwalan	⌵ 🔒 📄
GET	/penjadwalan/{id}	⌵ 🔒
PATCH	/penjadwalan/{id}	⌵ 🔒
DELETE	/penjadwalan/{id_penjadwalan}	⌵ 🔒

Gambar 4. 19 Routing penjadwalan

4.4 Dampak Implementasi Sistem

Dampak implementasi sistem terhadap penulis adalah penulis dapat belajar serta mengukur sejauh mana kemampuan untuk terus mencari dan memecahkan masalah, sedangkan dampak bagi perusahaan adalah perusahaan adalah lebih tersistemnya lagi proses pendataan absensi di perusahaan tersebut.

BAB V PENUTUP

5.1 Kesimpulan

Aplikasi back-end absensi dan penjadwalan ini dirancang dengan bahasa pemrograman TypeScript dengan memanfaatkan *framework* NestJS. Pada perancangan *backend* ini yaitu membuat dokumentasi API untuk nantinya bisa dihubungkan dengan *front-end* agar bisa dilihat oleh *user* atau pengguna. Aplikasi ini diharapkan dapat mempermudah pihak perusahaan dalam mendata absensi dan penjadwalan karyawan.

Perusahaan CV. Zuliyon Trimulti Pacific juga memberikan kesempatan kepada penulis untuk mendapatkan pengalaman kerja, keterampilan, serta tanggung jawab yang belum pernah dipelajari saat kuliah sebelumnya.

5.2 Saran

Adapun saran yang akan disampaikan untuk mahasiswa/i yang nantinya akan melaksanakan kerja praktek agar lebih memperdalam mengenai dasar-dasar dan prinsip-prinsip pemrograman terutama tentang database karena dengan rancangan *database* yang baik akan menghasilkan sistem atau perangkat lunak yang baik dan dapat berkembang. Untuk tempat magang sebaiknya ditempat yang sesuai dengan program studi rekayasa perangkat lunak. salah satunya, CV. Zuliyon Trimulti Pasific yang bergerak di bidang jasa pengadaan *software* dan *hardware*, menggunakan teknologi yang lebih update dalam pengembangan perangkat lunak, dan diajarkan bagaimana pemecahan masalah dalam pengerjaan project dan kerjasama tim yang penting dalam pembuatan suatu perangkat lunak.

DAFTAR PUSTAKA

- Buku Pedoman Laporan Kerja Praktek (KP) Politeknik Negeri Bengkalis, 2017.
- Listiani, L., SM, N. N. F., Hamdani, M. P., & Al Gifari, S. (2023, February). Aplikasi Absensi dan Penjadwalan Karyawan PT. Trans Indonesia Superkoridor Berbasis Web. In *SISITI: Seminar Ilmiah Sistem Informasi dan Teknologi Informasi* (Vol. 12, No. 1, pp. 255-264).
- Hasanah, N., & Indriawan, M. N. (2021, March). Rancangan Aplikasi Batam Travel Menggunakan Metode Software Development Life Cycle (SDLC). In *CoMBInES-Conference on Management, Business, Innovation, Education and Social Sciences* (Vol. 1, No. 1, pp. 925-938).



CV.ZULIYAN TRIMULTI PACIFIC

Jl. Samarinda 26 E, Pekanbaru | Telp : +62 82388102969 | Email : info@zuliyannet.net

SURAT PEMBERITAHUAN

No. C5/SPU/ZTP/II-2023/002

Hal : Surat Pemberitahuan Pengajuan Kerja Praktek (KP)

Ditujukan Kepada

POLITEKNIK NEGERI BENGKALIS

Jl. Bathin Alam, Sungailam, Bengkalis - Riau

Telp : +62766 24566

Menanggapi surat permohonan Kerja Praktek (KP) Nomor : 394/PL31/TU/2023 yang Bpk/Ibu kirimkan sebelumnya, dengan ini **CV. ZULIYAN TRIMULTI PACIFIC (Perusahaan)** sebagai perusahaan yang bergerak dibidang Kontraktor dan Konsultan IT menyatakan bersedia menerima mahasiswa/i untuk Kerja Praktek di perusahaan kami yang berlokasi di Pekanbaru.

