

## BAB III

### PELAKSANAAN KERJA PROFESI

#### 3.1 Bidang Kerja

Mahasiswa menjalani Kerja Praktek di PT. Manunggal Rizki Karyatama Telnics (MARKTEL), ditempatkan di kantor pusat di Bandung sebagai *Database Developer*. Tanggung jawab utama mahasiswa mencakup perancangan, pengelolaan, dan pemeliharaan database untuk sistem absensi karyawan berbasis CCTV menggunakan PostgreSQL. *Database* ini terintegrasi dengan perangkat CCTV untuk mencatat kehadiran karyawan secara otomatis, memastikan data terekam dengan aman, akurat, dan dapat diakses secara *real-time*.

Sebagai *Database Developer*, peran mahasiswa meliputi pembuatan skema *database*, perancangan tabel data, serta pengelolaan proses CRUD (*Create, Read, Update, Delete*) untuk data karyawan. Selain itu, mahasiswa juga bertugas mengoptimalkan performa database agar sistem berfungsi secara efisien dan andal. Proyek ini memberikan pengalaman praktis dalam pengembangan *database*, integrasi data *real-time*, serta menjaga keamanan dan integritas data dalam sistem absensi berbasis CCTV.

Mahasiswa juga membentuk tim bersama beberapa rekan untuk mendukung divisi pengembangan sistem tersebut. Proses perancangan *database* melibatkan tahap konseptual, logikal, dan fisikal untuk merencanakan struktur data. Keterlibatan dalam tahap ini sangat penting, mengingat dampak signifikan terhadap data yang digunakan saat perangkat lunak diimplementasikan. Tahap konseptual berfokus pada pembuatan model dasar tanpa mempertimbangkan aspek teknis, sedangkan tahap logikal lebih menekankan pada pengembangan struktur data yang lebih rinci (Michelle Larassati Ayusmara Latukolan, April, 2019). *Database* memiliki peran penting dalam dunia teknologi informasi dengan menyimpan, mengorganisir, dan mengelola data, serta memastikan akses yang cepat dan efektif melalui berbagai aplikasi dan sistem.

Mahasiswa menggunakan PostgreSQL sebagai sistem manajemen basis data untuk merancang *database* yang khusus dibuat untuk mengelola data

absensi karyawan yang diambil melalui sistem CCTV. PostgreSQL, sebagai salah satu server *database* yang terkenal, menggunakan SQL sebagai bahasa untuk mengakses data dalam *database*-nya. PostgreSQL dikenal dengan kemampuannya dalam menangani data yang kompleks dan memiliki lisensi *open source*. *Database* ini bertujuan untuk mendukung sistem absensi karyawan di PT. MARKTEL. Untuk mempermudah administrasi dan interaksi dengan *database*, mahasiswa memanfaatkan platform pgAdmin. pgAdmin adalah salah satu perangkat lunak penting dalam pengelolaan *database* menggunakan PostgreSQL (Hartiwati, 2022). Penggunaan pgAdmin sebagai antarmuka administratif merupakan langkah strategis untuk memudahkan manajemen dan manipulasi data absensi, memastikan bahwa informasi yang diperoleh dari sistem CCTV dapat diakses dan digunakan secara efisien dalam upaya pemantauan kehadiran karyawan di PT. MARKTEL.

Mahasiswa memiliki kemampuan untuk menerapkan pengetahuan dan pemahaman dalam pemrograman yang telah dipelajari, serta dapat berkomunikasi langsung dengan mentor di PT. MARKTEL. Mereka juga mampu bekerja sama dengan profesional yang ahli di bidangnya, memperoleh pengetahuan baru, dan beradaptasi dengan budaya kerja di PT. MARKTEL. Proyek yang mereka kerjakan akan menjadi dasar penilaian pada akhir program KP, yang menjadi tanggung jawab mahasiswa yang menjalani program tersebut.

### **3.2 Pelaksanaan Kerja**

Dalam mengembangkan sistem sesuai target yang telah ditetapkan, praktikan dan pembimbing eksternal bekerja sama dalam diskusi untuk menyusun langkah-langkah kerja yang sistematis dan terstruktur. Untuk memastikan proses pengembangan berjalan lancar, praktikan membuat linimasa yang jelas sebagai panduan. Langkah-langkah ini dirancang dengan teliti agar setiap tahap pekerjaan dilakukan secara urut dan sesuai jadwal yang telah ditentukan.

Tabel 3.1 Linimasa Pelaksanaan Kerja Profesi

Aktivitas	Juli - Agustus								
	M1	M2	M3	M4	M5	M6	M7	M8	M9
Orientasi dan pengenalan perusahaan.	■								
Perancangan skema <i>database</i> .		■							
Instalasi dan konfigurasi PostgreSQL.			■						
Integrasi <i>database</i> dengan sistem CCTV.				■					
Pengujian sistem dan perbaikan bug.					■				
Optimasi performa <i>database</i>						■			
Dokumentasi teknis							■		
Evaluasi sistem dan rivew dengan pembimbing eksternal.								■	
Presentasi hasil kerja.									■

Tabel 3.2 merupakan tahapan keseluruhan pelaksanaan Kerja Profesi (KP) dilakukan sesuai linimasa sembilan minggu yang terstruktur. Minggu pertama fokus pada orientasi perusahaan, diikuti perancangan skema *database* pada minggu kedua. Instalasi dan konfigurasi PostgreSQL dilakukan pada minggu ketiga, sementara integrasi dengan sistem CCTV dilaksanakan pada minggu keempat. Pengujian dan perbaikan bug dijalankan pada minggu kelima, dan optimasi performa *database* pada minggu keenam. Dokumentasi teknis disusun pada minggu ketujuh, dilanjutkan evaluasi dan *review* di minggu kedelapan. Terakhir, minggu kesembilan adalah untuk presentasi hasil kerja.

### 3.2.1 Perancangan

*Database* sistem ini dibangun menggunakan PostgreSQL, yang menyediakan fitur yang kuat untuk manajemen data relasional. Sistem presensi karyawan berbasis CCTV di PT. Manunggaling Rizki Karyatama Telnics dirancang untuk memantau dan mencatat kehadiran karyawan secara otomatis. Dengan menggunakan teknologi pengenalan wajah, sistem ini dapat mengidentifikasi karyawan yang hadir, mencatat waktu masuk mereka, dan menyimpan informasi tersebut di dalam *database*. PostgreSQL dikenal sebagai sistem manajemen basis data yang tangguh, mampu menangani data kompleks,

dan didukung oleh komunitas open source yang besar (Hartiwati, 2022).

