

BAB III

PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

Pelaksanaan mata kuliah Kerja Profesi di PT. Manunggaling Rizky Karyatama Telnics (MARKTEL) dengan penempatan pada kantor pusat di Kec. Buahbatu, Kota Bandung. Praktikan bertanggung jawab pada riset yang berhubungan dengan indeks kualitas udara serta sebagai engineer dalam testing API lokasi dan pembuatan operasi *Create, Read, Update, dan Delete (CRUD)* Lokasi pada persimpangan di kota Medan, Sumatera Utara untuk Dinas Perhubungan (DISHUB) Kota Medan.

Sensor yang digunakan dalam praktik tersebut adalah HCD6818 yang ditempatkan pada titik strategis untuk pemantauan kualitas udara ambien. Untuk lebih jelasnya praktikan diberikan tanggung jawab melakukan riset terhadap sensor yang akan dikerjakan. Dalam pengerjaan proyek tersebut praktikan diwajibkan mengetahui dasar-dasar perhitungan kualitas udara dan aturan-aturan tentang standar kualitas udara dan kategori kualitas udara sesuai dengan aturan dan hukum yang berlaku untuk diaplikasikan beberapa titik persimpangan di kota Medan, Sumatera Utara.

HCD6818 merupakan sensor yang digunakan dalam proyek yang dikerjakan. HCD6818 merupakan sensor yang penggunaannya masih sangat jarang di Indonesia. Praktikan diperlukan riset terhadap sensor tersebut sehingga sensor tersebut dapat digunakan serta diaplikasikan untuk pemantauan kualitas udara.

Postman adalah sebuah platform kolaborasi untuk mengembangkan, menguji, dan memantau API (Application Programming Interface). Postman digunakan oleh pengembang perangkat lunak untuk melakukan berbagai tugas terkait API, termasuk pembuatan, pengujian, dokumentasi, dan berbagai aspek pengelolaan API. Luaran yang dihasilkan oleh Sensor merupakan sebuah data yang nantinya akan diuji coba API dengan menggunakan Postman. Dengan API data sukses dikirim melalui API Postman yang sudah tersambung dengan Server. Dengan adanya API Postman dapat memudahkan Praktikan dalam menguji dan mengelola permintaan API. Dalam hal ini digunakan dalam pengiriman data dari sensor menuju server.

3.2 Pelaksanaan Kerja

Berikut ini pelaksanaan praktik kerja yang diperoleh praktikan selama magang di MARKTEL Bandung, sebagai berikut :

3.2.1 Literasi Dan Penelitian Sesuai Dengan Proyek Yang dikerjakan

Latar belakang dalam dikembangkannya proyek ini keberadaan sistem pemantauan udara yang dapat diandalkan dan efisien memiliki peran kunci dalam membantu masyarakat dan pihak berwenang mengidentifikasi polusi udara, mengambil langkah pencegahan, dan mempertahankan kualitas udara yang bersih. DISHUB Kota Medan melakukan pemantauan menggunakan sensor HCD6818 yang mengukur secara akurat dan *real-time* terhadap kualitas udara ambien serta memberikan informasi yang sangat penting tentang tingkat polutan seperti PM2.5, PM10, CO, SO₂, NO₂, O₃, suhu (T), dan kelembaban relatif (RH). Data ini memiliki nilai yang sangat berarti dalam mengenali sumber polusi, mengukur paparan manusia, dan memahami dampaknya pada kesehatan dan lingkungan.

Dalam pengembangan suatu sistem yang berhubungan dengan masyarakat, tentu tim pengembang perlu memperhatikan hukum dan aturan-aturan yang tertera. Praktikan melakukan riset terhadap aturan yang menjadi landasan hukum untuk pengembangan proyek tersebut. Baku mutu udara ambien sesuai dengan Peraturan Pemerintah No 22 tahun 2021 TENTANG PENYELENGGARAAN, PERLINDUNGAN, DAN PENGOLAHAN LIMBAH pasal 194.