Berikut nama dan data mahasiswa/i yang kami terima untuk kegiatan Kerja Praktek di perusahaan kami :

- | | |
|---------------------------|------------------|
| 1. MUHAMMAD NOVAL HIDAYAT | NIM : 6304191167 |
| 2. M. FARID AMIRUL | NIM : 6304191190 |

Untuk informasi lebih lanjut tentang peraturan perusahaan, jadwal KP, dan penyerahan/pengantaran mahasiswa/i KP, kami harapkan kepada Bpk/Ibu untuk menghubungi nomor : +6282388102969 (Bpk. Iqbal).

Demikian surat pemberitahuan ini disampaikan, atas perhatian dan kerjasamanya kami ucapkan terima kasih.





CV.ZULIYAN TRIMULTI PACIFIC

Jl. Samarinda 26 E, Pekanbaru | Telp : +62 82388102969 | Email : info@zulyian.net

SURAT KETERANGAN

No. C6/SPU/ZTP/VII-2023/001

Yang bertanda tangan dibawah ini menerangkan bahwa :

Nama : M. Noval Hidayat
Tempat/Tgl.lahir : Bengkalis, 1 November 2001
Alamat : Jl. Bantan, Gg. A. Rahman

Telah melakukan Kerja Praktek (KP) di perusahaan kami CV. Zulyian Trimulti Pacific, sejak tanggal 13 Maret 2023 sampai dengan 13 Juli 2023 sebagai tenaga Kerja Praktek.

Selama bekerja diperusahaan kami, yang bersangkutan telah menunjukkan ketekunan dan kesungguhan bekerja dengan baik, dan berdedikasi.

Demikian surat keterangan ini diberikan agar dapat dipergunakan sebagaimana mestinya, atas perhatian dan kerjasamanya kami ucapkan terima kasih.









DAFTAR HADIR SEMINAR KP

Nama Mahasiswa : Muhammad Noval Hidayat

NIM : 6304191167

Judul KP : Perancangan Back-End Aplikasi Absensi Dan
Penjadwalan Karyawan Di Cv Zuliyana Trimulti
Pasific

NO	NAMA	JABATAN	PARAF
1.	Indah Wahyuni	TI	
2.	Nia Putri Agustina	TI	
3.	Malyda Aulia	RPL	
4.	Nursafika	RPL	
5.	Maryatul Kippiyah	Mahasiswa	
6.	Sigit Kurniawan	TI	

**PENILAIAN DARI PERUSAHAAN KERJA
PRAKTEKCV. ZULIYAN TRIMULTI PACIFIC**

Nama : Muhammad Noval Hidayat
NIM : 6304191167
Program Studi : Sarjana Terapan Rekayasa Perangkat Lunak
Politeknik Bengkalis

No.	Aspek Penilaian	Bobot	Nilai
1.	Disiplin	20%	90
2.	Tanggung- jawab	25%	95
3.	Penyesuaian diri	10%	90
4.	Hasil Kerja	30%	90
5.	Perilaku secara umum	15%	85
	Total Jumlah (1+2+3+4+5)	100%	90

Keterangan :
Nilai : Kriteria
81 – 100 : Istimewa
71 – 80 : Baik sekali
66 – 70 : Baik
61 – 65 : Cukup Baik
56 – 60 : Cukup

Catatan :

- Disiplinkan diri dan tingkatkan lagi ilmu dalam programming

- Selesaikan bahan secepatnya.