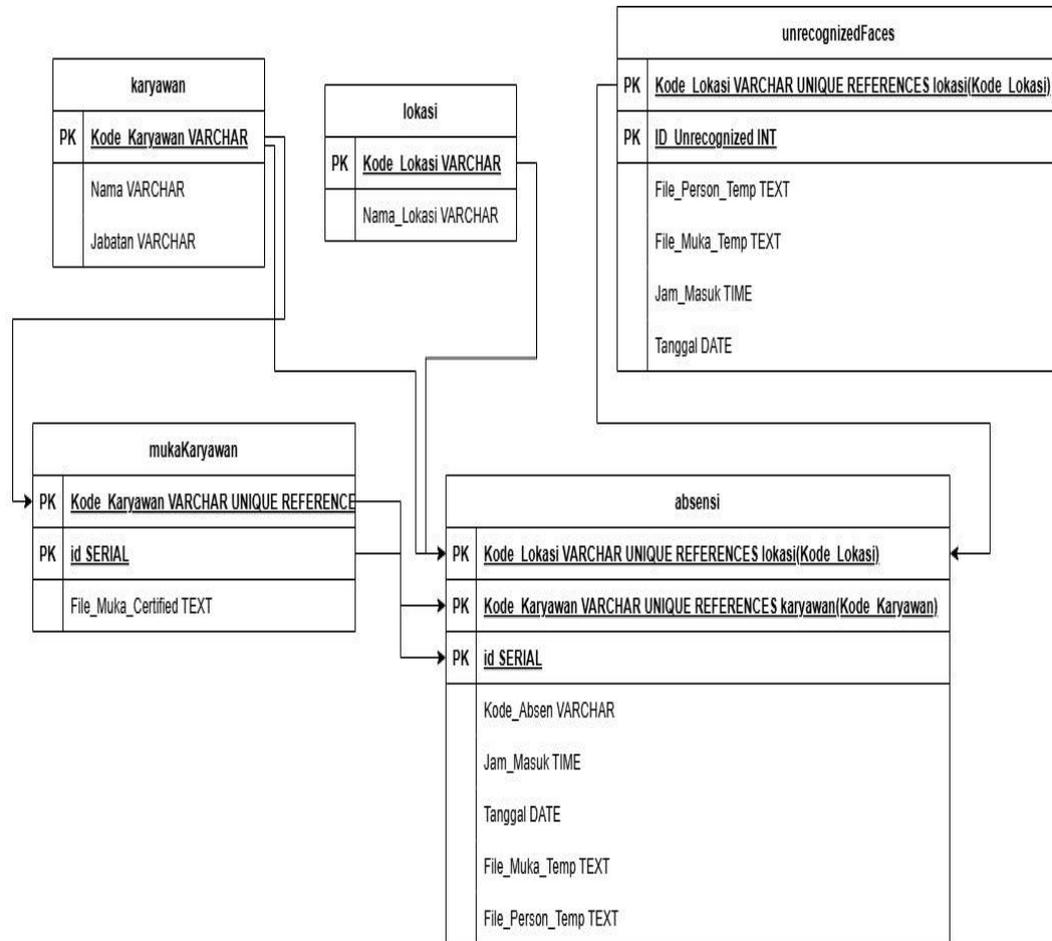
Dalam proses perancangannya, digunakan Entity-Relationship Diagram (ERD) untuk memodelkan struktur data. ERD membantu menggambarkan hubungan antara entitas dalam database, seperti tabel Karyawan, Absensi, dan tabel lainnya. Penggunaan ERD ini memastikan desain database yang sistematis dan sesuai kebutuhan perusahaan (Rahmawati, 2021).

Proses perancangan database untuk mengelola absensi di PT. MARKTEL memerlukan urutan yang terbagi dalam beberapa tahapan. Hal ini dilakukan untuk memastikan rancangan database yang disusun sesuai dengan kebutuhan yang diperlukan perusahaan, serta mampu menangani data secara efisien dan terstruktur (Putra, 2020).

Proses perancangan *database* untuk mengelola absensi di PT. MARKTEL memerlukan urutan yang terbagi dalam beberapa tahapan. Hal ini dilakukan untuk memastikan rancangan *database* yang disusun sesuai dengan kebutuhan yang diperlukan perusahaan, diantaranya:



**a. Entity-Relationship Diagram (ERD)**



Gambar 3.2 Entity-Relationship Diagram (ERD)

Gambar 3.1 menggambarkan struktur dari Entity-Relationship Diagram (ERD). ERD adalah salah satu alat yang sangat penting dalam dunia perancangan sistem database. Diagram ini digunakan untuk memodelkan hubungan antar entitas yang ada dalam suatu sistem database, serta menggambarkan elemen-elemen yang saling terhubung di dalamnya. Dengan menggunakan ERD, pengembang sistem dapat lebih mudah merancang, memahami, dan mengelola database, khususnya dalam konteks penyimpanan data dalam volume besar. ERD menjadi pilihan utama bagi berbagai organisasi di seluruh dunia karena menyediakan metode yang handal untuk penyimpanan dan pengambilan data secara efisien. Selain itu, ERD juga menggabungkan performa sistem yang baik dengan kemudahan implementasi, menjadikannya sebagai alat yang banyak

digunakan dalam merancang sistem database yang kompleks dan berskala besar.

Menurut Sutarman (2017), Entity-Relationship Diagram (ERD) adalah salah satu metode yang digunakan untuk menggambarkan struktur data dan hubungan antar entitas dalam sistem database. ERD memberikan visualisasi yang jelas dan mudah dipahami mengenai bagaimana data dalam sistem berinteraksi dan saling terhubung. Dengan demikian, ERD sangat memudahkan dalam perancangan dan implementasi database yang efektif dan efisien. ERD bukan hanya digunakan untuk menggambarkan struktur statis dari database, tetapi juga dapat digunakan untuk memodelkan dinamika yang terjadi pada hubungan antar entitas tersebut. Oleh karena itu, ERD menjadi alat yang penting dalam mendesain sistem yang memiliki skala besar dan membutuhkan pengelolaan data dalam jumlah yang besar pula.

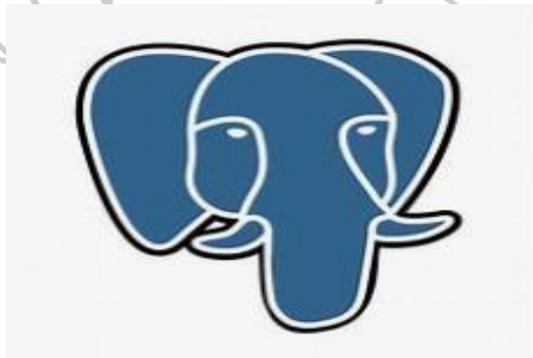
Dalam konteks organisasi, ERD digunakan untuk menyusun basis data yang dapat menangani berbagai macam kebutuhan, mulai dari pengolahan transaksi hingga analisis data yang lebih kompleks. Salah satu kelebihan utama dari penggunaan ERD adalah kemampuannya untuk menyediakan gambaran yang komprehensif tentang struktur data tanpa perlu mempelajari seluruh kode atau detail implementasi database. Dengan memanfaatkan ERD, proses pembuatan dan pengelolaan database dapat dilakukan dengan lebih cepat dan lebih akurat. Hal ini sangat penting bagi organisasi yang mengelola data dalam jumlah besar dan membutuhkan performa sistem yang optimal. Oleh karena itu, ERD sering menjadi pilihan utama bagi berbagai organisasi di dunia yang ingin merancang database yang tidak hanya besar tetapi juga efektif dan efisien dalam mengelola data.

