

NPM	254311011
Nama	Mohamad Malik Fajar Baihaqi
Prodi/Kelas	TRPL/1A

KEMAMPUAN AKHIR YANG DIRENCANAKAN

Mahasiswa mampu memahami dan menggunakan CRUD (Create Read Update Delete) dalam program sederhana

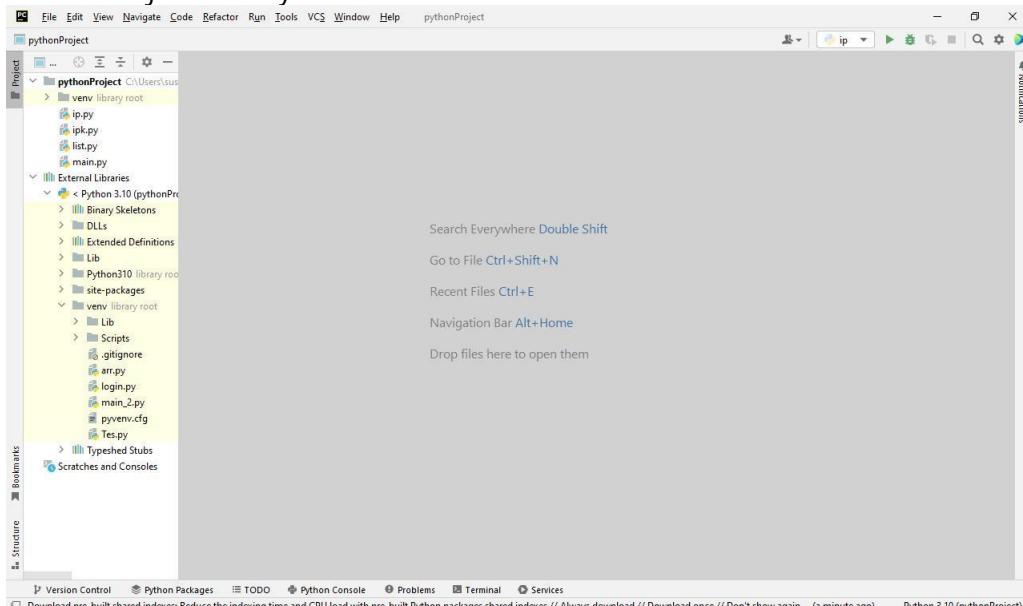
INDIKATOR

1. *Mampu memahami konsep tentang CRUD*
2. *Mampu menerapkan penggunaan queue dalam program sederhana*