Pekanbaru, 15 Juli 2023



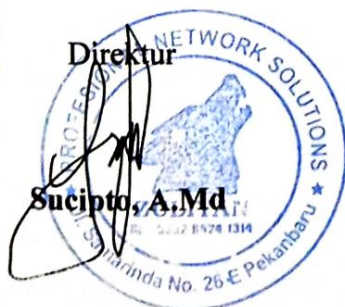
Sucipto, Amd

Direktur Cv. Zuliyon Trimulti Pacific

**ABSENSI KERJA PRAKTEK (KP)
CV. ZULIYAN TRIMULTI PACIFIC**

No	Nama	Bulan	Hari						Keterangan
			Senin	Selasa	Rabu	Kamis	Juma't	Sabtu	
1	M. Noval Hidayat	Minggu 1	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
2	M. Noval Hidayat	Minggu 2	✓	✓	libur	✓	✓	✓	
	M. Farid Amirul		✓	✓	libur	✓	✓	✓	
3	M. Noval Hidayat	Minggu 3	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
4	M. Noval Hidayat	Minggu 4	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
5	M. Noval Hidayat	Minggu 5	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
6	M. Noval Hidayat	Minggu 6	✓	✓	libur	B	✓	R	
	M. Farid Amirul		✓	✓	libur	B	✓	R	
7	M. Noval Hidayat	Minggu 7	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
8	M. Noval Hidayat	Minggu 8	S	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	S	✓	✓	✓	✓	
9	M. Noval Hidayat	Minggu 9	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
10	M. Noval Hidayat	Minggu 10	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
11	M. Noval Hidayat	Minggu 11	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
12	M. Noval Hidayat	Minggu 12	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
13	M. Noval Hidayat	Minggu 13	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
14	M. Noval Hidayat	Minggu 14	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
15	M. Noval Hidayat	Minggu 15	✓	✓	✓	S	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	
16	M. Noval Hidayat	Minggu 16	✓	✓	✓	✓	✓	✓	
	M. Farid Amirul		✓	✓	✓	✓	✓	✓	

Mengetahui,




Wakil Direktur

Muhammad Iqbal, S.Kom

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 1
Tanggal : 13-18 Maret 2023

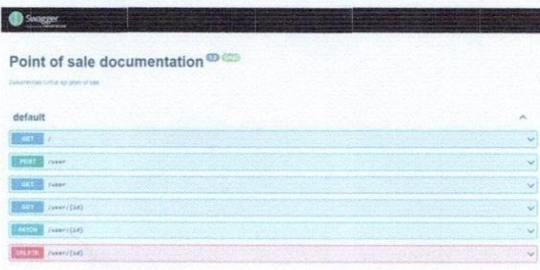

NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	<ol style="list-style-type: none">1. Pembahasan projek aplikasi website perusahaan (company profile)2. Mempelajari framework nestJs3. Merancang desain aplikasi website perusahaan dengan figma	Bapak Iqbal	

NO	Gambaran Kerja	KETERANGAN
1		Membahas projek aplikasi website perusahaan

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 2
Tanggal : 20 - 25 Maret 2023

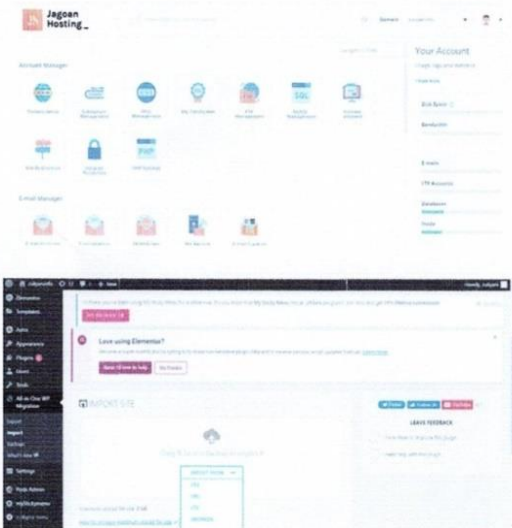
NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	1. Mempelajari cms wordpress 2. Membangun API sistem penjualan dengan NestJs 3. membuat website dengan elementor wordpress	Bapak Iqbal dan Bapak Sucipto	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		Membangun API sistem penjualan dengan NestJs
2		membuat website dengan elementor wordpress

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 3
Tanggal : 27 Maret - 01 April 2023

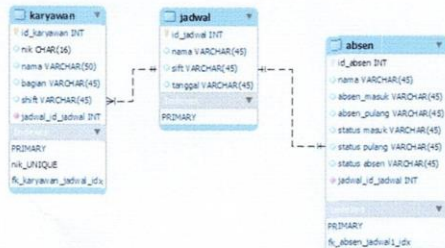
NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	<ol style="list-style-type: none"> 1. Upload wordpress lokal ke hosting 2. Diskusi tentang project magang 3. Mendesain tampilan layanan dan portfolio di wordpress dengan elementor 	Bapak Iqbal	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		Upload wordpress lokal ke hosting