Selain itu, ERD juga memberikan keuntungan dalam hal pemeliharaan dan pengembangan sistem di masa depan. Dalam dunia teknologi yang terus berkembang pesat, kebutuhan akan sistem yang dapat beradaptasi dengan cepat terhadap perubahan sangat penting. ERD memungkinkan pengembang untuk melihat secara langsung hubungan antar entitas dan mempermudah penyesuaian struktur database tanpa mengganggu performa sistem secara keseluruhan. Dengan demikian, ERD tidak hanya membantu dalam desain awal, tetapi juga dalam proses pemeliharaan dan pengembangan sistem di masa

mendatang.

Secara keseluruhan, ERD adalah metode yang sangat berguna dalam merancang dan mengelola database, khususnya dalam konteks organisasi besar yang membutuhkan penyimpanan dan pengolahan data dalam jumlah besar. Dengan berbagai manfaat yang ditawarkan, seperti kemudahan implementasi, efisiensi pengelolaan data, dan kemudahan pemeliharaan, ERD tetap menjadi pilihan utama bagi pengembang sistem database di seluruh dunia.

### 3.2.1 PostgreSQL



Gambar 3.3 Logo *postgreSQL*

*PostgreSQL* adalah sistem manajemen basis data yang menggunakan model relasional untuk menyimpan dan mengelola data. Sistem ini dirancang untuk memberikan kontrol transaksi yang aman, menjaga integritas data, serta mendukung ekstensibilitas yang tinggi bagi penggunaannya.

Salah satu keunggulan utama *postgreSQL* adalah kemampuannya menangani beban kerja yang kompleks tanpa mengorbankan kinerja atau akurasi. Sebagai solusi database open-source, *PostgreSQL* tersedia secara gratis dan banyak digunakan dalam berbagai aplikasi.

Pengembangan *postgreSQL* dimulai pada tahun 1986 di Universitas California, Berkeley, untuk memenuhi kebutuhan akan sistem basis data yang lebih andal. Tujuannya adalah menciptakan sistem yang mampu mengolah *query* kompleks dan menangani data dalam jumlah besar. *PostgreSQL* menyimpan data dalam tabel menggunakan model relasional, dengan setiap operasi

pengguna seperti *insert*, *update*, atau *delete* dijalankan melalui *query SQL*.

PostgreSQL juga menerapkan mekanisme transaksi multi-versi (*Multi-Version Concurrency Control/MVCC*) untuk menjaga konsistensi data tanpa memerlukan penguncian dalam skala besar. Teknologi ini memungkinkan PostgreSQL menawarkan kinerja tinggi dan kemampuan pengolahan data yang optimal.

## **b. Perancangan Database**

### Tabel dan Struktur

#### 1. Tabel Karyawan

- Kode\_Karyawan (*Primary Key, VarChar, Unique*): ID unik untuk setiap karyawan.
- Nama (*VarChar*): Nama lengkap karyawan.
- Jabatan (*VarChar*): Posisi atau jabatan karyawan.

#### 2. Tabel MukaKaryawan

- id (*Primary Key, SERIAL*): ID unik untuk setiap entri wajah.
- Kode\_Karyawan (*Foreign Key, VarChar, Unique*): Merujuk ke tabel Karyawan.
- File\_Muka\_Certified (*TEXT*): Menyimpan file wajah terverifikasi.

#### 3. Tabel Lokasi

- Kode\_Lokasi (*Primary Key, VarChar, Unique*): ID unik untuk setiap lokasi.
- Nama\_Lokasi (*VarChar*): Nama lokasi tempat karyawan melakukan absensi.

#### 4. Tabel Absensi

- id (*Primary Key, SERIAL*): ID unik untuk setiap entri absensi.
- Kode\_Absen (*VarChar*): Kode unik untuk absensi.
- Kode\_Karyawan (*Foreign Key, VarChar*): Merujuk ke tabel Karyawan.
- Jam\_Masuk (*TIME*): Waktu masuk karyawan.
- Tanggal (*Date*): Tanggal absensi.
- Kode\_Lokasi (*Foreign Key, VarChar*): Merujuk ke tabel Lokasi.
- File\_Muka\_Temp (*TEXT*): Menyimpan file wajah sementara.

- File\_Person\_Temp (*TEXT*): Menyimpan foto sementara karyawan.

5. Tabel UnrecognizedFaces

- ID\_Unrecognized (*Primary Key, INT*): ID unik untuk wajah yang tidak dikenali.
- File\_Person\_Temp (*TEXT*): Menyimpan foto yang tidak dikenali.
- File\_Muka\_Temp (*TEXT*): Menyimpan file wajah yang tidak dikenali.
- Jam\_Masuk (*TIME*): Waktu ketika wajah tidak dikenali.
- Tanggal (*Date*): Tanggal kejadian.
- Kode\_Lokasi (*Foreign Key, VarChar*): Merujuk ke tabel Lokasi.

c. Normalisasi *Database*

a. Normalisasi 1NF

absensi	
PK	<u>id SERIAL</u>
	Kode_Karyawan VARCHAR
	Nama VARCHAR
	Jabatan VARCHAR
	File_Muka_Certified TEXT
	Kode_Lokasi VARCHAR
	Nama_Lokasi VARCHAR
	Kode_Absen VARCHAR
	Jam_Masuk TIME
	Tanggal DATE
	File_Muka_Temp TEXT
	File_Person_Temp TEXT

Gambar 3.4 Normalisasi 1NF

Berdasarkan Gambar 3.3, berikut adalah penjelasan normalisasi *First Normal*

Form (1NF):

Syarat Normalisasi 1NF:

1. *Atomicity*: Semua nilai dalam atribut harus bersifat atomik, artinya setiap kolom hanya boleh berisi satu nilai, tidak berupa set, list, atau array.
2. Unik: Setiap baris dalam tabel harus unik dan memiliki *primary key*.
3. Tidak Ada Kolom Berulang: Tidak boleh ada kolom yang menyimpan data berulang dalam satu tabel.

Analisis Tabel "absensi" dalam Gambar:

1. *Primary Key* (PK):
  - Tabel memiliki kolom id SERIAL sebagai *primary key*, yang memastikan setiap baris unik.
2. Kolom dengan Data Atomik:
  - Setiap kolom, seperti Kode\_Karyawan, Nama, dan Jabatan, menyimpan data atomik (hanya satu nilai per sel).
3. Tidak Ada Kolom Berulang:
  - Semua data terkait absensi, seperti Jam\_Masuk, Tanggal, dan atribut lainnya, berada dalam kolom yang terpisah, sehingga tidak ada kolom berulang untuk data yang sama.

Kesimpulan:

Tabel "absensi" sudah memenuhi 1NF, karena:

- Setiap kolom menyimpan data atomik.
- Setiap baris memiliki *primary key* untuk memastikan unik.
- Tidak ada kolom yang menyimpan set data atau data berulang.