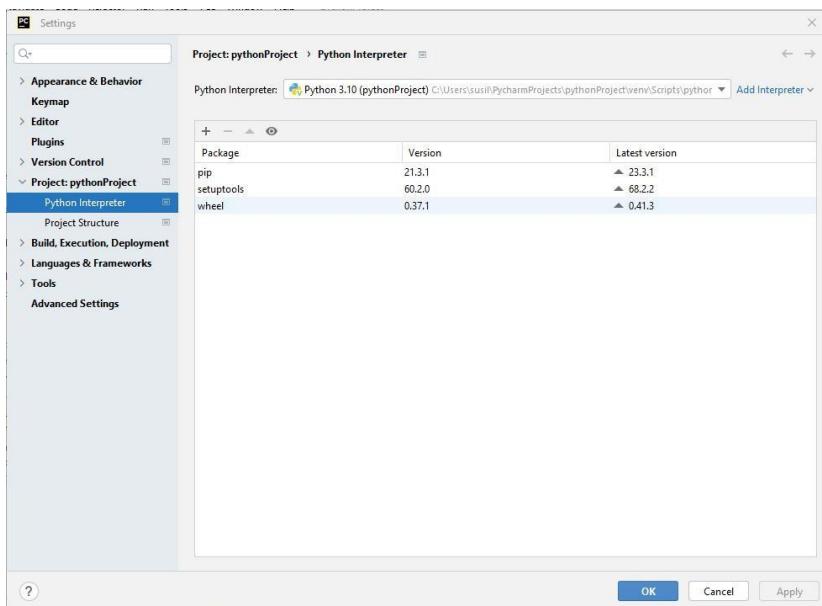
PRAKTIKUM

1. Install mysql-connector pada PyCharm

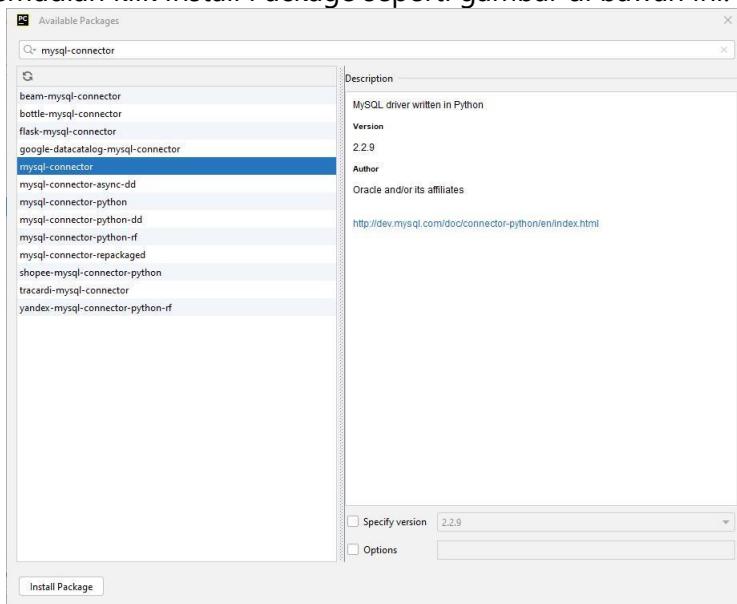
- a. Buka atau jalankan PyCharm



- b. Klik File -> Setting kemudian pilih Python Interpreter seperti gambar di bawah ini:



- c. Klik icon + untuk menambahkan package baru, pada pencarian ketikan mysql-connector, kemudian klik Install Package seperti gambar di bawah ini:



2. Membuat Koneksi Database pada MySQL

- Jalankan service MySQL, kemudian import database yang disertakan dalam praktikum ini (dapat dilakukan melalui PHP MyAdmin)
- Ketikan kode seperti di bawah ini (untuk parameter silakan disesuaikan dengan konfigurasi pada DBMS masing-masing), kemudian jalankan. Selanjutnya cek pada database.

```

1 import mysql.connector
2
3 def GetConnection():
4     return mysql.connector.connect(host='localhost', db='db_mahasiswa', user='root',
5                                     password='', port=3306)
6
7 conn = GetConnection()

```

3. Menambahkan Data Baru

- a. Ketikan kode seperti di bawah ini, kemudian jalankan.

```
1 import mysql.connector
2
3 def GetConnection():
4     return mysql.connector.connect(host='localhost', db='db_mahasiswa', user='root',
5                                     password='', port=3306)
6
7 def AddData():
8     conn = GetConnection()
9     npm = input("Masukkan NPM: ")
10    namaMahasiswa = input("Masukkan Nama Mahasiswa: ")
11    alamat = input("Masukkan Alamat: ")
12    data = (npm, namaMahasiswa, alamat)
13    query = "INSERT INTO mahasiswa (npm, namaMahasiswa, alamat) VALUES (%s, %s, %s)"
14    conn.cursor().execute(query, data)
15    conn.commit()
16
17 AddData()
```

4. Membaca Data dari Database

- a. Ketikan kode seperti di bawah ini, kemudian jalankan.

```
1 import mysql.connector
2
3 def GetConnection():
4     return mysql.connector.connect(host='localhost', db='db_mahasiswa', user='root',
5                                     password='', port=3306)
6
7 def ReadData():
8     conn = GetConnection()
9     query = "SELECT * FROM mahasiswa"
10    cursor = conn.cursor()
11    cursor.execute(query)
12    results = cursor.fetchall()
13    for row in results:
14        print(row)
15
16 ReadData()
```

