LAPORAN PROYEK

APLIKASI TULIS

PEMROGRAMAN VISUAL

Oleh:

Hendra Ahmad Yani

F1D022122

TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MATARAM

2025

Deskripsi Aplikasi

TULIS adalah aplikasi manajemen tugas berbasis PyQt5 yang dibuat untuk membantu pengguna dalam mencatat, mengelola, dan mengorganisasi kegiatan harian mereka. Aplikasi ini dibangun sepenuhnya menggunakan pemrograman Python tanpa bantuan Qt Designer, dengan antarmuka yang modern dan navigasi antar halaman menggunakan QStackedWidget.

Langkah-langkah Pembuatan Proyek

1. Membuat struktur awal aplikasi menggunakan QMainWindow, QStackedWidget, dan Qwidget.
2. Membuat halaman Welcome, Daftar Kegiatan, dan Tambah Kegiatan Baru secara manual menggunakan PyQt5 dengan 3 struktur layout yaitu, QVBoxLayout, QHBoxLayout, dan QGridLayout.
3. Menampilkan halaman Welcome dengan tombol mulai, logo aplikasi dan background bergerak dengan QPushButton, QIcon, QTimer, dan QPainter.
4. Menambahkan Nama dan Nim di halaman Welcome Page
5. Membuat halaman Daftar kegiatan.
6. Menampilkan Daftar task yang telah ada dengan menggunakan QScrollArea di Halaman Daftar Kegiatan dengan pengaturan menggunakan QFrame
7. Menambahkan fitur filter task berdasarkan tanggal, prioritas, dan waktu pelaksanaan dengan QToolButton pada halaman Daftar Kegiatan.
8. Menambahkan fitur pencarian task berdasarkan keyword dengan QlineEdit pada halaman Daftar Kegiatan.
9. Menambahkan fitur action pada task seperti Delete, View Detail, dan Edit Task dengn QToolButton, QinputDialog, dan QMenu pada halaman Daftar Kegiatan.
10. Membuat Tombol tambah task baru menuju halaman Tambah Kegiatan Baru d halaman Daftar Kegiatan.
11. Membuat fungsi tambah task baru dengan validasi input dengan QMessageBox, dan QDialog di Halaman Tambah Kegiatan Baru.
12. Membuat form pengisian dengan QLabel, QLineEdit, QTextEdit, QComboBox, QLabel, QSpinBox, Qslider, QRadioButton, QcheckBox, QpushButton.
13. Mendesain tampilan menggunakan setStyleSheet() untuk setiap tombol dan halaman agar lebih modern.
14. Menambahkan animasi transisi antar halaman dengan QPropertyAnimation.

Penjelasan Fungsi Utama yang Digunakan

A. `Class WelcomePage(QWidget)`, memiliki beberapa Fuction/Method diantaranya:

- `__init__(self, switch_func)`, adalah konstruktor yang digunakan untuk menginisialisasi objek dari kelas `WelcomePage()`. Selain itu membuat dan mengatur tampilan halaman welcome, menerima fungsi, mengatur animasi latar belakang.
- `paintEvent(self, event)`: Mengatur perubahan ukuran icon yang dapat bergerak sebagai latar belakang untuk Halaman Welcome Page.
- `update_background_positions(self)`: Mengatur perubahan posisi icon yang dapat bergerak sebagai latar belakang untuk Halaman Welcome Page. Selain itu Fuction ini juga untuk reset icon saat keluar dari layar dan mengatur agar icon tidak tumpang tindih satu sama lain saat muncul.
- `is_overlapping(self, rect1, rect2)`: Method ini digunakan untuk memeriksa apakah dua objek persegi panjang (rectangles) saling bertabrakan (overlap) berdasarkan koordinat dan ukuran icon.

B. `Class ToDoCard(QFrame)`, **memiliki Fuction/Method yaitu:**

- `__init__(self, title, description, created_time=None)`, digunakan untuk menginisialisasi sebuah kartu tugas (task card) dengan judul, deskripsi, dan waktu pembuatan. Kartu ini dirancang sebagai elemen visual dalam aplikasi, dengan tampilan yang mencakup judul, deskripsi, dan waktu pembuatan, serta memiliki gaya khusus seperti border radius dan efek bayangan.

C. `Class MainPage(QWidget)`, **memiliki beberapa Fuction/Method diantaranya:**

- `__init__(self, switch_to_add)`, digunakan untuk menginisialisasi halaman utama aplikasi yang menampilkan daftar tugas (to-do list). Konstruktor ini mengatur elemen-elemen seperti header, tombol aksi, filter pencarian, area scroll untuk daftar tugas, dan tombol melayang untuk menambahkan tugas baru. Parameter `switch_to_add` adalah fungsi callback yang digunakan untuk berpindah ke halaman tambah tugas saat tombol melayang ditekan.
- `resizeEvent(self, event)`, digunakan untuk menangani perubahan ukuran jendela aplikasi yaitu memastikan bahwa tombol melayang untuk tambah kegiatan baru tetap berada di posisinya, yaitu di pojok kanan bawah jendela, setiap kali ukuran jendela berubah. Method ini dipanggil secara otomatis oleh Qt saat jendela diubah ukurannya, dan dengan memanggil `super().resizeEvent(event)`.
- `add_task()` ,digunakan untuk menambahkan tugas baru ke daftar tugas (to-do list). Method ini membuat kartu tugas (`ToDoCard`) berdasarkan informasi seperti judul, deskripsi, prioritas, durasi, dan waktu pelaksanaan, lalu menambahkannya ke grid layout di halaman utama. Jika tugas ditandai sebagai penting, kartu tugas akan ditempatkan di urutan pertama; jika tidak, kartu akan ditambahkan di akhir. Method ini

juga menyimpan data tugas ke dalam atribut `self.tasks` untuk mendukung fitur pencarian dan filter.

