

1. Import Library

```
import sys
import sqlite3
import csv
from PyQt5.QtWidgets import ...
from PyQt5.QtCore import Qt, QDate
from PyQt5.QtGui import QFont
```

- `sys`: untuk menjalankan aplikasi.
- `sqlite3`: untuk menyimpan dan mengambil data dari database lokal.
- `csv`: untuk ekspor data ke format CSV.
- `PyQt5.QtWidgets`: menyediakan komponen GUI (form, tombol, tabel, dsb).
- `PyQt5.QtCore`: untuk fitur-fitur inti seperti tanggal (`QDate`) dan alignment (`Qt`).
- `PyQt5.QtGui`: untuk mengatur tampilan seperti font.

2. Class MoodTracker

```
def __init__(self):
    ...
```

- Merupakan konstruktor utama kelas.
- Menyiapkan jendela aplikasi (`QMainWindow`).
- Menyusun UI (`setupUI()`).
- Membuat database (`createDatabase()`).
- Menampilkan data ke dalam tabel (`loadData()`).

3. Fungsi setupUI(self)

a. menubar

```
menu_bar = self.menuBar()

file_menu = menu_bar.addMenu("File")

help_menu = menu_bar.addMenu("Help")
```

- Menu `File`: berisi **Export to CSV** dan **Exit**.

- Menu `Help`: berisi **About**.

b. Status bar

```
status_bar = QStatusBar()

status_bar.showMessage("Lalu Avryan Adjie Pratama | F1D021099")

self.setStatusBar(status_bar)
```

Menampilkan identitas pembuat aplikasi di bagian bawah jendela.

c. Form Input

```
self.date_input = QLineEdit(QDate.currentDate().toString("yyyy-MM-dd"))

self.mood_input = QComboBox()

self.mood_input.addItem("Bahagia", "Bahagia")
self.mood_input.addItem("Biasa", "Biasa")
self.mood_input.addItem("Stres", "Stres")
self.mood_input.addItem("Sedih", "Sedih")
```

Form ini berisi:

- Tanggal (otomatis hari ini, tidak bisa diedit)
- Mood (dropdown)
- Aktivitas (teks)
- Catatan (teks area)
- Status (teks)
- Tombol Simpan (berfungsi menyimpan data ke database).

d. Tabel Data

```
self.table = QTableWidgetItem()

self.table.setColumnCount(5)

self.table.setHorizontalHeaderLabels(["Tanggal", "Mood", "Aktivitas", "Catatan", "Status"])
```

- Tabel ini digunakan untuk menampilkan semua data yang sudah disimpan.
- Menggunakan warna berbeda berdasarkan mood:
 - Stres: Merah
 - Bahagia: Hijau
 - Sedih: Kuning

e. Scroll area

```
scroll_area = QScrollArea()  
...
```

Membuat bagian form bisa digulir (scroll), berguna saat aplikasi dijalankan di layar kecil.

f. Dockwidget

```
self.search_input = QLineEdit()  
self.search_input.setPlaceholderText("Cari berdasarkan aktivitas...")
```

- Kolom pencarian ini ditempatkan di sisi kanan layar menggunakan `QDockWidget`.
- Mencari data berdasarkan kata kunci di kolom **Aktivitas**.

4. Fungsi `createdatabase(self)`

```
self.conn = sqlite3.connect("mood_data.db")  
self.c = self.conn.cursor()  
self.c.execute(...)
```

Membuat file database bernama `mood_data.db`. Jika tabel `mood` belum ada, maka akan dibuat dengan kolom:

tanggal

mood

aktivitas

catatan

status

5. Fungsi `saveData(self)`

```
data = (
```

```
self.date_input.text(),  
self.mood_input.currentText(),  
...  
)  
self.c.execute("INSERT INTO mood VALUES (?, ?, ?, ?, ?)", data)
```

Mengambil data dari form input.

Menyimpan ke database SQLite.

Memanggil loadData() agar data langsung muncul di tabel.

Membersihkan form dengan clearForm().

6. Fungsi loadData(self)

```
self.c.execute("SELECT * FROM mood")  
records = self.c.fetchall()
```

Mengambil semua data dari database.

Menampilkan ke dalam tabel.

Memberi warna sel tergantung mood:

Bahagia → Hijau

Stres → Merah

Sedih → Kuning

7. Fungsi clearForm(self)

```
self.activity_input.clear()
```

...

Menghapus semua inputan (kecuali tanggal dan mood) setelah data disimpan agar siap untuk input baru.

8. Fungsi searchData(self)

```
keyword = self.search_input.text().lower()
filtered = [r for r in records if keyword in r[2].lower()]
```

Mengambil semua data dari database.

Menyaring data berdasarkan keyword dari kolom aktivitas.

Hasil pencarian langsung ditampilkan di tabel.

9. Fungsi exportToCSV(self)

```
path, _ = QFileDialog.getSaveFileName(...)
```

Membuka dialog untuk memilih lokasi dan nama file .csv.

Mengambil semua data dari database dan menyimpannya ke file CSV.

Menampilkan pesan berhasil jika proses ekspor sukses.

10. Fungsi showAbout(self)

```
QMessageBox.information(...)
```

Menampilkan informasi pembuat aplikasi (nama dan NIM) dalam bentuk pop-up.

11. Menjalankan Aplikasi

```
if __name__ == "__main__":  
    app = QApplication(sys.argv)  
    window = MoodTracker()  
    window.show()  
    sys.exit(app.exec_())
```

Membuat dan menjalankan aplikasi.

Menampilkan jendela utama.