5. Mengubah Data pada Database

- a. Ketikan kode seperti di bawah ini, kemudian jalankan. Selanjutnya cek pada database.

```

1 import mysql.connector
2
3 def GetConnection():
4     return mysql.connector.connect(host='localhost', db='db_mahasiswa', user='root',
5                                     password='', port=3306)
6
7 def UpdateData():
8     conn = GetConnection()
9     key = input("Masukkan NPM Lama: ")
10    npm = input("Masukkan NPM Baru: ")
11    namaMahasiswa = input("Masukkan Nama Mahasiswa Baru: ")
12    alamat = input("Masukkan Alamat Baru: ")
13    data = (npm, namaMahasiswa, alamat, key)
14    query = "UPDATE mahasiswa SET npm=%s, namaMahasiswa=%s, alamat=%s WHERE npm=%s"
15    conn.cursor().execute(query, data)
16    conn.commit()
17
18 UpdateData()

```

6. Menghapus Data dari Database

- a. Ketikan kode seperti di bawah ini, kemudian jalankan. Selanjutnya cek pada database.

```

1 import mysql.connector
2
3 def GetConnection():
4     return mysql.connector.connect(host='localhost', db='db_mahasiswa', user='root',
5                                     password='', port=3306)
6
7 def DeleteData():
8     conn = GetConnection()
9     key = input("Masukkan NPM Mahasiswa: ")
10    data = (key, )
11    query = "DELETE FROM mahasiswa WHERE npm=%s"
12    conn.cursor().execute(query, data)
13    conn.commit()
14
15 DeleteData()

```

7. Berdasarkan praktikum 1 sampai 7, buatlah aplikasi CRUD dengan mode looping dimana tampilan menu sebagai berikut:

Aplikasi CRUD Mahasiswa

1. Lihat Data
2. Tambah Data
3. Ubah Data
4. Hapus Data
5. Keluar

Masukkan pilihan anda :

Hasil Praktikum

Hasil Menu 1: Lihat Data

```
=====
SISTEM INFORMASI MANAJEMEN DATA MAHASISWA
=====
1. Lihat Data
2. Tambah Data
3. Ubah Data
4. Hapus Data
5. Keluar dari Program
-----
Pilih menu (1-5): 1
--- Tampilkan Data ---

=====
DATA MAHASISWA
=====
Tidak ada data mahasiswa ditemukan.
=====
```

Hasil Menu 2: Tambah Data

```
=====
SISTEM INFORMASI MANAJEMEN DATA MAHASISWA
=====
1. Lihat Data
2. Tambah Data
3. Ubah Data
4. Hapus Data
5. Keluar dari Program
-----
Pilih menu (1-5): 2
--- Tambahkan Data ---
Masukkan NPM: 254311011
Masukkan Nama Mahasiswa: Malik
Masukkan Alamat: JL
Data mahasiswa berhasil ditambahkan.

=====
SISTEM INFORMASI MANAJEMEN DATA MAHASISWA
=====
1. Lihat Data
2. Tambah Data
3. Ubah Data
4. Hapus Data
5. Keluar dari Program
-----
Pilih menu (1-5): 1
--- Tampilkan Data ---

=====
DATA MAHASISWA
=====
NPM: 254311011, Nama: Malik, Alamat: JL
=====
```

Hasil Menu 3: Ubah Data

```
=====
SISTEM INFORMASI MANAJEMEN DATA MAHASISWA
=====
1. Lihat Data
2. Tambah Data
3. Ubah Data
4. Hapus Data
5. Keluar dari Program
-----
Pilih menu (1-5): 3

--- Ubah Data ---
Masukkan NPM Lama (data yang ingin diubah): 254311011
Masukkan NPM Baru: 254311011
Masukkan Nama Mahasiswa Baru: Mohamad Malik Fajar Baihaqi
Masukkan Alamat Baru: JL.Apotik Hidup
Data mahasiswa dengan NPM '254311011' berhasil diupdate.

=====
SISTEM INFORMASI MANAJEMEN DATA MAHASISWA
=====
1. Lihat Data
2. Tambah Data
3. Ubah Data
4. Hapus Data
5. Keluar dari Program
-----
Pilih menu (1-5): 1

--- Tampilkan Data ---
=====

DATA MAHASISWA
=====
NPM: 254311011, Nama: Mohamad Malik Fajar Baihaqi, Alamat: JL.Apotik Hidup
NPM: 254311012, Nama: Leo Rhamanata, Alamat: JL
=====
```