- `filter_tasks(self)` , digunakan untuk memfilter daftar tugas berdasarkan kriteria seperti teks pencarian, tanggal, prioritas, dan waktu pelaksanaan. Method ini menghapus semua tugas dari tampilan grid layout, lalu menambahkan kembali tugas-tugas yang sesuai dengan kriteria filter yang dipilih oleh pengguna. Hal ini memungkinkan pengguna untuk dengan mudah menemukan tugas tertentu dalam daftar.
- `enter_delete_mode(self)` , digunakan untuk mengaktifkan mode seleksi penghapusan tugas. Dalam mode ini, pengguna dapat memilih tugas yang ingin dihapus dengan menampilkan border merah pada setiap kartu tugas (`ToDoCard`) untuk menandai bahwa tugas tersebut dapat dipilih. Ketika tugas dipilih, fungsi `delete_task` akan dipanggil untuk menghapus tugas tersebut. Selain itu, ikon tombol melayang diubah menjadi ikon "back" untuk keluar dari mode ini.
- `exit_select_mode(self)`, digunakan untuk keluar dari mode seleksi (seperti mode edit, delete, atau view) dan mengembalikan halaman ke tampilan normal. Method ini mengatur ulang elemen-elemen seperti header, ikon tombol melayang, dan gaya kartu tugas (`ToDoCard`) dengan menghapus border atau highlight yang ditambahkan selama mode seleksi. Selain itu, method ini juga mengembalikan fungsi tombol melayang ke fungsi default, yaitu untuk menambahkan tugas baru.
- `delete_task(self, widget)`, digunakan untuk menghapus tugas tertentu dari daftar tugas (to-do list). Ketika dipanggil, method ini menampilkan dialog konfirmasi kepada pengguna untuk memastikan penghapusan. Jika pengguna mengonfirmasi, tugas yang terkait dengan widget (kartu tugas) akan dihapus dari layout grid dan daftar `self.tasks`. Setelah itu, posisi tugas yang tersisa diatur ulang menggunakan method `rearrange_tasks`.
- `enter_view_mode(self)`, digunakan untuk mengaktifkan mode tampilan (view mode), di mana pengguna dapat melihat detail tugas dengan mengklik kartu tugas (`ToDoCard`). Dalam mode ini, setiap kartu tugas diberi border biru untuk menandakan bahwa kartu tersebut dapat diklik. Ketika kartu diklik, method `handle_view_task_click` dipanggil untuk menampilkan detail tugas menggunakan dialog informasi. Selain itu, ikon tombol melayang diubah menjadi ikon "back" untuk keluar dari mode ini.
- `handle_view_task_click(self, widget, event)`, digunakan untuk menangani klik pada kartu tugas (`ToDoCard`) saat berada dalam mode tampilan (view mode). Ketika pengguna mengklik sebuah kartu tugas, method ini memanggil `show_task_details(widget)` untuk menampilkan detail tugas tersebut dalam bentuk dialog informasi. Method ini memastikan bahwa pengguna dapat melihat informasi lengkap dari tugas yang dipilih.

- `rearrange_tasks(self)`,digunakan untuk mengatur ulang posisi semua kartu tugas (`ToDoCard`) di grid layout setelah ada perubahan, seperti penghapusan tugas. Method ini menghapus semua widget dari grid layout, lalu menambahkan kembali kartu tugas yang tersisa berdasarkan urutan dalam daftar `self.tasks`, dengan memastikan tata letak tetap rapi dalam format grid.
- `view_task(self)`, digunakan untuk menampilkan detail tugas saat pengguna mengklik sebuah kartu tugas (`ToDoCard`). Method ini mengatur setiap kartu tugas agar dapat menangani event klik dengan memanggil `show_task_details(widget)`, yang akan menampilkan informasi lengkap tugas dalam dialog informasi. Method ini memastikan pengguna dapat melihat detail tugas tanpa mengubah atau menghapusnya.
- `show_task_details(self, widget)`, digunakan untuk menampilkan detail lengkap dari sebuah tugas yang dipilih oleh pengguna. Method ini mencari tugas yang sesuai dengan widget (`ToDoCard`) yang diklik, lalu menampilkan informasi seperti judul, deskripsi, prioritas, durasi, waktu pelaksanaan, dan waktu pembuatan dalam sebuah dialog informasi (`QMessageBox`). Hal ini memungkinkan pengguna untuk melihat rincian tugas tanpa mengeditnya.
- `enter_edit_mode(self)`, digunakan untuk mengaktifkan mode edit, di mana pengguna dapat memilih tugas yang ingin diedit. Dalam mode ini, setiap kartu tugas (`ToDoCard`) diberi border hijau untuk menandakan bahwa kartu tersebut dapat dipilih. Ketika pengguna mengklik kartu tugas, method `handle_edit_task_click` dipanggil untuk membuka form edit, memungkinkan pengguna mengubah judul, deskripsi, dan waktu pelaksanaan tugas. Selain itu, ikon tombol melayang diubah menjadi ikon "`back`".
- `handle_edit_task_click(self, widget, event)`, digunakan untuk menangani klik pada kartu tugas (`ToDoCard`) saat berada dalam mode edit. Ketika pengguna mengklik sebuah kartu tugas, method ini mencari tugas yang sesuai dengan widget yang diklik, lalu memanggil `show_edit_form(task)` untuk membuka form edit, sehingga pengguna dapat mengubah judul, deskripsi, dan waktu pelaksanaan tugas tersebut.
- `show_edit_form(self, task)` , digunakan untuk menampilkan form edit yang memungkinkan pengguna mengubah informasi tugas tertentu, seperti judul, deskripsi, dan waktu pelaksanaan. Method ini memvalidasi input pengguna untuk memastikan bahwa judul dan deskripsi tidak kosong, serta memperbarui data tugas di daftar `self.tasks` dan tampilan kartu tugas (`ToDoCard`) setelah perubahan dilakukan. Jika pengguna membatalkan dialog, perubahan tidak akan diterapkan.