NO	PARAMETER	WAKTU PENGUKURAN	BAKU MUTU	SISTEM PENGUKURAN
1.	Sulfur Dioksida (SO ₂)	1 jam	150 µg/m ³	aktif kontinu
				aktif manual
		24 jam	75 µg/m ³	aktif kontinu
2.	Karbon Monoksida (CO)	1 tahun	45 µg/m ³	aktif kontinu
		1 jam	10000 µg/m ³	aktif kontinu
		8 jam	4000 µg/m ³	aktif kontinu
3.	Nitrogen Dioksida (NO ₂)	1 jam	200 µg/m ³	aktif kontinu
				aktif manual
		24 jam	65 µg/m ³	aktif kontinu
		1 tahun	50 µg/m ³	aktif kontinu

4.	Oksidan fotokimia (O _x) sebagai Ozon (O ₃)	1 jam	150 µg/m ³	aktif kontinu aktif manual*
		8 jam	100 µg/m ³	aktif kontinu**
		1 tahun	35 µg/m ³	aktif kontinu
5.	Hidrokarbon Non Metana (NMHC)	3 jam	160 µg/m ³	aktif kontinu***
6.	Partikulat debu < 100 µm (TSP)	24 jam	230 µg/m ³	aktif manual
		24 jam	75 µg/m ³	aktif kontinu aktif manual
	Partikulat debu < 10 µm (PM ₁₀)	1 tahun	40 µg/m ³	aktif kontinu
		24 jam	55 µg/m ³	aktif kontinu aktif manual
Partikulat debu < 2,5 µm (PM _{2,5})	1 tahun	15 µg/m ³	aktif kontinu	
	24 jam	2 µg/m ³	aktif manual	
7.	Timbal (Pb)	24 jam	2 µg/m ³	aktif manual

Gambar 3. 1 Tabel baku mutu ambien sesuai PERPU

Gambar 3.1 merupakan acuan yang digunakan untuk tolak ukur pengukuran pencemaran udara. Cara pencegahan pencemaran udara dilakukan dengan penerapan baku mutu emisi yang akan diaplikasikan sebagai acuan perhitungan pada sensor HCD 6818 proyek tersebut.

Indeks Standar Pencemaran Udara (ISPU) merupakan parameter yang digunakan sebagai standarisasi indeks pencemaran udara pada titik lokasi sensor ditetapkan. ISPU digunakan agar dapat mengkategorikan udara yang sehat hingga udara yang berbahaya. Untuk aturan yang digunakan adalah PERATURAN MENTERI LINGKUNGAN HIDUP NOMOR P.14/MENLHK/SETJEN/KUM.1/7/2020 TENTANG INDEKS STANDAR PENCEMARAN UDARA.

ISPU	PM10	PM2.5	SO2	CO	O3	NO2	HC	warna	kategori
0 – 50	50	15.5	52	4,000	120	80	45		baik
51 - 100	150	55.4	180	8,000	235	200	100		sedang
101 - 200	350	150.4	400	15,000	400	1,130	215		tidak sehat
201 - 300	420	250.4	800	30,000	800	2,260	432		sangat tak sehat
>300	500	500	1,200	45,000	1,000	3,000	648		berbahaya

Gambar 3. 2 Tabel Ispu

Gambar 3.2 merupakan parameter ISPU nantinya akan ditampilkan kepada interface sesuai dengan kategorisasi dengan aturan tersebut.

3.2.2 Riset dan Penelitian Sensor

Dalam proyek sistem monitoring kualitas udara persimpangan di Kota Medan, Sumatera Utara diperlukan alat ataupun sensor yang digunakan untuk memperoleh data dari lapangan.



Gambar 3. 3 Sensor HCD6818

Gambar 3.3 merupakan sensor HCD6818 yang diproduksi oleh Handan Yunnong Intelligent Agricultural Technology Co., LTD, yang dirancang untuk mengukur berbagai parameter meteorologi dalam lingkungan pintar, kota pintar, dan seputar polusi industri. Perangkat ini secara komprehensif dapat mengukur PM2.5, PM10, CO, SO₂, NO₂, O₃, suhu (T), dan kelembaban relatif (RH). Dengan struktur terintegrasi yang sangat baik, pemantauan online dapat dilakukan selama 24 jam terhadap parameter meteorologi di luar ruangan, dan data hasil pemantauan dapat disampaikan ke Badan Perlindungan Lingkungan (EPA). Selain itu, perangkat ini dapat diperluas fungsinya untuk mengukur H₂S, polutan organik volatil, partikulat tersuspensi total, dan berbagai sensor meteorologi lainnya seperti kecepatan angin, arah angin, tekanan atmosfer, curah hujan, dan radiasi matahari.