## b. Normalisasi 2NF

karyawan		lokasi		absensi	
PK	<u>Kode_Karyawan</u> VARCHAR	PK	<u>Kode_Lokasi</u> VARCHAR	PK	<u>Kode_Lokasi</u> VARCHAR
	Nama VARCHAR		Nama_Lokasi VARCHAR	PK	<u>Kode_Karyawan</u> VARCHAR
	Jabatan VARCHAR			PK	<u>id</u> SERIAL
	File_Muka_Certified TEXT				Kode_Absen VARCHAR
					Jam_Masuk TIME
					Tanggal DATE
					File_Muka_Temp TEXT
					File_Person_Temp TEXT
					CONSTRAINT fk_karyawan_absensi FOREIGN KEY
					CONSTRAINT fk_lokasi_absensi FOREIGN KEY (Ko

Gambar 3.5 Normalisasi 2NF

Berdasarkan gambar 3.4, tabel telah melalui proses normalisasi *Second Normal Form* (2NF). Berikut adalah penjelasan detail:

Syarat Normalisasi 2NF:

1. Sudah Memenuhi 1NF:
  - Tabel harus sudah memenuhi aturan 1NF, yaitu semua atribut bersifat atomik, tidak ada kolom berulang, dan memiliki *primary key*.
2. Tidak Ada Ketergantungan Parsial:
  - Semua atribut *non-prime* (atribut yang bukan bagian dari *primary key*) harus bergantung sepenuhnya pada *primary key*, bukan pada bagian dari *primary key* (ketergantungan parsial).

Analisis Gambar:

Pada gambar, tabel telah dibagi menjadi tiga tabel: karyawan, lokasi, dan absensi.

1. Tabel Karyawan:

- Memiliki Kode\_Karyawan sebagai *primary key*.
  - Atribut lainnya, seperti Nama, Jabatan, dan File\_Muka\_Certified, sepenuhnya bergantung pada Kode\_Karyawan.
2. Tabel Lokasi:
- Memiliki Kode\_Lokasi sebagai *primary key*.
  - Atribut Nama\_Lokasi sepenuhnya bergantung pada Kode\_Lokasi.
3. Tabel Absensi:
- Memiliki id sebagai *primary key* utama (unik untuk setiap baris), dengan Kode\_Karyawan dan Kode\_Lokasi sebagai *foreign key*.
  - Atribut lainnya, seperti Kode\_Absen, Jam\_Masuk, Tanggal, File\_Muka\_Temp, dan File\_Person\_Temp, semuanya bergantung sepenuhnya pada kombinasi kunci id.

#### Penyelesaian Ketergantungan Parsial:

Pada tabel absensi di versi sebelumnya (gambar 1NF), beberapa atribut seperti Nama, Jabatan, dan Nama\_Lokasi berada di tabel yang sama meskipun tidak sepenuhnya bergantung pada *primary key* tabel absensi. Hal ini menyebabkan ketergantungan parsial. Pada gambar 2NF ini, ketergantungan parsial telah dihilangkan dengan memindahkan atribut tersebut ke tabel karyawan dan lokasi, sesuai dengan entitas yang seharusnya. Dengan demikian:

- Data karyawan dipisah ke tabel karyawan.
- Data lokasi dipisah ke tabel lokasi.
- Tabel absensi hanya menyimpan referensi ke tabel karyawan dan lokasi melalui *foreign key*.

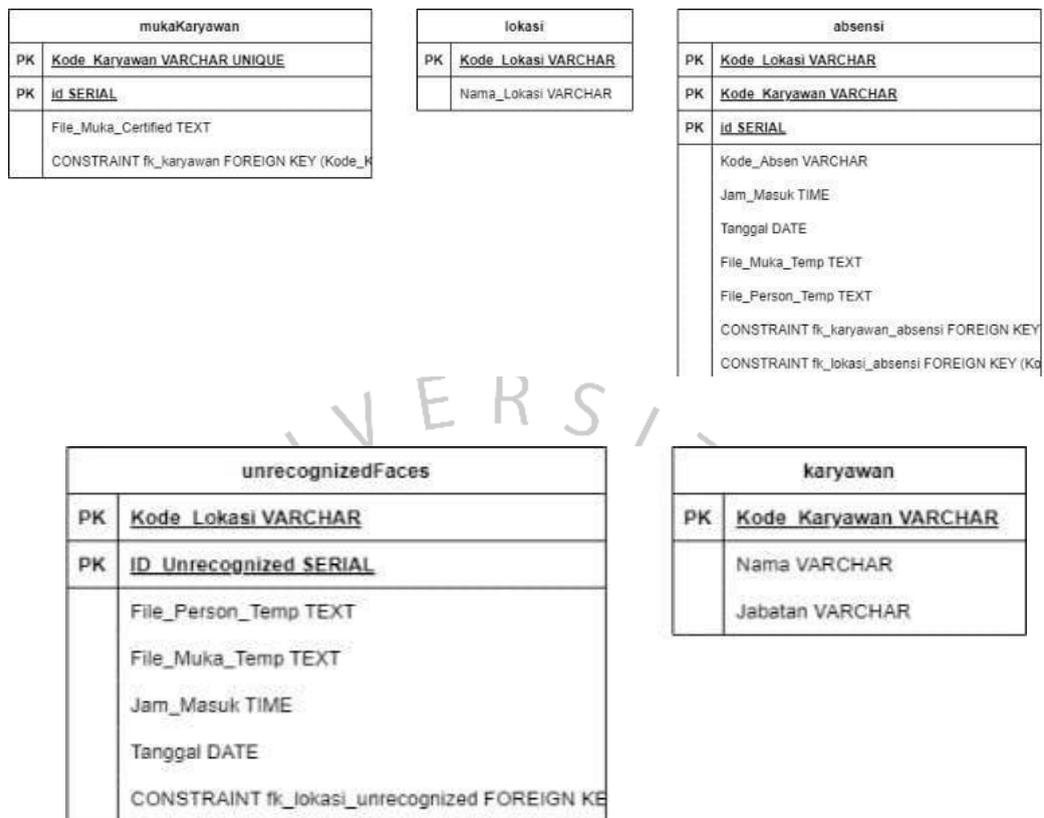
#### Kesimpulan:

Tabel di gambar telah memenuhi 2NF karena:

1. Semua tabel sudah memenuhi aturan 1NF.
2. Tidak ada atribut yang bergantung secara parsial pada *primary key*. Setiap atribut *non-prime* bergantung sepenuhnya pada *primary key* tabelnya masing-masing.

Langkah ini mengurangi redundansi data dan meningkatkan integritas data.

### c. Normalisasi 3NF



Gambar 3.6 Normalisasi 3NF

Berdasarkan gambar 3.5, tabel telah melalui proses normalisasi hingga *Third Normal Form* (3NF). Berikut adalah penjelasan detailnya:

Syarat Normalisasi 3NF:

1. Sudah Memenuhi 2NF:
  - Semua atribut *non-prime* bergantung secara penuh pada *primary key*.
2. Menghilangkan Ketergantungan Transitif:
  - Atribut *non-prime* tidak boleh memiliki ketergantungan transitif pada *primary key*. Artinya, atribut *non-prime* hanya boleh

bergantung langsung pada *primary key*, bukan pada atribut *non-prime* lainnya.