D. `Class AddTaskPage(QWidget)`, **memiliki beberapa Fuction/Method diantaranya:**

- `__init__(self,add_task_callback,back_to_main_callback)` , digunakan untuk menginisialisasi halaman tambah tugas, yang memungkinkan pengguna memasukkan informasi tugas baru seperti judul, deskripsi, kategori, durasi, prioritas, dan waktu

pelaksanaan. Parameter `add_task_callback` adalah fungsi yang dipanggil untuk menyimpan tugas baru, sedangkan `back_to_main_callback` adalah fungsi untuk kembali ke halaman utama. Konstruktor ini juga mengatur tata letak elemen input dan tombol simpan dengan validasi input yang diperlukan.

- `save_task(self)`, digunakan untuk menyimpan tugas baru yang dimasukkan oleh pengguna. Method ini mengambil data dari input pengguna seperti judul, deskripsi, kategori, durasi, prioritas, dan waktu pelaksanaan, lalu memvalidasi input tersebut. Jika validasi berhasil, method memanggil fungsi callback `add_task_callback` untuk menambahkan tugas ke daftar tugas di halaman utama, kemudian kembali ke halaman utama menggunakan `back_to_main_callback` dan mengosongkan semua input dengan `clear_inputs()`.
- `clear_inputs(self)`, digunakan untuk mengosongkan semua input pada halaman tambah tugas setelah tugas berhasil disimpan atau saat pengguna kembali ke halaman utama. Method ini mereset nilai input seperti judul, deskripsi, kategori, durasi, prioritas, dan waktu pelaksanaan ke nilai default, serta memastikan checkbox dan radio button kembali ke kondisi awal. Hal ini bertujuan agar halaman tambah tugas siap digunakan untuk memasukkan tugas baru tanpa data dari tugas sebelumnya.

E. `Class NumberSlider(QSlider)`, **memiliki beberapa Fuction/Method diantaranya:**

- `__init__(self, orientation, parent=None)` , digunakan untuk menginisialisasi slider horizontal dengan nilai numerik yang ditampilkan di atas handle slider. Konstruktor ini mengatur orientasi slider, rentang nilai (0 hingga 10), nilai default (0), dan gaya visual slider menggunakan stylesheet. Method ini memastikan bahwa slider memiliki tampilan modern dan fungsionalitas untuk menampilkan angka secara dinamis saat slider digeser.
- `paintEvent(self, event)` , digunakan untuk menggambar angka di atas handle slider, yang menunjukkan nilai saat ini dari slider. Method ini memanfaatkan `QPainter` untuk menggambar teks angka dengan font khusus dan warna putih di posisi tengah handle slider. Dengan ini, pengguna dapat melihat nilai slider secara langsung saat menggeser, memberikan pengalaman visual yang lebih informatif dan interaktif.

F. `Class WaktuPelaksanaanDialog(QDialog)`, **memiliki beberapa Fuction/Method diantaranya:**

- `__init__(self, parent=None, current_value="Pagi")` ,digunakan untuk menginisialisasi dialog kustom yang memungkinkan pengguna memilih waktu pelaksanaan tugas, seperti "Pagi", "Siang", atau "Malam". Konstruktor ini membuat elemen-elemen seperti label, dropdown menu (`QComboBox`) untuk pilihan waktu, serta tombol **OK** dan **Cancel** untuk mengonfirmasi atau membatalkan pilihan. Nilai default waktu pelaksanaan dapat diatur melalui parameter `current_value`, sehingga dialog dapat menampilkan pilihan yang sesuai dengan data tugas yang sedang diedit.

- `get_selected_value(self)` , digunakan untuk mengambil nilai waktu pelaksanaan yang dipilih oleh pengguna dari dropdown menu (`QComboBox`). Method ini mengembalikan teks pilihan saat ini, seperti "Pagi", "Siang", atau "Malam", yang kemudian dapat digunakan untuk memperbarui data tugas atau keperluan lainnya. Hal ini memungkinkan integrasi nilai yang dipilih pengguna ke dalam logika aplikasi.