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 5
Tanggal : 10 - 15 April 2023

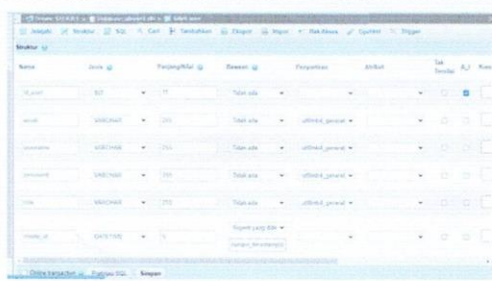
NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	1. Merancang database absensi karyawan	Bapak Iqbal	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		Merancang database absensi karyawan

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 6
Tanggal : 17 - 21 April 2023

NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	1. Membuat tabel user di phpmyadmin 2. Membuat tabel karyawan 3. Membuat tabel absensi 4. Membuat tabel shift	Bapak Iqbal	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		Membuat tabel user

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 7
Tanggal : 1 - 6 Mei 2023



NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	1. Membuat Laporan Perencanaan Aplikasi Absensi 2. Update database absensi	Bapak Iqbal	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 9
Tanggal : 22 - 27 Mei 2023

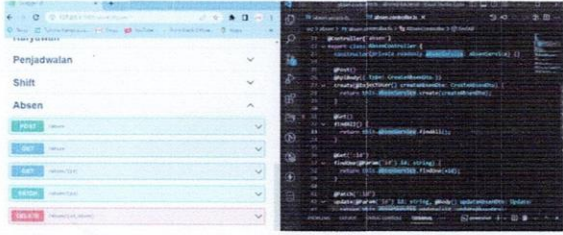
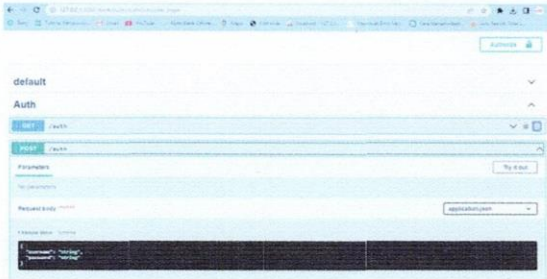
NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	1. Membuat crud data user dengan swagger ui nestjs 2. Membuat crud data karyawan dengan swagger ui nestjs	Bapak Iqbal	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		Membuat crud data user dengan swagger ui nestjs
		Membuat crud data karyawan dengan swagger ui nestjs

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 10
Tanggal : 29 Mei - 03 Juni 2023

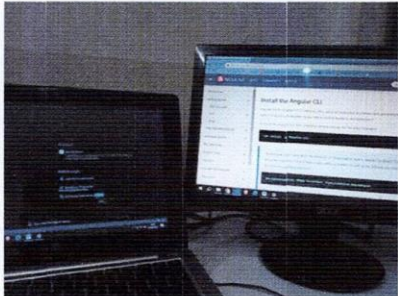
NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	1. Membuat back-end crud data absen 2. Membuat back-end untuk autentifikasi	Bapak Iqbal	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		Membuat back-end crud data absen
2		Membuat back-end untuk autentifikasi

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 11
Tanggal : 05 - 10 Juni 2023

