

Nama : Muhammad Fajar Maulana

NIM : F1D022072

Kelas : D

### Mini Project My To-Do List

#### Pemrograman Visual

Aplikasi “Japar’s To-Do List” adalah aplikasi berbasis PyQt5 yang dibuat untuk membantu Japar dalam mengelola tugas-tugasnya yang harus diselesaikan dengan cara mencatat, menetapkan prioritas, dan menetapkan tanggal serta waktu penyelesaian tugas. Aplikasi ini memungkinkan Japar untuk menambahkan tugas baru, menetapkan tingkat prioritas, menentukan batas waktu penyelesaian, serta menyertakan deskripsi tugas. Selain itu, aplikasi ini menyediakan fitur untuk menghapus tugas yang sudah selesai dan membersihkan seluruh daftar tugas. Semua data tugas dapat dilihat dalam sebuah tabel yang memudahkan Japar untuk mengelola tugasnya.

Langkah-langkah pembuatan project dimulai dengan pembuatan UI pada aplikasi QtDesigner yang meliputi semua elemen antarmuka yang diperlukan. Elemen tersebut terdiri dari input form (QLineEdit untuk nama tugas dan deskripsi), QComboBox untuk memilih prioritas dan hari, QDateEdit untuk memilih tanggal, QTimeEdit untuk menentukan waktu tugas, dan QTableWidgetItem untuk menampilkan daftar tugas yang sudah ditambahkan. Interaksi antar komponen dilakukan menggunakan QPushButton yang terhubung dengan fungsi menggunakan Signal & Slot. Layout yang digunakan adalah Form Layout untuk pengaturan form input dan QTableWidgetItem untuk menampilkan daftar tugas. Selain itu, elemen-elemen seperti QLabel juga digunakan untuk menampilkan informasi terkait aplikasi to do list ini. Berikut ini kesesuaian dengan required features

1. Input Form: Use at least 3 different input components: QLineEdit, QComboBox, QSpinBox, QSlider, QCheckBox, QRadioButton, etc.
  - a. QLineEdit: Digunakan untuk mengisi deskripsi tugas.
  - b. QComboBox: untuk memilih prioritas tugas(Rendah, Sedang, Tinggi).
  - c. QDateEdit: untuk memilih tanggal penyelesaian tugas dengan kalender.
  - d. QTimeEdit: Digunakan untuk memilih waktu penyelesaian tugas dalam format jam dan menit.

2. Component Interaction: Use Signals and Slots to connect user actions to the application logic

- a. Digunakan untuk menambahkan tugas baru ke dalam tabel setelah memvalidasi input dari pengguna

```
self.ui.btn_add_task.clicked.connect(self.add_task)
```

- b. Digunakan untuk menghapus semua tugas dari dalam table

```
self.ui.btn_clear_task.clicked.connect(self.clear_tasks)
```

3. Proper Layout: Use appropriate layout structures (QVBoxLayout, QHBoxLayout, or Grid), either manually or via Qt Designer

- a. QFormLayout: Digunakan untuk mengatur form input pada aplikasi To Do List.

- b. QVBoxLayout: Sebagai layout utama untuk aplikasi To Do List.

- c. QHBoxLayout: Digunakan untuk menyusun tombol-tombol Add dan Clear.

4. Output Display: Use QListWidget, QTableWidgetItem, QTextEdit, or QLabel to show results from user input or interaction

- a. QTableWidgetItem untuk menampilkan daftar tugas yang telah ditambahkan.

5. Menu, Toolbar, or Dialog: Include at least one dialog (QMessageBox, QInputDialog) or a simple menu (QMenuBar)

- a. QMessageBox untuk dialog peringatan

```
QtWidgets.QMessageBox.warning(self, "Peringatan", "Semua field harus diisi dengan benar!")
```

6. Student ID (NIM) and Name: The student ID and Student name must be shown on the interface and must not be editable by users

```
self.id_student=QtWidgets.QLabel(self.formLayoutWidget)
.....
self.id_student.setText(_translate("MainWindow", "Muhammad Fajar Maulana | F1D022072"))
```