G. `Class MainWindow(QStackedWidget)`, **memiliki beberapa Fuction/Method diantaranya:**

- `__init__(self)` , digunakan untuk menginisialisasi jendela utama aplikasi, yang mengelola navigasi antar halaman menggunakan `QStackedWidget`. Konstruktor ini membuat tiga halaman utama, yaitu **WelcomePage**, **MainPage**, dan **AddTaskPage**, menambahkan halaman-halaman tersebut ke dalam stack, serta mengatur fungsi navigasi antar halaman. Selain itu, method ini juga menambahkan data dummy untuk mengisi daftar tugas di halaman utama sebagai contoh awal.
- `add_dummy_tasks(self)` , digunakan untuk menambahkan data tugas dummy (contoh tugas) ke halaman utama aplikasi. Method ini membuat beberapa tugas dengan informasi seperti judul, deskripsi, prioritas, durasi, dan waktu pelaksanaan, lalu memanggil method `add_task` di `MainPage` untuk menambahkan tugas-tugas tersebut ke daftar tugas. Hal ini bertujuan untuk memberikan contoh awal kepada pengguna saat aplikasi pertama kali dijalankan.
- `animate_switch(self, index)` , digunakan untuk memberikan efek animasi transisi saat berpindah antar halaman di dalam aplikasi. Method ini memindahkan widget halaman berikutnya ke posisi awal di luar layar, lalu menggunakan animasi `QPropertyAnimation` untuk menggeser halaman tersebut ke posisi tengah layar dengan efek transisi yang halus. Hal ini memberikan pengalaman visual yang lebih menarik saat pengguna berpindah antar halaman, seperti dari halaman utama ke halaman tambah tugas/sebaliknya.
- `show_main_page(self)` ,digunakan untuk berpindah ke halaman utama aplikasi, yaitu **MainPage**, dengan efek animasi transisi. Method ini memanggil `animate_switch(1)`, di mana indeks 1 merujuk ke halaman utama dalam `QStackedWidget`. Hal ini memungkinkan navigasi yang mulus dari halaman lain, seperti **WelcomePage** atau **AddTaskPage**, ke halaman utama.
- `show_add_page(self)` , digunakan untuk berpindah ke halaman tambah tugas, yaitu **AddTaskPage**, dengan efek animasi transisi. Method ini memanggil `animate_switch(2)`, di mana indeks 2 merujuk ke halaman tambah tugas dalam `QStackedWidget`. Hal ini memungkinkan pengguna untuk menambahkan tugas baru dengan navigasi yang mulus dari halaman lain, seperti **MainPage**.
- `add_task(self, title, description, important=False, priority=1, duration=0, duration_unit="Hari", waktu_pelaksanaan="Pagi")` , digunakan untuk menambahkan tugas baru ke halaman utama aplikasi. Method ini memanggil

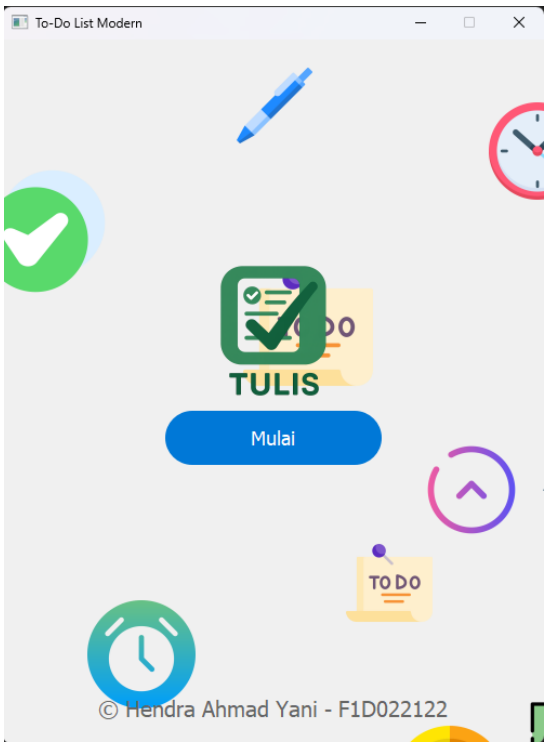
method `add_task` di **class** `MainPage` untuk membuat kartu tugas baru berdasarkan data yang diberikan, seperti judul, deskripsi, prioritas, dan waktu pelaksanaan