Proyek yang dikerjakan oleh praktikan hanya beberapa fitur yang digunakan oleh praktikan disesuaikan oleh kebutuhan *client* yaitu PM2.5, PM10, CO, SO₂, NO₂, O₃, suhu (T), dan kelembaban relatif (RH). Oleh sebab itu tim pengembang nantinya hanya berfokus kepada pengembangan fitur-fitur yang digunakan.

3.2.3 Database Lokasi Pemantauan Kualitas Udara

Dalam konteks pengembangan aplikasi perangkat lunak, peran database menjadi sangat signifikan terutama dalam pelaksanaan operasi CRUD, yang mencakup *Create*, *Read*, *Update*, dan *Delete*. Keempat operasi dasar ini menjadi landasan utama untuk memanipulasi data dalam suatu sistem. Di bawah ini, akan diuraikan peran masing-masing operasi CRUD pada database:

(1) *Create*

Melibatkan proses pembuatan data baru yang memerlukan penyimpanan di dalam database. Saat pengguna atau sistem menginisiasi operasi "*Create*", database menyediakan struktur, seperti tabel atau koleksi, untuk menampung informasi baru yang dihasilkan.

(2) *Read*

Berfokus pada pengambilan data yang sudah ada dalam database. Aplikasi dapat membaca data dari tabel atau koleksi tersebut dan mengembalikannya kepada pengguna atau proses aplikasi yang memerlukan informasi tersebut.

(3) *Update*

Menangani perubahan data yang sudah ada dalam database. Saat ada pembaruan pada data, baik itu perubahan nilai kolom atau atribut lainnya, database memastikan bahwa perubahan tersebut disimpan dan dapat diakses secara konsisten.

(4) *Delete*

Melibatkan penghapusan data dari struktur penyimpanan database. Saat operasi "*Delete*" diaktifkan, database memastikan bahwa data yang dihapus tidak lagi ada dalam sistem, menjaga kebersihan dan integritas dari data yang disimpan.

Dalam penggunaan SQL (*Structured Query Language*), bahasa khusus untuk database, perintah-perintah seperti *INSERT* untuk *Create*, *SELECT* untuk *Read*, *UPDATE* untuk *Update*, dan *DELETE* untuk *Delete* digunakan untuk menjalankan operasi-operasi tersebut. Dengan kolaborasi operasi CRUD dan database, aplikasi dapat mengelola data dengan efisien dan terstruktur. Database berperan sebagai wadah utama yang memfasilitasi interaksi antara aplikasi dan data, memastikan operasionalitas yang handal dan terorganisir.

Praktikan terlebih dahulu merancang tabel database khusus yang dirancang untuk menyimpan data informasi lokasi secara mendetail tempat perangkat yang nantinya akan diletakan dititik titik persimpangan Kota Medan dengan nama "lokasi". Tabel akan terhubung dengan Tabel Kotakab_id yang sudah dikerjakan oleh praktikan lain berdasarkan id yang ada pada tabel tersebut, Tabel Provinsi, dan zona.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2 kode	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/>	3 nama	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/>	4 kotakab_id	int(11)		Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/>	5 latitude	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/>	6 longitude	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/>	7 is_active	tinyint(1)		Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/>	8 createdAt	datetime			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	9 updatedAt	datetime			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	10 provinsi_id	int(11)		Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/>	11 zona_id	int(11)		Ya		NULL			Ubah Hapus Lainnya

Gambar 3. 4 Tabel Lokasi pada Database

Gambar 3.4 merupakan struktur dari tabel lokasi dimana terdapat 12 variabel yaitu:

- (1) id
Variabel 'id' merupakan sebuah id yang digunakan sebagai identitas unik yang ada pada kolom tersebut. variabel ini bertipe data integer dan dibatasi hingga 11 karakter. Variabel ini menggunakan *AUTO_INCREMENT* agar data dari variabel tersebut hanya bisa tercetak sekali dan tidak dapat diulang.
- (2) kode
Variabel 'kode' merupakan kode perangkat yang akan ditempati dimasing masing titik persimpangan. Variabel ini bertipe data varchar dan dibatasi hingga 255 karakter.
- (3) nama
Variabel 'nama' merupakan nama lokasi perangkat yang terpasang dipersimpangan tersebut. Variabel ini bertipe data varchar dan dibatasi hingga 255 karakter.
- (4) kotakab_id
Variabel 'kotakab_id' merupakan id lokasi kota atau kabupaten dari perangkat yang terpasang dipersimpangan tersebut. Variabel ini bertipe data integer dan dibatasi hingga 11 karakter.
- (5) latitude
Variabel 'latitude' merupakan koordinat posisi dimana tempat perangkat yang terpasang dipersimpangan tersebut. Variable ini bertipe data varchar dan dibatasi hingga 255 karakter.
- (6) longitude
Variabel 'longitude' merupakan variabel ketinggian dimana tempat perangkat yang terpasang dipersimpangan tersebut. Variabel ini bertipe data varchar dan dibatasi hingga 255 karakter.
- (7) is_active
variabel 'is_active' merupakan variabel yang menandakan status dari perangkat dipersimpangan tersebut apakah menyala atau tidaknya. Variabel ini bertipe data *tiny integer* dan dibatasi hanya 1 karakter.

- (8) *CreatedAt*
Variabel '*CreatedAt*' merupakan variabel yang menandakan kapan data perangkat dibuat. Variabel ini bertipe data *datetime* untuk menentukan waktu data dibuat.
- (9) *UpdateAt*
Variabel '*UpdateAt*' merupakan variabel yang menandakan kapan data perangkat diperbarui. Variabel ini bertipe data *datetime* untuk menentukan waktu data diperbarui.
- (10) *provinsi_id*
Variabel '*provinsi_id*' merupakan id Provinsi dimana tempat perangkat yang terpasang dipersimpangan tersebut. Variable ini bertipe data integer dan dibatasi hingga 11 karakter.
- (11) *zona_id*
Variabel '*zona_id*' merupakan idzona dimana tempat perangkat yang terpasang dipersimpangan tersebut. Variable ini bertipe data integer dan dibatasi hingga 11 karakter.

3.2.4 CRUD Lokasi Untuk Pemantauan kualitas Udara HCD 6818

Proses pengembangan sistem *CRUD* dalam pemantauan kualitas udara memerlukan urutan yang terbagi menjadi beberapa tahap. Hal ini dilakukan untuk memastikan sistem *CRUD* dapat disusun sesuai dengan kebutuhan, di antaranya.

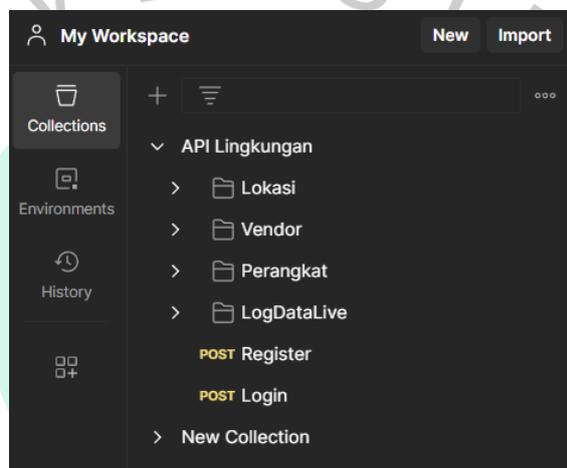
- (1) API Lingkungan
 - Dalam mengembangkan sistem *CRUD*, API Postman memiliki peran penting. Dengan menggunakan Postman, API ini menyediakan data yang diperlukan seperti lokasi, vendor, dan perangkat. Data-data ini terbagi menjadi bagian, yakni *post*, *get*, *put*, dan *del* yang memungkinkan untuk sistem *CRUD* yang telah tersedia. Data-data tersebut terbagi menjadi bagian, sebagai berikut.
 - a) Lokasi
Data lokasi ini merupakan keterangan mengenai posisi di beberapa titik sensor yang terletak di Medan. Informasi pada data lokasi ini terhubung secara langsung ke sensor, memungkinkan untuk pemantauan dan pengelolaan kualitas udara yang lebih efektif di wilayah tersebut.
 - b) Vendor
Data vendor ini berguna sebagai penyimpanan kontak person vendor, memudahkan komunikasi dan kerjasama antara pihak terkait dalam proyek pemantauan kualitas udara.
 - c) Perangkat
Data perangkat ini berguna sebagai informasi tentang keadaan dan kualitas perangkat yang berada di Medan. Hal ini memungkinkan untuk pemeliharaan dan peningkatan kinerja perangkat secara efektif, sehingga