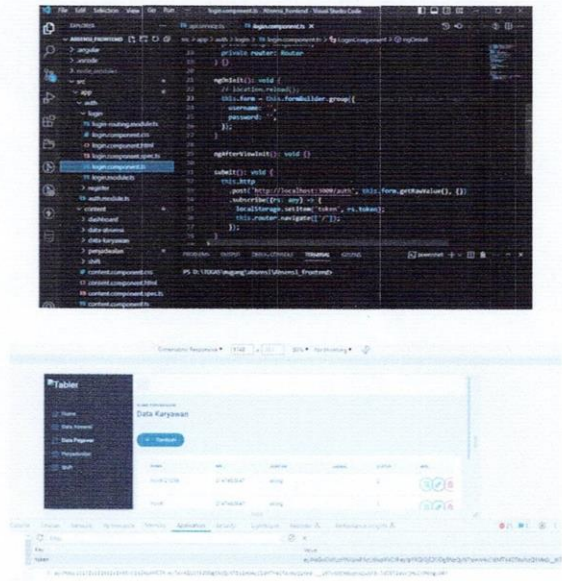
NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	1. Menambahkan field data kontrak 2. Mempelajari sruktur dan cara kerja framework angular 3. Mempelajari setup api backend ke angular	Bapak Iqbal	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		Mempelajari sruktur dan cara kerja framework angular

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 12
Tanggal : 12 - 17 Juni 2023


NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	<ol style="list-style-type: none"> 1. Menghubungkan login api backend ke angular dengan mengambil token 2. Update tampilan user yang sudah login dan belum login dan belum 3. Create fitur logout di angular 	Bapak Sucipto	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		Menghubungkan login api backend ke angular dengan mengambil token

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 13
Tanggal : 19 - 24 Juni 2023

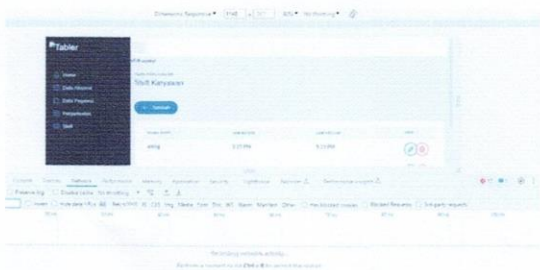
NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	1. Menampilkan data api absen di angular 2. Update penjadwalan dto (backend crud data penjadwalan) 3. Update entity penjadwalan (tabel penjadwalan)	Bapak Sucipto	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		Menampilkan data api absen di angular

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 14
Tanggal : 26 Juni – 01 Juli 2023

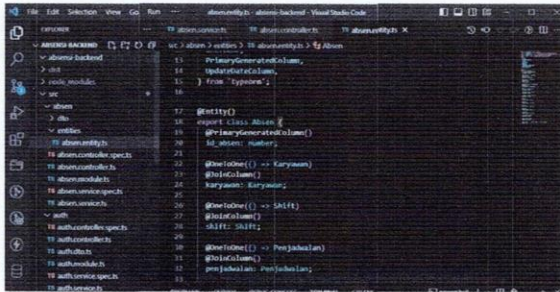
NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	1. Menampilkan data api shift 2. Update shift service 3. Update shift dto	Bapak Sucipto	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		Menampilkan data api shift

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**


Hari/Minggu : Senin-Sabtu / Minggu 15
Tanggal : 03 Juli – 08 Juli 2023

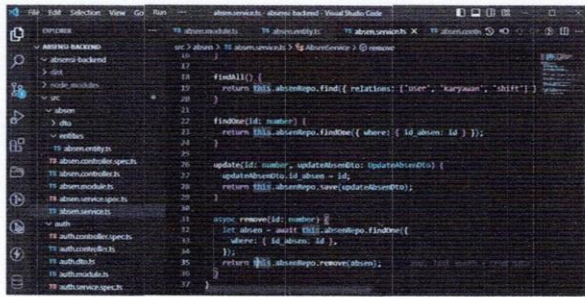
NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	1. update entity absen (tabel Absensi) 2. update createAbsen dto 3. update absensi service	Bapak Sucipto	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		update entity absen (tabel Absensi)

**KEGIATAN HARIAN
KERJA PRAKTEK (KP)**

Hari/Minggu : Senin-Sabtu / Minggu 16
Tanggal : 10 Juli – 15 Juli 2023

NO	URAIAN KEGIATAN	PEMBERI TUGAS	PARAF
	1. menambahkan relasi tabel di service absen 2. membuat laporan magang	Bapak Iqbal	
	Catatan Pembimbing Industri		

NO	Gambaran Kerja	KETERANGAN
1		menambahkan relasi tabel di service absen