Cara Kerja Aplikasi TULIS

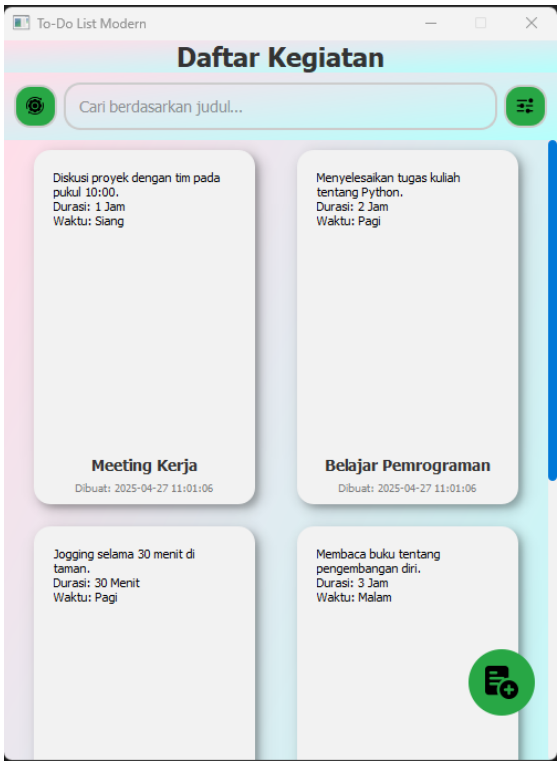
- Halaman Welcome Page : User akan pertama kali di bawa ke halam ini, user dapat melakukan klik tombol mulai untuk menuju halaman Utama (Daftar Kegiatan)
- Halaman Daftar Kegiatan: User dapat melihat tampilan daftar kegiatan yang dapat di scroll secara vertikal. Terdapat fitur pencarian yang dapat digunakan untuk mencari kegiatan yang diinginkan, terdapat juga fitur filter di bagian atas kanan berdasarkan filter tanggal, prioritas, dan waktu pelaksanaan kegiatan. Tedapat fitur Ation yang terdiri dari Delete (untuk menghapus kegiatan, Cara menghapus yaitu masuk ke mode delete dengan menekan action button di kiri atas, pilih kegiatan yang akan di hapus, klik “OK” untuk melanjutkan menghapus, kembali ke halaman utama dengan klik tombol kembali di pojok kanan bawah), View (untuk melihat detail Kegiatan, Cara melihat detail kegiatan yaitu masuk ke mode view dengan menekan action button di kiri atas, pilih kegiatan yang akan di lihat detailnya, kembali ke halaman utama dengan klik tombol kembali di pojok kanan bawah), Edit (untuk merubah atau mengedit judul/deskripsi dan waktu pelaksanaan kegiatan, Cara Mengedit yaitu masuk ke mode delete dengan menekan action button di kiri atas, pilih kegiatan yang akan di edit, masukan judul baru/deskripsi/waktu pelaksanaan, klik “Simpan” untuk menyimpan hasil edit, kembali ke halaman utama dengan klik tombol kembali di pojok kanan bawah) apabila dikosongkan akan ada pesan kesalahan, di bagian pojok kanan bawah terdapat icon mengambang untuk menambah kegiatan baru.
- Halaman Tambah Kegiatan Baru: User dapat membuat Kegiatan Baru sesuai keinginan dengan wajib menginputkan Judul, Deskripsi, Durasi (Dapat dipilih antara detik, menit, jam, hari, minggu, bulan, tahun), Prioritas kegiatan (skala 1 – 10), dan Waktu pelaksanaan (Dapat dipilih anatara pagi, siang, malam) apabila setiap form isian tidak diisi akan menampilkan pesan kesalahan, terdapat juga tambahan apabila kegiatan yang dibuat adalah kegiatan penting, user dapat menekan tombol Tandai Sebagai Penting (Opsional), maka kegiatan tersebut akan berada di urutan teratas di seluruh Daftar Kegiatan di halaman Utama.

Tangkapan Layar (Screenshots)