sistem pemantauan kualitas udara dapat berjalan dengan optimal. Namun praktikan berfokus dalam pengerjaan Lokasi

(2) Pembuatan *CRUD*

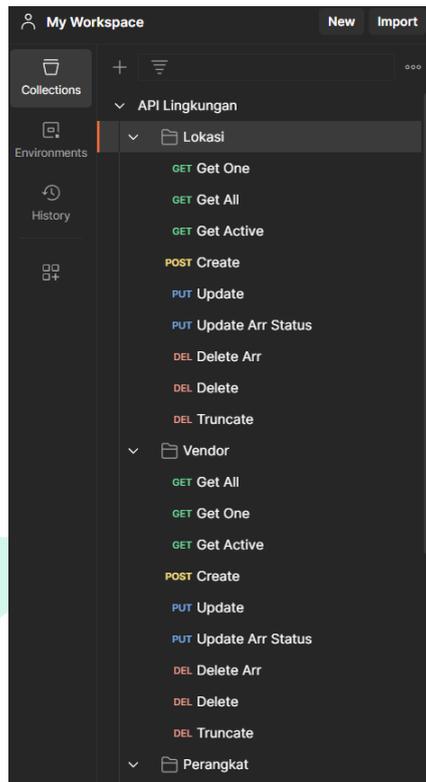
Berikut merupakan tahapan kedua bahwa adanya proses pembuatan *CRUD* menggunakan data yang di ambil dari API Postman yang telah berisi data-data yang diperlukan.

- a) Pada API Postman, data yang diinput berisi API Lingkungan yang memuat beberapa folder berisi informasi lokasi, vendor, dan perangkat.



Gambar 3. 5 Data API Lingkungan

- b) API lingkungan menyimpan informasi mengenai lokasi, vendor, dan perangkat di dalam folder terpisah. Setiap folder memiliki data yang berbeda, dan dapat dimanipulasi melalui metode GET, POST, PUT, dan *DELETE* pada masing-masing folder tersebut.

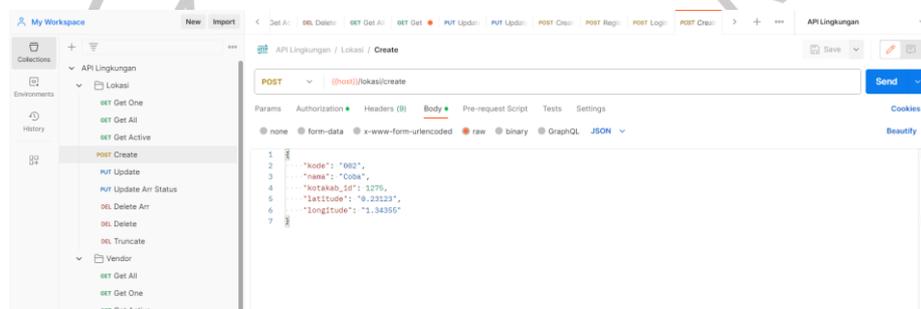


Gambar 3. 6 Post, Get, Put dan Del API Lingkungan

c) Namun, data tersebut terbagi ke dalam berbagai fungsi yang berbeda. Oleh karena itu, praktikan akan menjelaskan fungsi GET, POST, PUT dan DELETE dari data informasi lokasi, vendor dan perangkat agar data dapat dimasukkan ke dalam server.