Analisis Gambar:

Tabel telah dipisah menjadi lima tabel untuk menghilangkan redundansi data dan ketergantungan transitif:

1. Tabel karyawan:
  - Memiliki Kode\_Karyawan sebagai *primary key*.
  - Atribut Nama dan Jabatan bergantung langsung pada Kode\_Karyawan.
  - Tidak ada ketergantungan transitif karena semua atribut hanya bergantung pada *primary key*.
2. Tabel mukaKaryawan:
  - Memiliki id sebagai *primary key* dan Kode\_Karyawan sebagai *foreign key*.
  - Atribut File\_Muka\_Certified bergantung langsung pada Kode\_Karyawan melalui id.
  - Data wajah karyawan dipisah agar lebih modular dan mencegah redundansi dalam data wajah.
3. Tabel lokasi:
  - Memiliki Kode\_Lokasi sebagai *primary key*.
  - Atribut Nama\_Lokasi bergantung langsung pada Kode\_Lokasi.
  - Tidak ada ketergantungan transitif.
4. Tabel absensi:
  - Memiliki id sebagai *primary key* dan Kode\_Karyawan serta Kode\_Lokasi sebagai *foreign key*.
  - Atribut seperti Kode\_Absen, Jam\_Masuk, Tanggal, File\_Muka\_Temp, dan File\_Person\_Temp bergantung langsung pada id.
  - Tabel ini mengelola data absensi karyawan dengan referensi ke lokasi dan data karyawan.
5. Tabel unrecognizedFaces:
  - Memiliki ID\_Unrecognized sebagai *primary key* dan Kode\_Lokasi sebagai *foreign key*.

- Atribut seperti File\_Person\_Temp, File\_Muka\_Temp, Jam\_Masuk, dan Tanggal bergantung langsung pada ID\_Unrecognized.
- Tabel ini dirancang untuk menyimpan data wajah yang tidak dikenali oleh sistem, lengkap dengan informasi lokasi dan waktu kejadian.

Penyelesaian Ketergantungan Transitif:

Pada tabel di 2NF, beberapa atribut mungkin memiliki ketergantungan transitif. Contohnya, informasi wajah (File\_Muka\_Certified) yang bergantung pada Kode\_Karyawan dipisah ke tabel mukaKaryawan. Dengan cara ini:

- Data terkait wajah karyawan dipisah agar modular.
- Tidak ada atribut *non-prime* yang bergantung transitif pada *primary key*.

Kesimpulan:

Tabel dalam gambar telah memenuhi 3NF karena:

1. Sudah memenuhi aturan 2NF.
2. Tidak ada ketergantungan transitif antara atribut *non-prime*.
3. Data diorganisasikan dengan lebih efisien, menghilangkan redundansi dan meningkatkan integritas data.

Dengan struktur ini, *database* menjadi lebih fleksibel, mudah diperbarui, dan meminimalkan kemungkinan inkonsistensi data.

#### **d. Query**

Gambar di bawah ini menggambarkan *query* SQL yang digunakan untuk membangun struktur data dalam sistem absensi berbasis wajah. *Query* ini mencakup pembuatan dua tabel utama, yaitu tabel 'absensi' dan tabel 'unrecognizedFaces'. Tabel 'absensi' dirancang untuk mencatat data kehadiran karyawan, termasuk waktu masuk, lokasi, dan informasi wajah yang digunakan untuk verifikasi identitas. Sementara itu, tabel 'unrecognizedFaces' berfungsi untuk menyimpan data wajah yang tidak dikenali, lengkap dengan waktu dan lokasi kejadian. Melalui *query* ini, kita dapat melihat bagaimana relasi antar tabel dibangun dan bagaimana data akan disimpan secara terstruktur, mendukung integritas dan efisiensi dalam pengelolaan sistem absensi.

```

-- Membuat tabel karyawan
CREATE TABLE karyawan (
    Kode_Karyawan VARCHAR PRIMARY KEY,
    Nama VARCHAR,
    Jabatan VARCHAR
);

-- Membuat tabel mukaKaryawan
CREATE TABLE mukaKaryawan (
    id SERIAL PRIMARY KEY,
    Kode_Karyawan VARCHAR UNIQUE,
    File_Muka_Certified TEXT,
    CONSTRAINT fk_karyawan FOREIGN KEY (Kode_Karyawan) REFERENCES karyawan(Kode_Karyawan)
);

-- Membuat tabel lokasi
CREATE TABLE lokasi (
    Kode_Lokasi VARCHAR PRIMARY KEY,
    Nama_Lokasi VARCHAR
);

-- Membuat tabel absensi
CREATE TABLE absensi (
    id SERIAL PRIMARY KEY,
    Kode_Absen VARCHAR,
    Kode_Karyawan VARCHAR UNIQUE,
    Jam_Masuk TIME,
    Kode_Lokasi VARCHAR,
    Tanggal DATE,
    File_Muka_Temp TEXT,
    File_Person_Temp TEXT,
    CONSTRAINT fk_karyawan_absensi FOREIGN KEY (Kode_Karyawan) REFERENCES karyawan(Kode_Karyawan),
    CONSTRAINT fk_lokasi_absensi FOREIGN KEY (Kode_Lokasi) REFERENCES lokasi(Kode_Lokasi)
);

-- Membuat tabel unrecognizedFaces
CREATE TABLE unrecognizedFaces (
    ID_Unrecognized SERIAL PRIMARY KEY,
    File_Person_Temp TEXT,
    File_Muka_Temp TEXT,
    Jam_Masuk TIME,
    Tanggal DATE,
    Kode_Lokasi VARCHAR UNIQUE,
    CONSTRAINT fk_lokasi_unrecognized FOREIGN KEY (Kode_Lokasi) REFERENCES lokasi(Kode_Lokasi)
);

```

Gambar 3.7 Query Tools

Gambar 3.6 Query Tools menjelaskan kode program dalam pembuatan *database* untuk sistem absensi pada PT. Marktrel.

### 1. Tabel karyawan:

<input type="checkbox"/>	Name	Data type	Not NULL?	Primary key?
<input type="checkbox"/>	kode_karyawan	character varying	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	nama	character varying	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	jabatan	character varying	<input type="checkbox"/>	<input type="checkbox"/>

Gambar 3.8 Tabel karyawan

Pada gambar 3.7, tabel ini menyimpan data karyawan dengan kolom Kode\_Karyawan sebagai *primary key*. Kolom lainnya mencakup Nama dan Jabatan yang digunakan untuk menyimpan informasi dasar karyawan.

### 2. Tabel muka karyawan:

<input type="checkbox"/>	Name	Data type	Not NULL?	Primary key?
<input type="checkbox"/>	id	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	kode_karyawan	character varying	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	file_muka_certified	text	<input type="checkbox"/>	<input type="checkbox"/>

Gambar 3.9 Tabel muka karyawan

Pada gambar 3.8, tabel ini berfungsi menyimpan data wajah karyawan yang sudah terverifikasi. Tabel ini memiliki kolom *primary key* id, dan kolom Kode\_Karyawan sebagai *foreign key* yang merujuk ke tabel karyawan. Kolom File\_Muka\_Certified menyimpan file wajah terverifikasi dalam bentuk teks.