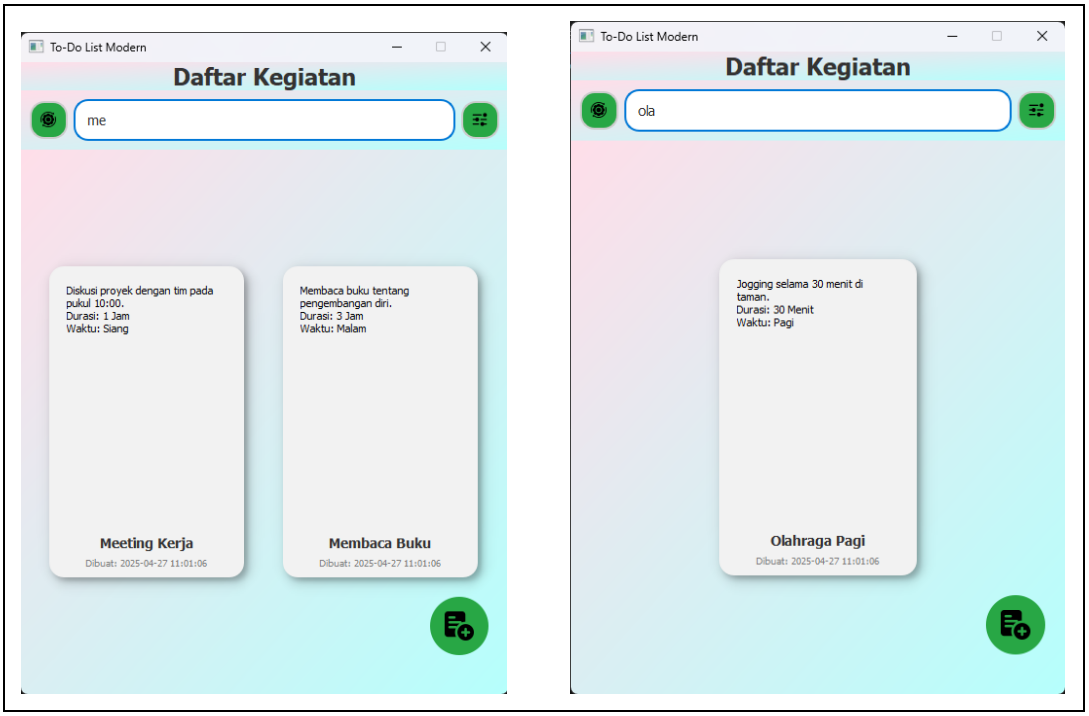
- Tampilan Welcome Page



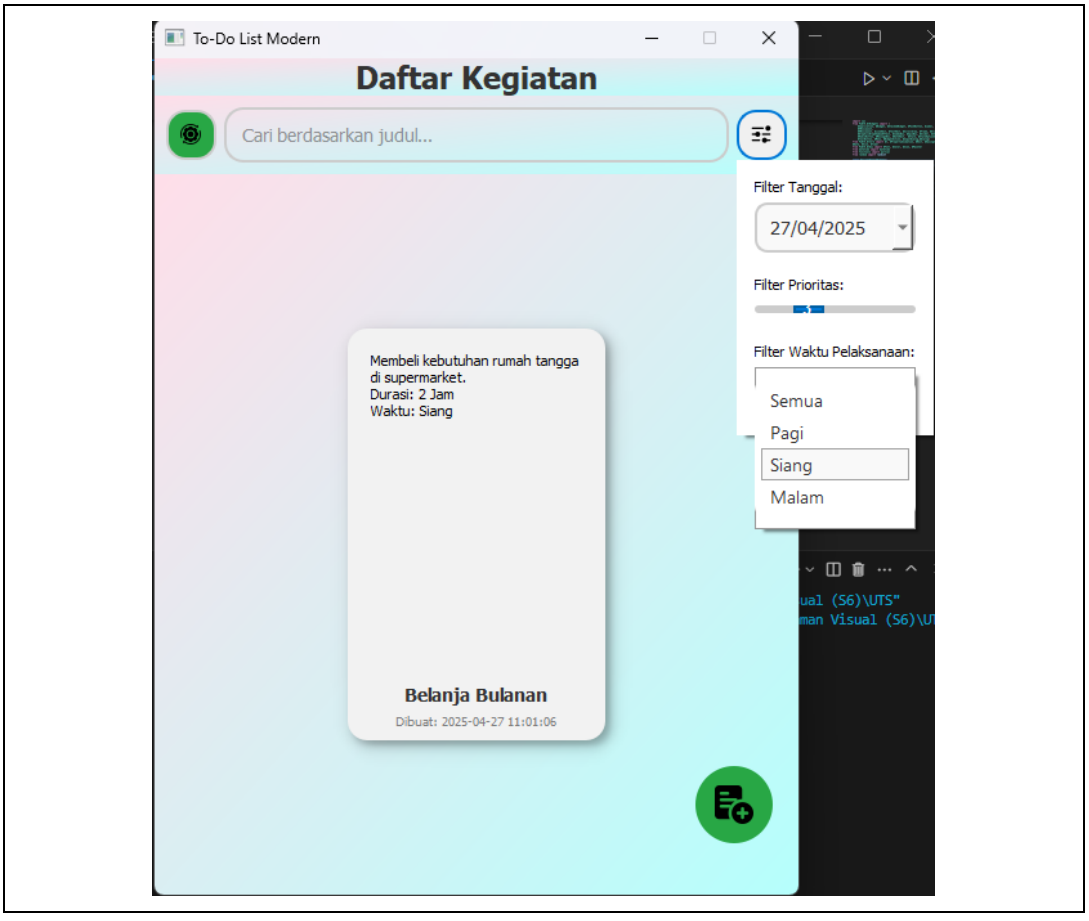
- Tampilan Daftar Kegiatan