Hasil Menu 4: Delete

```
=====
SISTEM INFORMASI MANAJEMEN DATA MAHASISWA
=====
1. Lihat Data
2. Tambah Data
3. Ubah Data
4. Hapus Data
5. Keluar dari Program
-----
Pilih menu (1-5): 4

--- Hapus Data ---
Masukkan NPM Mahasiswa yang ingin dihapus: 254311012
Data mahasiswa dengan NPM '254311012' berhasil dihapus.

=====
SISTEM INFORMASI MANAJEMEN DATA MAHASISWA
=====
1. Lihat Data
2. Tambah Data
3. Ubah Data
4. Hapus Data
5. Keluar dari Program
-----
Pilih menu (1-5): 1

--- Tampilkan Data ---

=====
DATA MAHASISWA
=====
NPM: 254311011, Nama: Mohamad Malik Fajar Baihaqi, Alamat: JL.Apotik Hidup
```

Hasil Menu 5: Keluar

```
=====
SISTEM INFORMASI MANAJEMEN DATA MAHASISWA
=====
1. Lihat Data
2. Tambah Data
3. Ubah Data
4. Hapus Data
5. Keluar dari Program
-----
Pilih menu (1-5): 5

Koneksi database ditutup.
Terima kasih. Program diakhiri.
```

Hasil Akhir pada database mysql:

```
mysql> select * from mahasiswa;
+-----+-----+
| npm      | namaMahasiswa           | alamat        |
+-----+-----+
| 254311011 | Mohamad Malik Fajar Baihaqi | JL.Apotik Hidup |
+-----+-----+
1 row in set (0.00 sec)
```

Screenshoot PyCharm:

The screenshot shows the PyCharm IDE interface with the following details:

- Project View:** Shows the project structure for "ProjectCRUD" with files like "ProgramCRUDMysql.py", ".venv", and "ProgramCRUD".
- Code Editor:** Displays the code for "ProgramCRUDMysql.py". The code defines a class `ConnectDatabase` with a constructor that connects to a MySQL database using `mysql.connector`. It also includes a `GetConnection` method and a try-except block for errors.
- Run Tab:** Shows the current file being run: "ProgramCRUDMysql".
- Terminal:** Displays the output of the program execution. The output shows the title "SISTEM INFORMASI MANAJEMEN DATA MAHASISWA" and a menu with options 1 through 5. The message "Pilih menu (1-5):" is visible at the bottom of the terminal window.
- Status Bar:** Shows indexing status, file encoding (UTF-8), and Python version (Python 3.13 (ProjectCRUD)).

Harmless Error dan Warning:

The screenshot shows a 'Problems' panel in a code editor with 142 items. The items are mostly warnings (yellow icons) related to Python syntax, such as 'Method must have a first parameter, usually called 'self'' at various line numbers (20, 46, 77, 108). There are also some errors (red icons) and information messages (green icons).

```
ProgramCRUDMysql.py C:\Users\user\Documents\GitHub\Semester-1_Malik\Matkul tengah semester\Praktik Algoritma Pemrograman\Laprak Algoritma Pemrograman Minggu 11\Project
① Method must have a first parameter, usually called 'self':20
① Method must have a first parameter, usually called 'self':46
① Method must have a first parameter, usually called 'self':77
① Method must have a first parameter, usually called 'self':108
⚠ Unused import statement 'import sys':2
⚠ No data sources are configured to run this SQL and provide advanced code assistance. Disable this inspection via problem menu (Alt+Enter).:33
⚠ No data sources are configured to run this SQL and provide advanced code assistance. Disable this inspection via problem menu (Alt+Enter).:53
⚠ No data sources are configured to run this SQL and provide advanced code assistance. Disable this inspection via problem menu (Alt+Enter).:91
⚠ No data sources are configured to run this SQL and provide advanced code assistance. Disable this inspection via problem menu (Alt+Enter).:119
⚠ Function name should be lowercase :7
⚠ Method 'GetConnection' may be 'static':7
⚠ Function name should be lowercase :20
⚠ Method 'AddData' may be 'static':20
⚠ Variable in function should be lowercase :28
⚠ Function name should be lowercase :46
⚠ Method 'ReadData' may be 'static':46
⚠ Function name should be lowercase :77
⚠ Method 'UpdateData' may be 'static':77
⚠ Variable in function should be lowercase :86
```

Code:

The screenshot shows a code editor with a file named 'ProgramCRUDMysql.py'. The code defines two classes: 'ConnectDatabase' and 'CreateClass'. The 'ConnectDatabase' class has a 'GetConnection' method that returns a MySQL connection object. The 'CreateClass' class has an 'AddData' method that attempts to connect to the database and add data if successful.

```
ProgramCRUDMysql.py
import mysql.connector
import sys
import time

#Deklarasi Class ConnectDatabase dan CRUD
class ConnectDatabase: 7 usages
    def GetConnection(): 1 usage

    return mysql.connector.connect(host='localhost', db='db_mahasiswa', user='root',password='', port=3306)

conn = None
try:
    conn = GetConnection()
except mysql.connector.Error as err:
    print("Koneksi database gagal saat startup.")
    print(f"Error: {err}")

class CreateClass: 1 usage
    def AddData(): 1 usage
        conn = ConnectDatabase.conn
        if not conn:
            print("Gagal: Koneksi database tidak tersedia.")
            return
```

```
26     try:
27         npm = input("Masukkan NPM: ")
28         namaMahasiswa = input("Masukkan Nama Mahasiswa: ")
29         alamat = input("Masukkan Alamat: ")
30
31         data = (npm, namaMahasiswa, alamat)
32
33         query = "INSERT INTO mahasiswa (npm, namaMahasiswa, alamat) VALUES (%s, %s, %s)"
34
35         cursor = conn.cursor()
36         cursor.execute(query, data)
37         conn.commit()
38         print("Data mahasiswa berhasil ditambahkan.")
39
40     except mysql.connector.Error as e:
41         print(f"Gagal menambahkan data. Error: {e}")
42     except Exception as e:
43         print(f"Terjadi kesalahan: {e}")
44
45 class ReadClass: 1 usage
46     def ReadData(): 1 usage
47         conn = ConnectDatabase.conn
48
49         if not conn:
50             print("Gagal: Koneksi database tidak tersedia.")
51             return
52
53         try:
54             query = "SELECT * FROM mahasiswa"
55
56             cursor = conn.cursor()
57             cursor.execute(query)
58             results = cursor.fetchall()
59
60             print("\n" + "="*50)
61             print("DATA MAHASISWA")
62             print("="*50)
63
64             if results:
65                 for row in results:
66                     print(f"NPM: {row[0]}, Nama: {row[1]}, Alamat: {row[2]}")
67             else:
68                 print("Tidak ada data mahasiswa ditemukan.")
```

```
69         print("*"*50)
70
71     except mysql.connector.Error as e:
72         print(f"Gagal membaca data. Error: {e}")
73     except Exception as e:
74         print(f"Terjadi kesalahan: {e}")
75
76 class UpdateClass: 1 usage
77     def UpdateData(): 1 usage
78         conn = ConnectDatabase.conn
79         if not conn:
80             print("Gagal: Koneksi database tidak tersedia.")
81             return
82
83         try:
84             key = input("Masukkan NPM Lama (data yang ingin diubah): ")
85             npm = input("Masukkan NPM Baru: ")
86             namaMahasiswa = input("Masukkan Nama Mahasiswa Baru: ")
87             alamat = input("Masukkan Alamat Baru: ")
88
89             data = (npm, namaMahasiswa, alamat, key)
90
91             query = "UPDATE mahasiswa SET npm=%s, namaMahasiswa=%s, alamat=%s WHERE npm=%s"
92
93             cursor = conn.cursor()
94             cursor.execute(query, data)
95             conn.commit()
96
97             if cursor.rowcount > 0:
98                 print(f"Data mahasiswa dengan NPM '{key}' berhasil diupdate.")
99             else:
100                 print(f"Data mahasiswa dengan NPM '{key}' tidak ditemukan.")
101
102         except mysql.connector.Error as e:
103             print(f"Gagal mengupdate data. Error: {e}")
104         except Exception as e:
105             print(f"Terjadi kesalahan: {e}")
106
107 class DeleteClass: 1 usage
108     def DeleteData(): 1 usage
109         conn = ConnectDatabase.conn
110         if not conn:
111             print("Gagal: Koneksi database tidak tersedia.")
112             return
113
114         try:
115             key = input("Masukkan NPM Mahasiswa yang ingin dihapus: ")
116             data = (key,)
117
```

```
118
119         query = "DELETE FROM mahasiswa WHERE npm=%s"
120
121         cursor = conn.cursor()
122         cursor.execute(query, data)
123         conn.commit()
124
125         if cursor.rowcount > 0:
126             print(f"Data mahasiswa dengan NPM '{key}' berhasil dihapus.")
127         else:
128             print(f"Data mahasiswa dengan NPM '{key}' tidak ditemukan.")
129
130     except mysql.connector.Error as e:
131         print(f"Gagal menghapus data. Error: {e}")
132     except Exception as e:
133         print(f"Terjadi kesalahan: {e}")
134
135 #Main Program(mode Loop)
136
137 def display_menu(): 1 usage
138     """Menampilkan pilihan menu CLI."""
139     print("\n" + "="*50)
140     print("SISTEM INFORMASI MANAJEMEN DATA MAHASISWA")
141     print("="*50)
142     print("1. Lihat Data")
143     print("2. Tambah Data")
144     print("3. Ubah Data")
145     print("4. Hapus Data")
146     print("5. Keluar dari Program")
147     print("-" * 50)
148
149 def main_loop(): 1 usage
150     """Loop utama aplikasi CLI."""
151
152     is_running = True
153
154     if ConnectDatabase.conn is None:
155         print("APLIKASI TIDAK DAPAT BERJALAN. Harap periksa koneksi MySQL Anda.")
156         is_running = False
157
158     while is_running:
159         display_menu()
160
161         try:
162             choice = input("Pilih menu (1-5): ").strip()
163
164             if choice == '1':
165                 print("\n--- Tampilkan Data ---")
```

```
166         ReadClass.ReadData()
167
168     elif choice == '2':
169         print("\n--- Tambahkan Data ---")
170         CreateClass.AddData()
171
172     elif choice == '3':
173         print("\n--- Ubah Data ---")
174         UpdateClass.UpdateData()
175
176     elif choice == '4':
177         print("\n--- Hapus Data ---")
178         DeleteClass.DeleteData()
179
180     elif choice == '5':
181         if ConnectDatabase.conn:
182             ConnectDatabase.conn.close()
183             print("\nKoneksi database ditutup.")
184             print("Terima kasih. Program diakhiri.")
185             is_running = False
186
187     else:
188         print("\nPilihan tidak valid. Silakan masukkan angka 1 sampai 5.")
```

```
190     except EOFError:
191         print("\nInput terhenti. Keluar dari program.")
192         is_running = False
193     except Exception as e:
194         print(f"\nKesalahan umum terjadi di loop: {e}")
195         time.sleep(1)
196
197 ➤ if __name__ == "__main__":
198     main_loop()
```