### 3. Tabel lokasi:

<input type="checkbox"/>	Name	Data type	Not NULL?	Primary key?	Comment
<input type="checkbox"/>	kode_lokasi	character varying	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	nama_lokasi	character varying	<input type="checkbox"/>	<input type="checkbox"/>	

Gambar 3.10 Tabel lokasi

Pada gambar 3.9, tabel ini digunakan untuk mencatat informasi lokasi dengan Kode\_Lokasi sebagai *primary key*. Kolom Nama\_Lokasi menyimpan nama lokasi tertentu, seperti cabang atau gedung tempat karyawan melakukan absensi.

#### 4. Tabel unrecognizedFaces:

<input type="checkbox"/>	Name	Data type	Not NULL?	Primary key?
<input type="checkbox"/>	id	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	kode_absen	character varying	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	kode_karyawan	character varying	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	jam_masuk	time without time zone	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	kode_lokasi	character varying	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	tanggal	date	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	file_muka_temp	text	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	file_person_temp	text	<input type="checkbox"/>	<input type="checkbox"/>

Gambar 3.11 Tabel absensi

Pada gambar 3.10, tabel ini mencatat data absensi karyawan. Kolom utama meliputi id sebagai *primary key*, Kode\_Absen sebagai kode unik absensi, Kode\_Karyawan sebagai *foreign key* yang merujuk ke tabel karyawan, Jam\_Masuk, Tanggal, serta Kode\_Lokasi sebagai *foreign key* yang merujuk ke tabel lokasi. Selain itu, tabel ini menyimpan file sementara dari wajah dan foto karyawan pada kolom File\_Muka\_Temp dan File\_Person\_Temp.

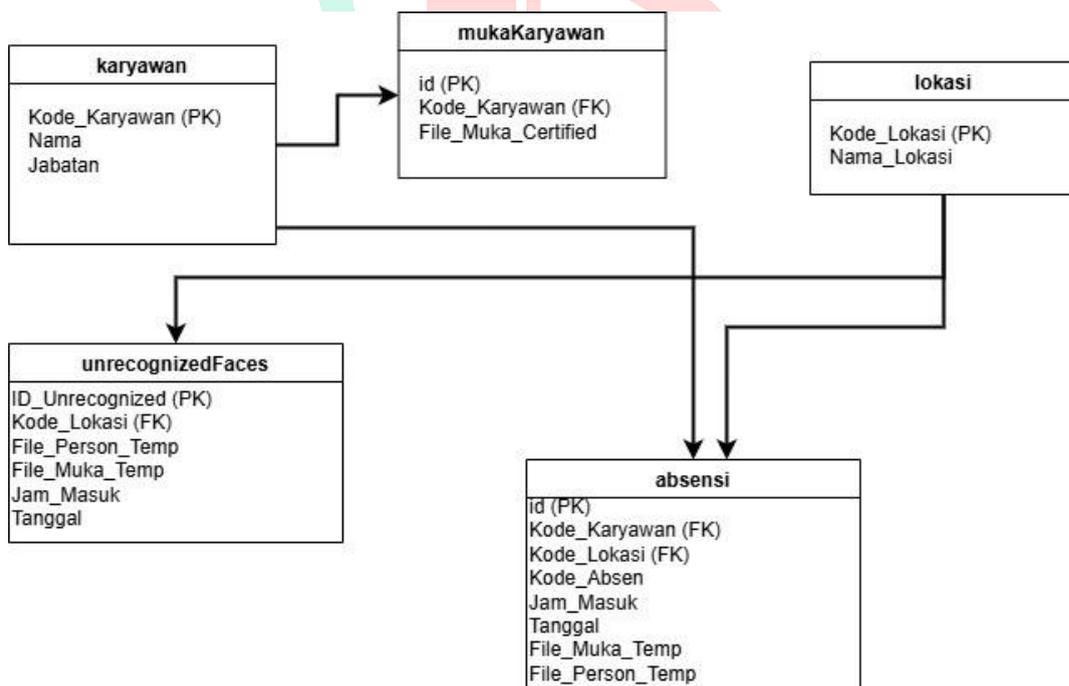
#### 5. Tabel unrecognizedFaces:

<input type="checkbox"/>	Name	Data type	Not NULL?	Primary key?
<input type="checkbox"/>	id_unrecognized	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	file_person_temp	text	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	file_muka_temp	text	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	jam_masuk	time without time zone	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	tanggal	date	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	kode_lokasi	character varying	<input type="checkbox"/>	<input type="checkbox"/>

Gambar 3.12 Tabel unrecognizedfaces

Pada gambar 3.11, tabel ini menyimpan data wajah yang tidak dikenali oleh sistem. Kolom utama meliputi ID\_Unrecognized sebagai *primary key*, Jam\_Masuk, Tanggal, dan Kode\_Lokasi sebagai *foreign key* yang merujuk ke tabel lokasi. Tabel ini juga menyimpan file sementara dari wajah dan foto yang tidak dikenali pada kolom File\_Muka\_Temp dan File\_Person\_Temp.

#### e. Alur proses *database*



Gambar 3.13 Diagram alur *database*

Pada gambar 3.13, Diagram alur *database* ini menggambarkan hubungan antar tabel pada sistem absensi berbasis pengenalan wajah di PT. Manunggaling

Rizky Karyatama Telnics. Berikut penjelasan masing-masing tabel dan hubungannya:

#### 1. Tabel karyawan

- Deskripsi: Tabel ini menyimpan data karyawan seperti:
  - Kode\_Karyawan (PK): Identitas unik setiap karyawan.
  - Nama: Nama karyawan.
  - Jabatan: Posisi pekerjaan karyawan.
- Hubungan:
  - Terhubung dengan tabel mukaKaryawan dan absensi melalui Kode\_Karyawan.

#### 2. Tabel mukaKaryawan

- Deskripsi: Tabel ini menyimpan informasi tambahan mengenai file wajah karyawan yang telah tersertifikasi, digunakan untuk pengenalan wajah:
  - id (PK): Identitas unik untuk setiap data wajah.
  - Kode\_Karyawan (FK): Relasi ke karyawan yang bersangkutan.
  - File\_Muka\_Certified: File wajah karyawan yang telah diverifikasi.
- Hubungan:
  - Relasi FK (*Foreign Key*) ke tabel karyawan.

#### 3. Tabel lokasi

- Deskripsi: Tabel ini menyimpan informasi lokasi tempat absensi:
  - Kode\_Lokasi (PK): Identitas unik lokasi.
  - Nama\_Lokasi: Nama lokasi fisik untuk absensi.
- Hubungan:
  - Terhubung dengan tabel absensi dan unrecognizedFaces.

#### 4. Tabel absensi

- Deskripsi: Tabel ini menyimpan data absensi karyawan:
  - id (PK): Identitas unik untuk setiap entri absensi.
  - Kode\_Karyawan (FK): Relasi ke karyawan yang melakukan absensi.
  - Kode\_Lokasi (FK): Relasi ke lokasi absensi.

- Kode\_Absen: Kode identitas absensi.
- Jam\_Masuk: Waktu absensi.
- Tanggal: Tanggal absensi.
- File\_Muka\_Temp: File wajah sementara dari CCTV saat absensi.
- File\_Person\_Temp: File keseluruhan gambar individu saat absensi.
- Hubungan:
  - Relasi FK ke tabel karyawan melalui Kode\_Karyawan.
  - Relasi FK ke tabel lokasi melalui Kode\_Lokasi.