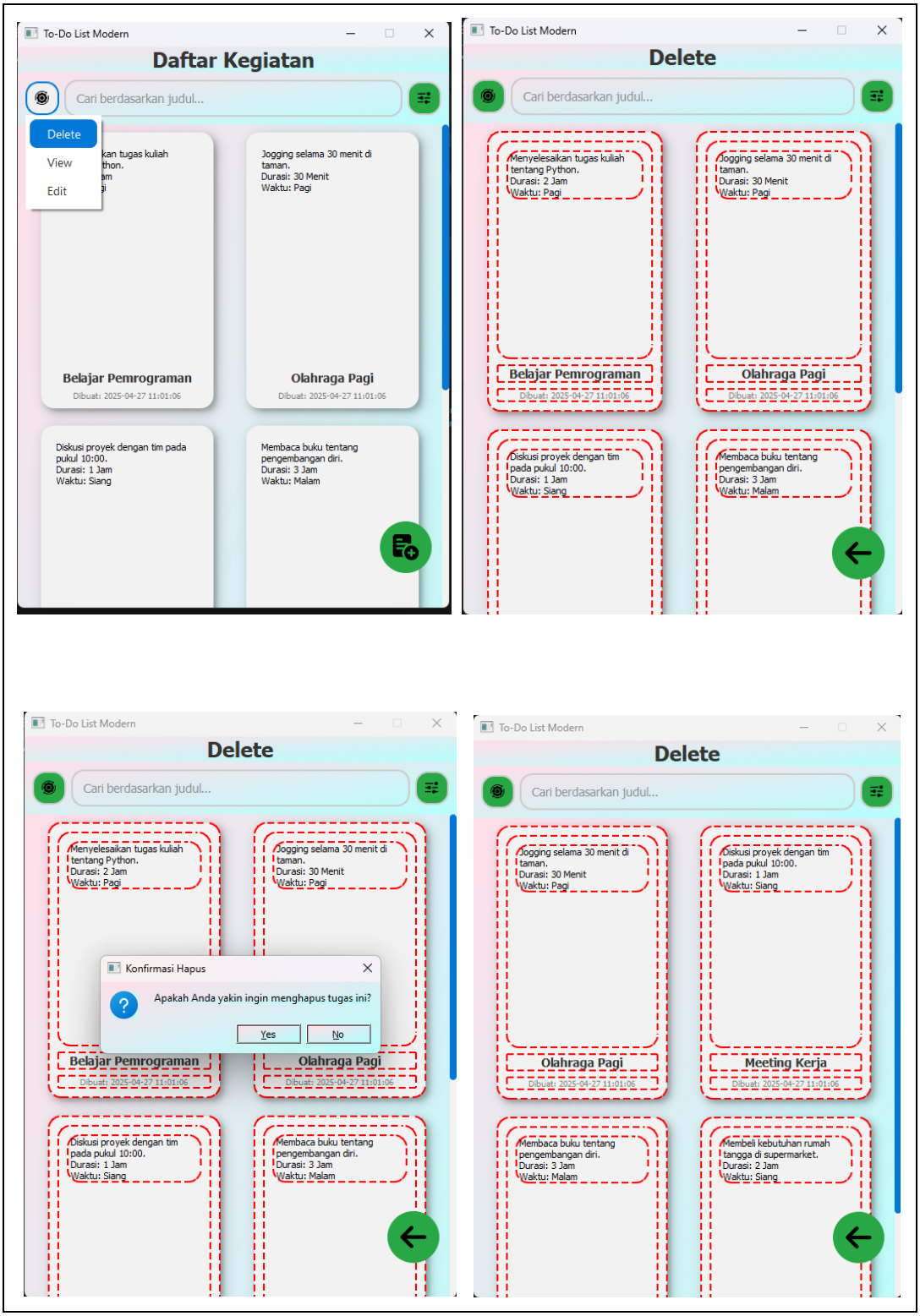
- Tampilan Melakukan Pencarian



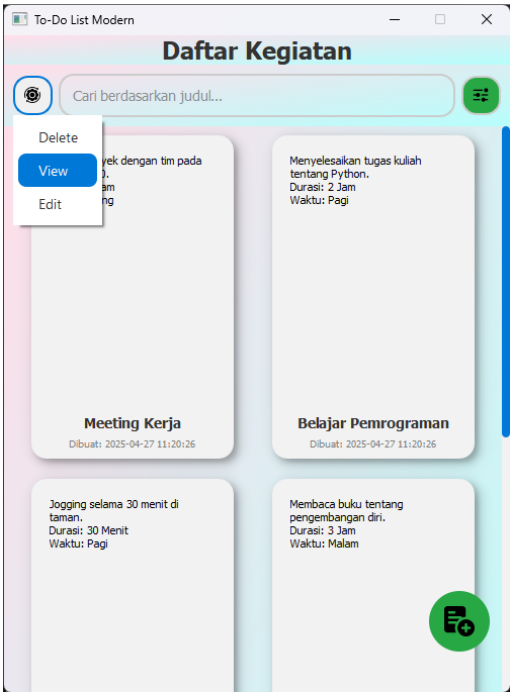
- Tampilan Filter (Tanggal, Prioritas, Waktu Pelaksanaan)