1) Create lokasi

Pada data lokasi tersebut pertama tama dimulai dengan untuk nge Post/Create data-data yang ingin di input berisi kode, nama, kotkab_id, latitude, dan longitude



Gambar 3.7 Data Post Lokasi

- 2) Hasil inputan yang telah di buat pada Gambar 3.4, yang mencakup data pada lokasi, menyatakan data lokasi berhasil ditambahkan dan masuk ke dalam server.

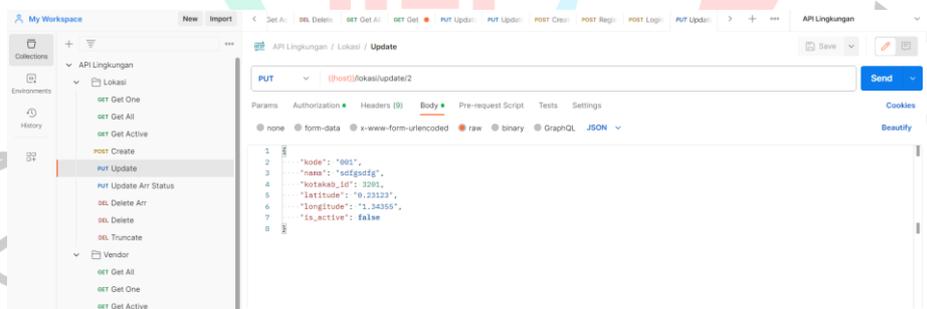
```
Lokasi_CREATE.py M X
Extension: GitHub Theme
PUT_update.py M
GET_one.py M
GET_all.py M
GET_active.py M
DELETE_arr.py

Lokasi_CREATE.py M
create_location
# Ganti dengan nama database Anda
database="db_sensor"
)
# Membuat kursor
cursor = db_connection.cursor()
print(db_connection)
# Menjalankan query INSERT untuk menyimpan data lokasi baru
insert_query = "INSERT INTO lokasi SET kode = %, nama = %, kotakab_id = %, latitude = %, longitude = %"
values = (kode, nama, kotakab_id, latitude, longitude)
cursor.execute(insert_query, values)
# Melakukan commit perubahan ke database
db_connection.commit()
print("Data Lokasi baru berhasil ditambahkan.")
except mysql.connector.Error as error:
    print(f"Terjadi kesalahan dalam menambahkan data Lokasi!: {error}")
finally:
    # Menutup koneksi ke database
    if db_connection.is_connected():
        cursor.close()
        db_connection.close()

PS C:\Users\KingBib\Desktop\vendor> .\Lokasi_CREATE.py
Masukkan kode: tst
Masukkan nama: Darmaan Santika
Masukkan kotakab_id: 91819
Masukkan latitude: 11199
Masukkan longitude: 292929
<mysql.connector.connection_cext.CMySQLConnection object at 0x00000195f8400000>
Data lokasi baru berhasil ditambahkan.
PS C:\Users\KingBib\Desktop\vendor>
```

Gambar 3. 8 Hasil Post Data Lokasi

- 3) *Update* lokasi
Pada data lokasi yang telah ditambahkan, data informasi tersebut dapat diperbarui sesuai kebutuhan jika di perlukan.