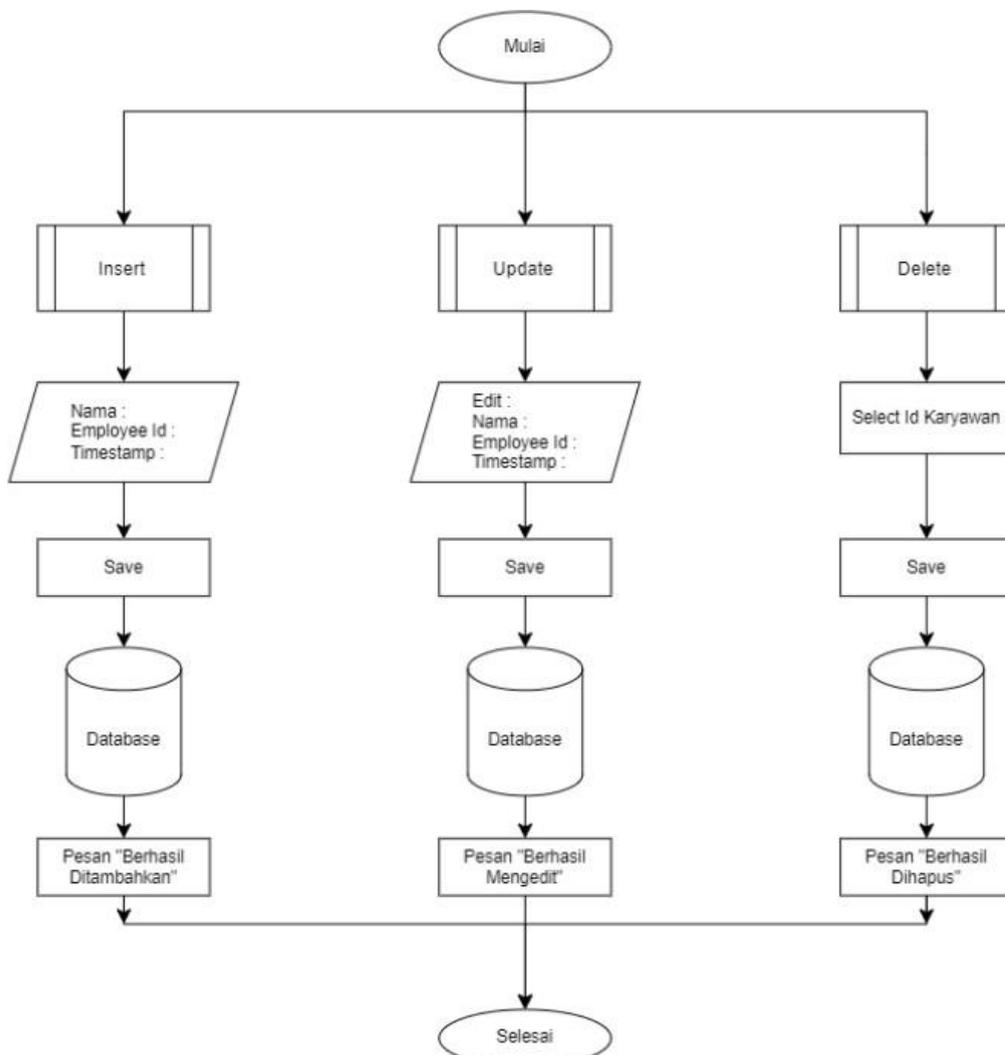
#### 5. Tabel unrecognizedFaces

- Deskripsi: Tabel ini menyimpan data wajah yang tidak dikenali saat proses absensi:
  - ID\_Unrecognized (PK): Identitas unik untuk data wajah tidak dikenali.
  - Kode\_Lokasi (FK): Relasi ke lokasi tempat wajah tidak dikenali.
  - File\_Person\_Temp: File gambar individu yang tidak dikenali.
  - File\_Muka\_Temp: File wajah individu yang tidak dikenali.
  - Jam\_Masuk: Waktu pengambilan gambar.
  - Tanggal: Tanggal pengambilan gambar.
- Hubungan:
  - Relasi FK ke tabel lokasi.

#### Hubungan Antar Tabel:

1. Relasi Tabel karyawan dengan mukaKaryawan:
  - Setiap karyawan memiliki data wajah yang tersertifikasi di tabel mukaKaryawan.
2. Relasi Tabel karyawan dengan absensi:
  - Data karyawan digunakan untuk mencatat absensi mereka.
3. Relasi Tabel lokasi dengan absensi:
  - Setiap absensi dikaitkan dengan lokasi tertentu.
4. Relasi Tabel lokasi dengan unrecognizedFaces:
  - Data wajah yang tidak dikenali dikaitkan dengan lokasi pengambilan gambar.

## f. Flowchart



Gambar 3.14 Flowchart

Dalam gambar 3.14, Flowchart tersebut menggambarkan alur manajemen data karyawan yang terdiri dari tiga aktivitas utama, yaitu \*Insert\*, \*\*Update, dan \*\*Delete. Proses \*\*Insert\* dimulai dengan pengisian data baru (Nama, Employee ID, Timestamp), yang kemudian disimpan ke dalam basis data, dan diakhiri dengan pesan notifikasi "Berhasil Ditambahkan". Proses \*Update\* melibatkan pengeditan data yang sudah ada, meliputi Nama, Employee ID, dan Timestamp, yang setelah disimpan akan memunculkan notifikasi "Berhasil Mengedit". Sementara itu, proses \*Delete\* diawali dengan pemilihan ID karyawan yang akan dihapus, dilanjutkan dengan penyimpanan perubahan pada basis data, dan

diakhiri dengan pesan "Berhasil Dihapus". Setiap proses dimulai dari titik Mulai dan berakhir di titik Selesai, dengan penekanan pada integrasi sistematis operasi CRUD yang dilengkapi notifikasi untuk memastikan transparansi status operasi.

### 3.2.3 Implementasi

Pada tahapan implementasi, praktikan menggunakan perangkat lunak dengan spesifikasi sebagai berikut:

**Tabel 3.15 Kebutuhan Perangkat Lunak**

No.	Nama Perangkat Lunak	Kebutuhan Perangkat Lunak
1.	Sistem Operasi	Window 11
2.	Database	PostgreSQL
3.	Platform administrasi dan Pengembangan	PgAdmin4

Adapun kebutuhan perangkat keras yang digunakan oleh praktikan untuk membantu dalam pengembangan aplikasi ini. Tabel di bawah ini menunjukkan kebutuhan perangkat keras yang diperlukan untuk pembuatan aplikasi tersebut.

**Tabel 3.16 Kebutuhan Perangkat Keras**

No.	Nama Perangkat Keras	Kebutuhan Perangkat Keras
1.	Processor	Intel(R) Core(TM) i5-1035G1
2.	GPU	Intel(R) UHD Graphics Family
3.	RAM	4 GB

### 3.3 Kendala Yang Dihadapi

Selain kendala ketersediaan data, mahasiswa juga menghadapi tantangan lain seperti aksesibilitas data yang terkadang terbatas karena kebijakan perusahaan terkait kerahasiaan dan proteksi data. Proses mendapatkan izin untuk mengakses data memerlukan koordinasi dan persetujuan dari beberapa level manajerial, yang sering kali memakan waktu lama.