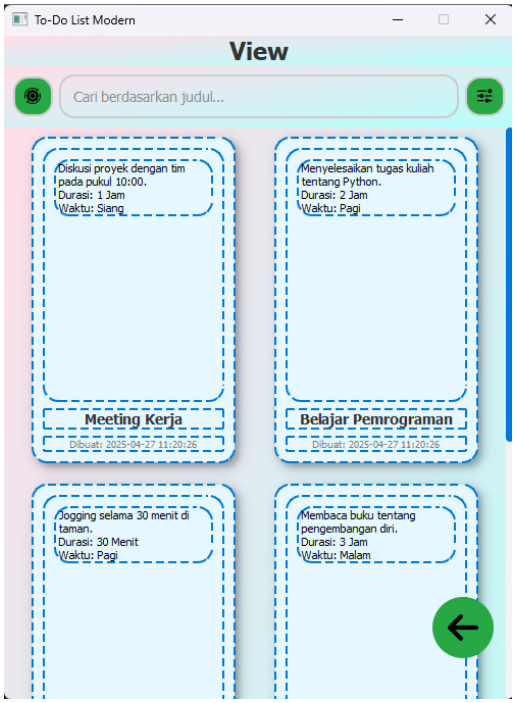


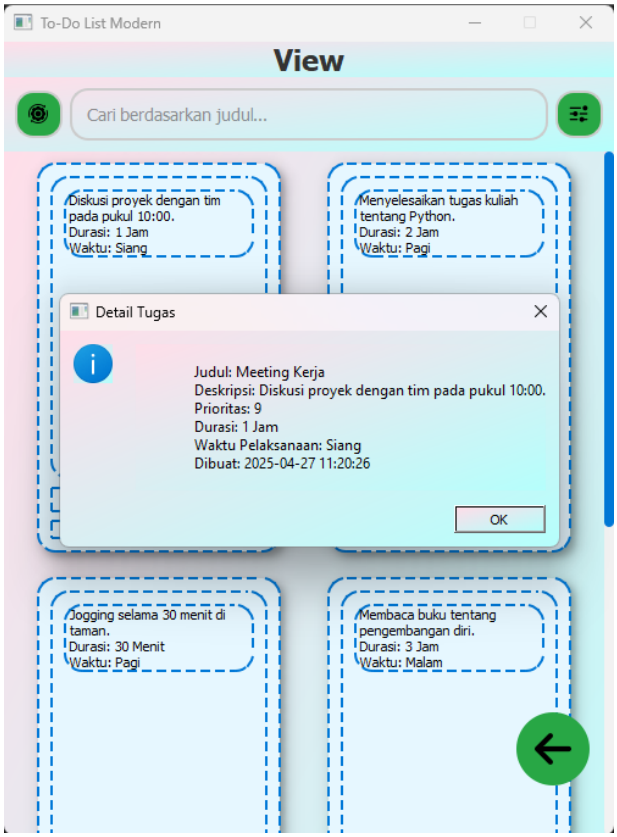
- Tampilan Action Task (Delete)



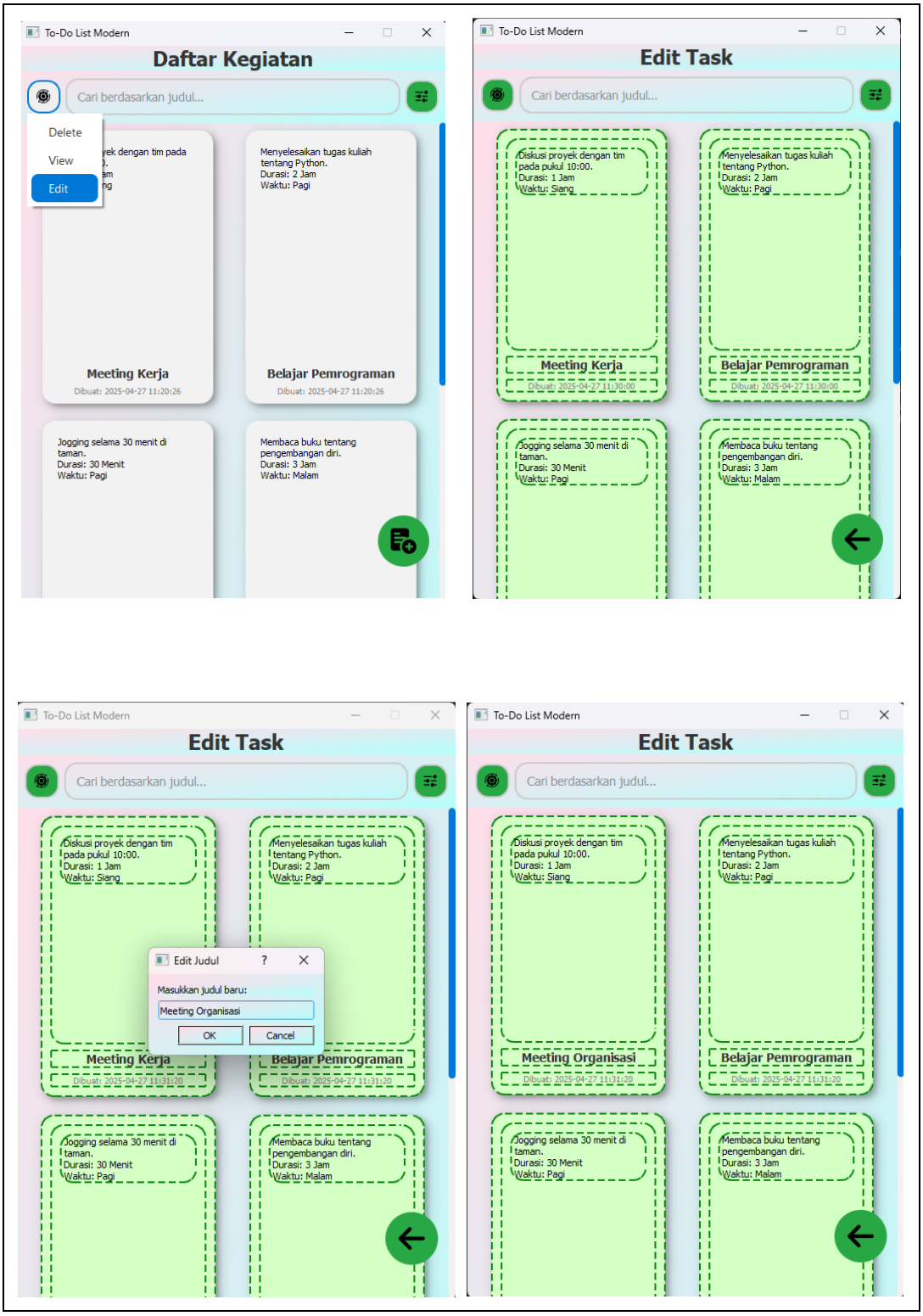
- Tampilan Action Task (View)







- Tampilan Action Task (Edit)



- Tampilan Tambah Kegiatan Baru