**TaskTracker**

**PEMROGRAMAN BERBASIS OBJEK**



**DOSEN PENGAMPUH : Bambang Widodo, S.Kom. M.Ti**

**DISUSUN OLEH :**

**Fahmi Saddam Nurwan Abdullah / 1002230049**

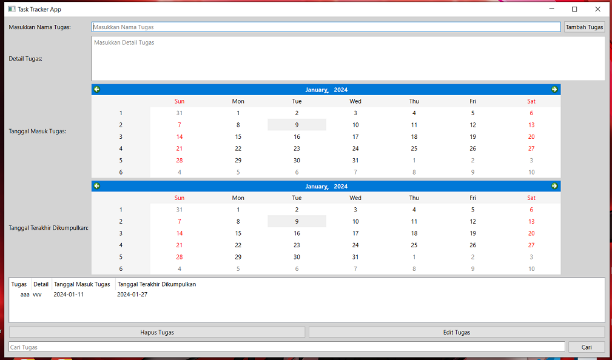
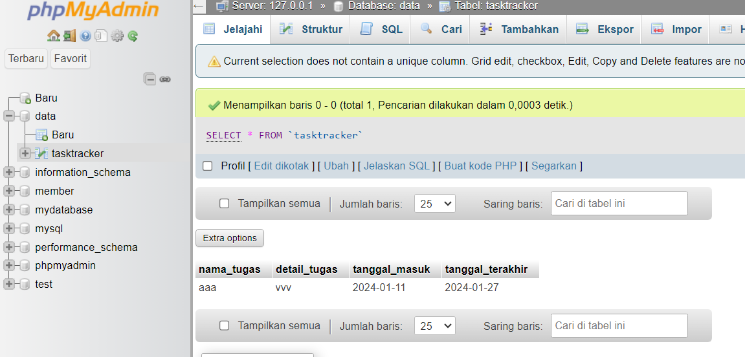
**Nabila Sarwadan / 1001230009**

**Rifa Anindya / 1001230010**

**Nabila Sarwadan / 1001230009**

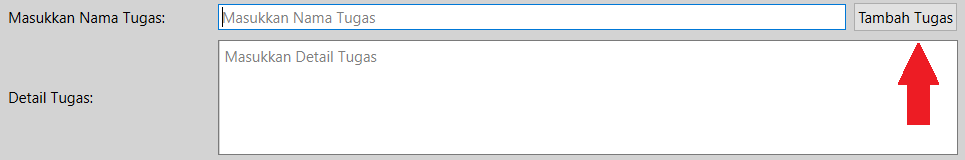
**A NurFaozan / 1003230001**

Dokumentasi untuk aplikasi "Task Tracker App":

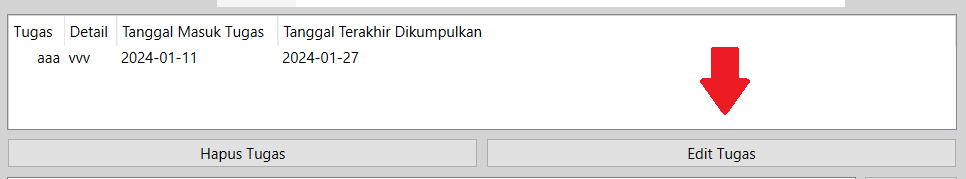
TodoApp adalah aplikasi pelacakan tugas yang memungkinkan pengguna untuk menambahkan, mengedit, dan menghapus tugas. Aplikasi ini dibangun dengan menggunakan PyQt6 untuk antarmuka pengguna dan MySQL Connector untuk koneksi ke database MySQL.

Fitur Utama:

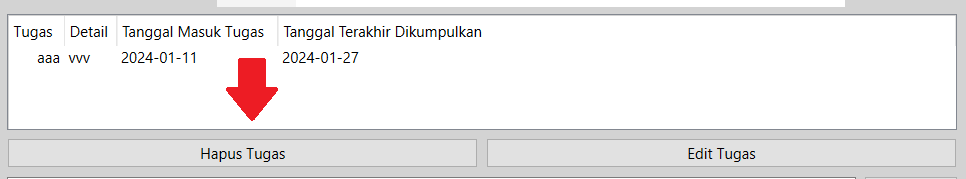
- Menambahkan Tugas Baru: Pengguna dapat menambahkan tugas baru dengan nama, detail, tanggal masuk, dan tanggal terakhir diperbarui.



- Mengedit Tugas yang Ada: Memungkinkan pengguna untuk mengedit informasi tugas yang sudah ada.



- Menghapus Tugas: Pengguna dapat menghapus tugas yang dipilih dari daftar.



- Pencarian Tugas Berdasarkan Nama: Fitur pencarian memungkinkan pengguna untuk mencari tugas berdasarkan nama.



- Pelacakan Tanggal Masuk dan Tanggal Terakhir Diperbarui untuk Setiap Tugas: Aplikasi mencatat tanggal masuk dan tanggal terakhir diperbarui untuk setiap tugas. 