Tantangan lainnya adalah kurangnya dukungan perangkat teknis yang memadai. Beberapa perangkat keras yang digunakan untuk pengolahan data tidak selalu tersedia secara eksklusif, sehingga mahasiswa harus berbagi dengan tim lain yang memiliki prioritas pekerjaan berbeda. Kondisi ini memperlambat proses pengerjaan dan memengaruhi jadwal penyelesaian tugas.

Mahasiswa juga menghadapi kendala teknis dalam memahami sistem yang sudah berjalan di PT. MARKTEL, yang memerlukan waktu adaptasi lebih lama dari yang diharapkan. Sistem yang digunakan memiliki kompleksitas tinggi dan belum terdokumentasi dengan baik, membuat mahasiswa harus sering berdiskusi dengan rekan-rekan kerja dan melakukan penelitian mandiri.

Selain itu, kurangnya pengalaman dalam menggunakan PostgreSQL secara mendalam menjadi tantangan tersendiri. Meskipun mahasiswa sudah memiliki pengetahuan dasar, implementasi skala besar dengan skema data yang kompleks membutuhkan pendalaman lebih lanjut. Mahasiswa perlu meluangkan waktu untuk mempelajari dan melakukan praktik langsung yang kadang harus disesuaikan dengan keterbatasan waktu kerja praktek.

Hambatan lain muncul dalam hal komunikasi dan koordinasi antar tim. PT. MARKTEL memiliki struktur organisasi yang cukup besar dan terdistribusi, sehingga memastikan semua tim memiliki pemahaman yang sama dan berkolaborasi dengan efektif memerlukan upaya yang tidak sedikit. Mahasiswa sering kali harus mengatur jadwal pertemuan yang fleksibel untuk berdiskusi dengan tim lintas departemen agar masalah yang ada dapat segera diselesaikan.

Mahasiswa juga menghadapi tantangan dalam mengelola stres dan beban kerja. Jadwal yang ketat, tuntutan menyelesaikan proyek dalam waktu terbatas, dan kebutuhan untuk terus belajar di lapangan menjadi faktor tekanan yang cukup tinggi. Hal ini menguji kemampuan mahasiswa dalam manajemen waktu dan keterampilan multitasking.

Selama pelaksanaan program KP, mahasiswa harus mampu menyesuaikan diri dengan budaya kerja di perusahaan yang berorientasi pada hasil, yang menuntut ketelitian dan ketepatan dalam setiap langkah pengerjaan. Hal ini memerlukan upaya tambahan dalam menjaga kualitas kerja dan memastikan bahwa tugas yang dikerjakan sesuai dengan ekspektasi perusahaan.

Dari segi koordinasi, perbedaan pendekatan kerja antara mahasiswa dengan karyawan yang lebih berpengalaman juga mempengaruhi proses kolaborasi. Mahasiswa harus belajar memahami dinamika tim, bagaimana menyampaikan ide dengan jelas, dan menyesuaikan gaya komunikasi untuk dapat diterima di lingkungan profesional.

Dengan adanya berbagai kendala tersebut, mahasiswa dituntut untuk

tetap proaktif dan mengembangkan solusi kreatif agar program KP berjalan dengan lancar. Mereka perlu membangun kemampuan problem-solving yang solid dan inisiatif untuk mencari jalan keluar dari tantangan yang dihadapi, seperti mencari sumber data alternatif, belajar secara mandiri, dan memanfaatkan sumber daya yang tersedia sebaik mungkin.

### **3.4 Cara Mengatasi Kendala**

Berikut ini ada beberapa cara mahasiswa dalam mengatasi kendala saat melaksanakan Kerja Profesi di PT.MARKTEL, diantaranya:

- a. Ketika mengalami kesulitan dalam mendapatkan data yang dibutuhkan secara langsung, mahasiswa dapat mencari sumber data alternatif yang relevan dengan proyek yang sedang dikerjakan. Penggunaan data sementara atau sumber data serupa dapat menjadi solusi sementara untuk melanjutkan pengerjaan proyek tanpa menunggu data utama. Setelah data utama tersedia, mahasiswa dapat memperbarui atau mengganti data sementara tersebut sesuai kebutuhan.
- b. Untuk mengatasi kendala ini, mahasiswa dapat meningkatkan komunikasi dan koordinasi dengan tim di PT. MARKTEL. Dengan melakukan diskusi yang aktif, mahasiswa dapat memperoleh informasi terkini terkait ketersediaan data yang diperlukan. Kolaborasi yang baik diharapkan dapat memberikan dukungan yang diperlukan untuk mempercepat proses pengisian *database*.

### **3.5 Pembelajaran Yang Didapat dari Kerja Profesi**

Selama melaksanakan Kerja Profesi di PT. MARKTEL, mahasiswa mendapatkan pembelajaran diantaranya:

- 1) Mahasiswa diberi pemahaman tentang pentingnya disiplin dalam hal menghargai waktu, mengingat waktu merupakan faktor yang memiliki dampak besar terhadap keberhasilan hasil pekerjaan.
- 2) Mahasiswa dilatih untuk bekerja sama dalam tim, di mana saling membantu

dalam menghadapi kendala dalam mengerjakan suatu proyek.

- 3) Mahasiswa belajar untuk menunjukkan keahlian dalam memecahkan suatu masalah, dan didorong untuk berani dalam memberikan pendapat dan kritik selama diskusi dalam tim.
- 4) Mahasiswa diajarkan untuk memahami pentingnya komunikasi yang jelas dan efektif dalam menyampaikan laporan atau hasil kerja kepada atasan, rekan tim.
- 5) Mahasiswa mendapatkan pengalaman langsung dalam penggunaan perangkat lunak dan alat teknologi yang digunakan di perusahaan, yang memperluas wawasan tentang aplikasi praktis teknologi dalam industri.
- 6) Mahasiswa didorong untuk mengambil inisiatif dalam mencari solusi terhadap masalah yang ada, tanpa menunggu arahan dari atasan, sehingga meningkatkan rasa percaya diri dan kemampuan kepemimpinan.
- 7) Mahasiswa mempelajari bagaimana menyusun prioritas kerja dengan mempertimbangkan urgensi dan dampak dari setiap tugas yang dikerjakan, yang membantu dalam manajemen waktu.
- 8) Mahasiswa memperoleh wawasan tentang pentingnya menjaga etika profesional dalam setiap aspek pekerjaan, termasuk menjaga kerahasiaan data dan informasi perusahaan.
- 9) Mahasiswa belajar untuk mengintegrasikan teori yang telah diperoleh selama masa studi dengan praktik nyata di lapangan, yang memberikan pemahaman lebih mendalam tentang relevansi pendidikan formal terhadap kebutuhan industri.
- 10) Mahasiswa mendapatkan pelajaran berharga tentang bagaimana menghadapi situasi yang penuh tekanan dan tetap menjaga fokus untuk mencapai hasil yang diinginkan.