Gambar 3. 9 Data Update Lokasi

- 4) Hasil *Update* yang telah di buat pada Gambar 3.6, yang mencakup data pada lokasi, menyatakan data lokasi berhasil di *Update* dan masuk ke dalam server.

```

Lokasi > PUT_update.py > update_location
33     password="", # Ganti dengan password database Anda
34     database="db_sensor" # Ganti dengan nama database Anda
35 )
36
37 # Membuat kursor
38 cursor = db_connection.cursor()
39
40 print(db_connection)
41
42 # Menjalankan query INSERT untuk menyimpan data vendor baru
43 insert_query = "UPDATE lokasi SET kode = %, nama = %, kotakab_id = %, latitude = %, longitude = %, is_active = % WHERE
44 values = (kode, nama, kotakab_id, latitude, longitude, is_active, id)
45 cursor.execute(insert_query, values)
46
47 # Melakukan commit perubahan ke database
48 db_connection.commit()
49
50 print("Data lokasi baru berhasil di Update.")
51 except mysql.connector.Error as error:
52     print("Terjadi kesalahan dalam update data lokasi:", error)
53 finally:
54     # Menutup koneksi ke database
55     if db_connection.is_connected():
56         cursor.close()
57         db_connection.close()
58 # Memanggil fungsi untuk memperbarui data lokasi

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\KingBib\Desktop\vendor>
PS C:\Users\KingBib\Desktop\vendor> & C:/Users/KingBib/AppData/Local/Programs/Python/Python311/python.exe c:/Users/KingBib/Desktop/vendor/Lokasi/PUT_update.py
Masukkan id lokasi yang akan diperbarui: 10
masukkan kode: tdr
Masukkan nama: Darmawan Santika abillah
Masukkan kotakab id: 91819
Masukkan latitude: 11199
Masukkan longitude: 292929
Masukkan status is_active (True/False): true
-mysql.connector.connection_cext.CMySQLConnection object at 0x00000222C091810>
Data lokasi baru berhasil di Update.
PS C:\Users\KingBib\Desktop\vendor>

```

Gambar 3. 10 Hasil Update Data Lokasi

- 5) **Get lokasi**
Memilih data lokasi yang sudah di *Update*, data informasi tersebut dapat dipilih datanya, untuk membuktikan data itu masuk atau tidak ke dalam server.

```

Lokasi > GET_all.py > get_locations
17 # Menjalankan query SELECT untuk mengambil semua data lokasi
18 select_query = "SELECT * FROM lokasi"
19 cursor.execute(select_query)
20
21 # Mengambil hasil query
22 results = cursor.fetchall()
23
24 # Menampilkan data lokasi
25 for row in results:
26     print(row)
27
28 except mysql.connector.Error as error:
29     print("Terjadi kesalahan dalam mengambil data lokasi:", error)
30 finally:
31     # Menutup koneksi ke database
32     if db_connection.is_connected():
33         cursor.close()
34         db_connection.close()
35
36 # Memanggil fungsi untuk mengambil data lokasi
37 get_locations()
38
39

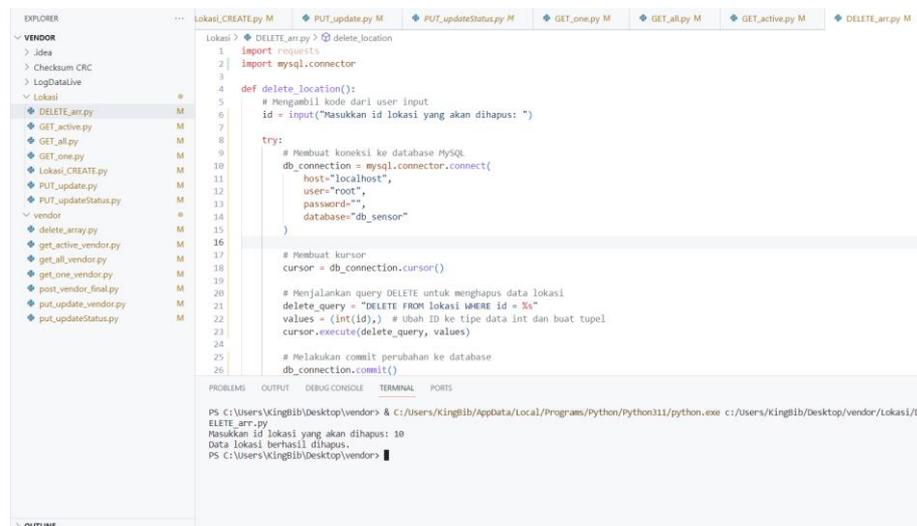
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\KingBib\Desktop\vendor> & C:/Users/KingBib/AppData/Local/Programs/Python/Python311/python.exe c:/Users/KingBib/Desktop/vendor/Lokasi/GET_all.py
(1, 'tsts', 'habib', 9101020, '191919.0', '191919.0', 1, datetime.datetime(2023, 7, 6, 4, 19, 34), datetime.datetime(2023, 7, 10, 9, 3, 9), 11, 2, 1)
(8, 'tst', 'habib', 191919, '291291.0', '1111.0', None, None, None, None, None, 1)
(9, 'tst', 'darmaan santika', 13929, '01919.0', '191919.0', None, None, None, None, None, 1)
(10, 'tdr', 'darmaan Santika abillah', 91819, '11199.0', '292929.0', 1, None, None, None, None, 1)
PS C:\Users\KingBib\Desktop\vendor>