- Pemeriksaan Tugas yang Terlewat: Aplikasi secara otomatis memeriksa dan menghapus tugas yang sudah melewati batas waktu tertentu.

Komponen Utama:

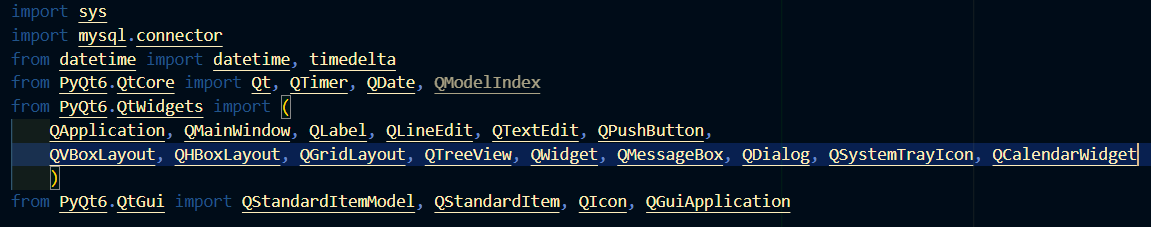
- `QMainWindow`: Merupakan jendela utama aplikasi.

- `QLineEdit` dan `QTextEdit`: Digunakan untuk memasukkan nama, detail tugas, serta menampilkan informasi tugas.

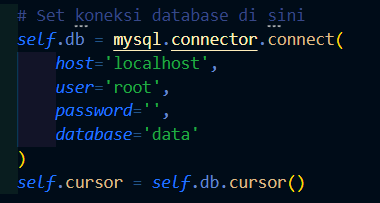
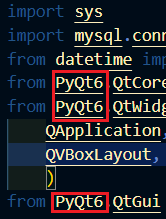
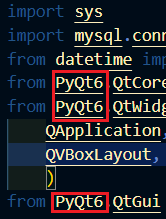
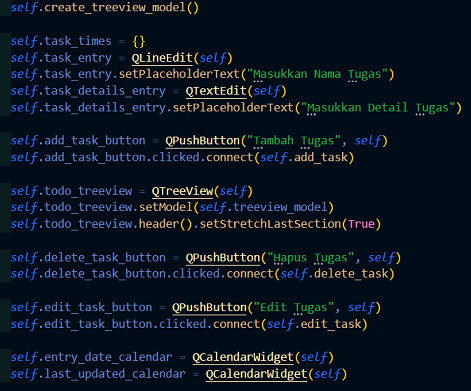
- `QPushButton`: Tombol untuk menambah, mengedit, dan menghapus tugas.

- `QTreeView`: Menampilkan daftar tugas dalam format tabel.

- `QCalendarWidget`: Digunakan untuk memilih tanggal masuk dan tanggal terakhir tugas.

- `QStandardItemModel` dan `QStandardItem`: Model untuk menyimpan data tugas dalam QTreeView.

Struktur Kode:

1. Koneksi Database: Membuat dan mengelola koneksi ke database MySQL.
2. Antarmuka Pengguna: Membangun antarmuka aplikasi menggunakan PyQt6.
3. Manajemen Tugas: Menambahkan, mengedit, dan menghapus tugas dalam database dan tampilan QTreeView.

4. Pemeriksaan Tugas Terlewat: Memeriksa dan menghapus tugas yang sudah melewati batas waktu tertentu dari database dan tampilan QTreeView.

Implementasi Fungsi Utama:

- `set\_app\_size()`: Mengatur ukuran aplikasi berdasarkan ukuran layar.

- `search\_task()`: Mencari tugas berdasarkan teks yang dimasukkan pengguna.

- `search\_task\_auto()`: Mencari tugas secara otomatis saat teks pencarian berubah atau kosong.

- `create\_treeview\_model()`: Membuat model QTreeView untuk menampilkan daftar tugas.

- `add\_treeview\_item()`: Menambahkan item tugas ke dalam QTreeView.

- `populate\_treeview\_from\_db()`: Memuat daftar tugas dari database ke dalam QTreeView.

- `treeview\_item\_clicked()`: Menangani klik item pada QTreeView untuk pengeditan langsung.

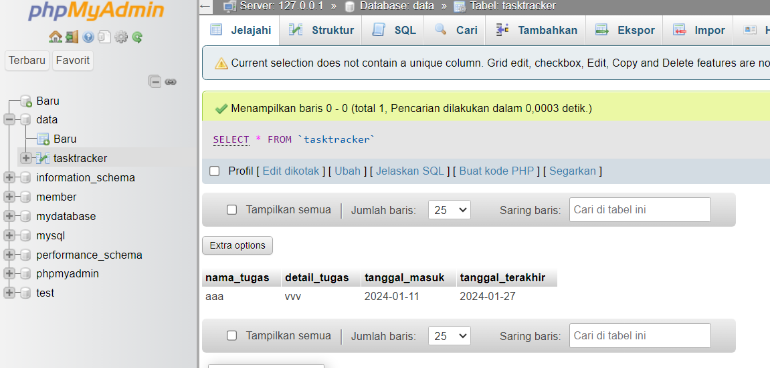
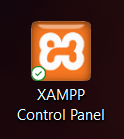
- `check\_overdue\_tasks()`: Memeriksa dan menghapus tugas yang terlewat dari database dan QTreeView.