7. StyleSheet: Use setStyleSheet() on at least one component, such as a label, button, or background to enhance visual appearance

```
self.ui.btn_add_task.setStyleSheet("background-color: #333; color: white; font-size: 12pt; padding: 10px; border-radius: 5px;")
```

```
self.table_todo.setStyleSheet("border: 1px solid #ddd; background-color: #f7f7f7; font-size: 10pt;")
```

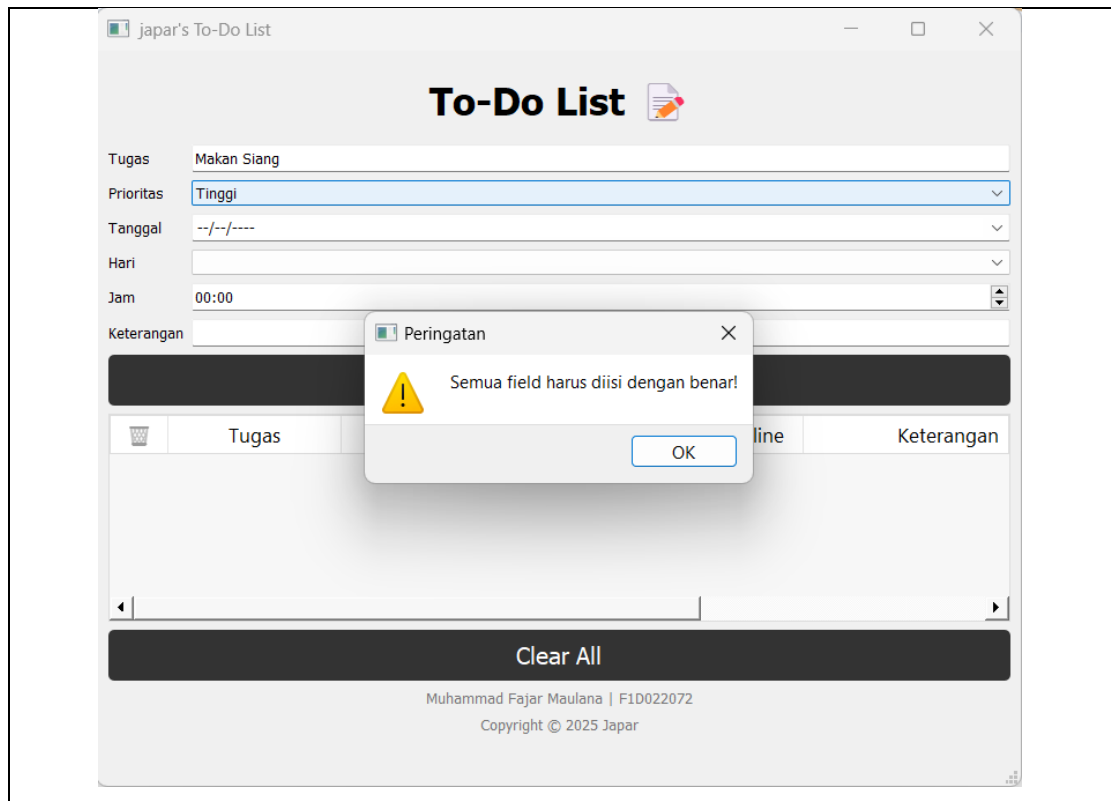
```
self.btn_clear_task.setStyleSheet("background-color: #333; color: white; font-size: 12pt; padding: 10px; border-radius: 5px;")
```

Aplikasi ini memiliki beberapa fungsi utama, yaitu `add_task()` yang mengecek input pengguna dan menambahkan tugas baru ke dalam tabel jika semua informasi diisi dengan benar, `delete_row()` yang berfungsi untuk menghapus baris tugas tertentu saat tombol delete pada baris ditekan, `delete_task()` yang memungkinkan pengguna menghapus tugas dari tabel ketika tombol delete (✖) di kolom pertama diklik, serta `clear_tasks()` yang menghapus seluruh daftar tugas dalam tabel sehingga aplikasi kembali ke kondisi semula. Berikut adalah tampilan hasil run dari aplikasi Japar's To-Do List