```

Gambar 3. 11 Hasil Get Data Lokasi

- 6) **Delete Lokasi**
Pada data lokasi yang sudah di pilih, data informasi tersebut dapat di hapus datanya sesuai kebutuhan server.



```

1 import requests
2 import mysql.connector
3
4 def delete_location():
5     # Mengambil kode dari user input
6     id = input("Masukkan id lokasi yang akan dihapus: ")
7
8     try:
9         # Membuat koneksi ke database MySQL
10        db_connection = mysql.connector.connect(
11            host="localhost",
12            user="root",
13            password="",
14            database="db_sensor"
15        )
16
17        # Membuat kursor
18        cursor = db_connection.cursor()
19
20        # Menjalankan query DELETE untuk menghapus data lokasi
21        delete_query = "DELETE FROM lokasi WHERE id = %s"
22        values = (int(id),) # ubah id ke tipe data int dan buat tuple
23        cursor.execute(delete_query, values)
24
25        # Melakukan commit perubahan ke database
26        db_connection.commit()

```

PS C:\Users\Kingib\Desktop\vendor> & c:/Users/Kingib/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Kingib/Desktop/vendor/Lokasi/D ELETE_arr.py
Masukkan id lokasi yang akan dihapus: 10
Data lokasi berhasil dihapus.
PS C:\Users\Kingib\Desktop\vendor>

Gambar 3. 12 Hasil Delete Data Lokasi

3.3 Kendala Yang Dihadapi

Dalam pelaksanaan Kerja Profesi (KP) di PT. Manunggaling Rizky Karyatama Telnics(MARKTEL) praktikan mengalami beberapa permasalahan selama proses belajar dan pengerjaan

(1) Praktikan mengalami keterbatasan ilmu dalam bidang lain

Dalam mengerjakan mengerjakan proyek yang dikerjakan diperlukan ilmu lain seperti pengertian dasar tentang polusi udara serta indikator dalam menentukan kualitas udara yang baik dan buruk. Pengetahuan tentang polusi udara sangat fundamental dalam pengerjaan proyek tersebut. Sehingga diawal pengerjaan proyek praktikan kurang paham dengan proyek yang dikerjakan.

(2) Sensor HCD6818 yang terlambat datang

Praktikan mengalami kesulitan dikarenakan sensor HCD6818 yang terlambat datang sehingga praktikan masih melakukan penginputan CRUD menggunakan localhost untuk sementara waktu hingga sensor tersedia di lokasi praktek. Setelah sensor datang praktikan memerlukan penyesuaian dengan melakukan koneksi ulang keserver menggunakan API *Postman*.

3.4 Cara Mengatasi Kendala

- (1) Praktikan melakukan riset terlebih dahulu untuk mengetahui fundamental ilmu yang akan dikerjakan dalam proyek tersebut. Praktikan melakukan literasi jurnal terkait dengan proyek yang dikerjakan.
- (2) Praktikan melakukan koordinasi dan evaluasi harian kepada mentor dan rekan-rekan secara rutin untuk meminimalisir kesalahan dalam melaksanakan praktek proyek tersebut

3.5 Pembelajaran yang Diperoleh Dari Kerja Profesi

- (1) Praktikan mendapatkan pengalaman untuk terjun secara langsung kedalam dunia industri dengan proyek yang diberikan oleh perusahaan.
- (2) Praktikan diajarkan betapa pentingnya disiplin, manajemen waktu, serta inisiatif karena ketiga hal tersebut sangat berpengaruh dan signifikan dalam pengerjaan proyek tersebut
- (3) Praktikan belajar ilmu baru diluar bidang informatika sehingga menambah wawasan tentang bidang keilmuaan lain.
- (4) Praktikan belajar bagaimana bekerja secara tim serta pemecahan masalah secara efisien. Dengan faktor tersebut pengerjaan proyek dapat dikerjakan secara progresif dan efisien.