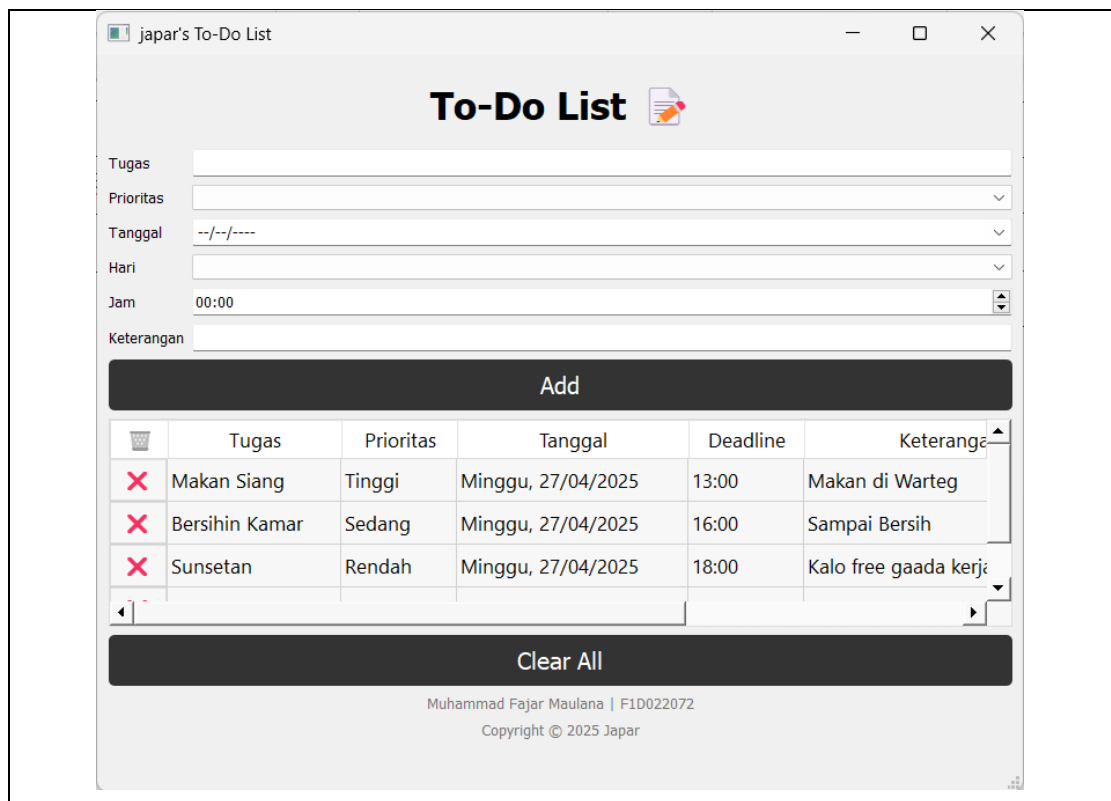
1. Tampilan awal

The screenshot shows the initial interface of the 'japar's To-Do List' application. The window title is 'japar's To-Do List'. The main heading is 'To-Do List' with a pencil icon. The form includes input fields for 'Tugas', 'Prioritas', 'Tanggal' (displaying '--/--/----'), 'Hari', 'Jam' (displaying '00:00'), and 'Keterangan'. A dark 'Add' button is positioned below the form. Underneath the button is a table with the following columns: 'Tugas', 'Prioritas', 'Tanggal', 'Deadline', and 'Keterangan'. The table is currently empty. Below the table is a dark 'Clear All' button. The footer contains the text 'Muhammad Fajar Maulana | F1D022072' and 'Copyright © 2025 Japar'.

2. Tampilan invalid ketika masih ada field yang kosong



3. Tampilan setelah menambah tugas



4. Tampilan setelah menghapus salah satu tugas

japar's To-Do List

## To-Do List

Tugas

Prioritas

Tanggal

--/--/----





Hari

Jam

00:00

Keterangan

Add

	Tugas	Prioritas	Tanggal	Deadline	Keterangan
	Makan Siang	Tinggi	Minggu, 27/04/2025	13:00	Makan di Warteg
	Bersihkan Kamar	Sedang	Minggu, 27/04/2025	16:00	Sampai Bersih
	PemVisual	Tinggi	Minggu, 27/04/2025	23:59	Mini Project UTS

Clear All

Muhammad Fajar Maulana | F1D022072

Copyright © 2025 Japar