- `add\_task()`: Menambahkan tugas baru ke dalam database dan QTreeView.

- `delete\_task()`: Menghapus tugas yang dipilih dari database dan QTreeView.

- `edit\_task()`: Mengedit tugas yang dipilih di dalam database dan QTreeView.

Catatan Tambahan:

- Pastikan MySQL telah diinstal dan database telah dibuat sebelum menggunakan aplikasi.

Struktur database yang sesuai untuk kode tersebut dapat dijelaskan sebagai berikut:

SQL :

CREATE TABLE tasktracker (

id INT AUTO\_INCREMENT PRIMARY KEY,

nama\_tugas VARCHAR(255) NOT NULL,

detail\_tugas TEXT,

tanggal\_masuk DATE,

tanggal\_terakhir DATE

);

Penjelasan struktur tabel:

1. `id`: Kolom ini digunakan sebagai kunci utama (primary key) dengan tipe data INTEGER yang akan otomatis bertambah nilainya (AUTO\_INCREMENT).

2. `nama\_tugas`: Kolom ini menyimpan nama tugas dengan tipe data VARCHAR(255) dan tidak boleh NULL.

3. `detail\_tugas`: Kolom ini menyimpan detail tugas dengan tipe data TEXT (bisa menyimpan teks panjang).

4. `tanggal\_masuk`: Kolom ini menyimpan tanggal masuk tugas dengan tipe data DATE.

5. `tanggal\_terakhir`: Kolom ini menyimpan tanggal terakhir diperbarui tugas dengan tipe data DATE.

Pastikan bahwa kolom-kolom yang sesuai dengan data yang Anda simpan dalam aplikasi. Selain itu, sesuaikan juga tipe data kolom dan panjangnya sesuai kebutuhan aplikasi Anda. Setelah membuat tabel dengan struktur tersebut, Anda dapat menggunakan database MySQL Anda untuk mengintegrasikannya dengan kode Python yang telah diberikan.

Dokumentasi ini memberikan gambaran umum tentang struktur, fitur, dan implementasi kunci dari aplikasi TodoApp menggunakan PyQt6 dan MySQL Connector. Dokumen ini dapat diperluas dengan menambahkan detail tentang fungsi tambahan, logika bisnis, atau informasi spesifik lainnya terkait aplikasi.Penggunaan Aplikasi

1. Menambah Tugas:

- Masukkan nama tugas, detail tugas, tanggal masuk, dan tanggal terakhir diperbarui.

- Klik tombol "Tambah Tugas" untuk menyimpan tugas.

2. Edit Tugas:

- Pilih tugas yang ingin diedit.

- Klik tombol "Edit Tugas" untuk membuka jendela edit.

- Ubah informasi tugas yang diinginkan dan klik "Simpan" untuk menyimpan perubahan.

3. Hapus Tugas:

- Pilih tugas yang ingin dihapus.

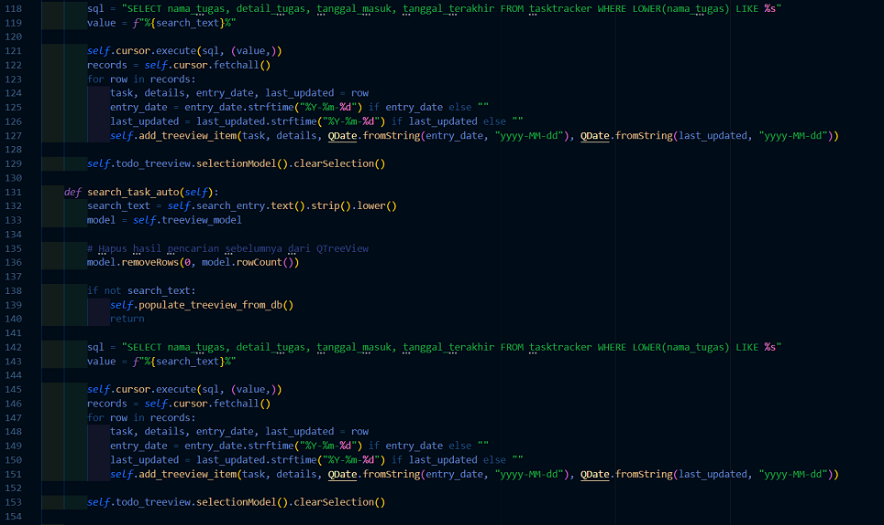
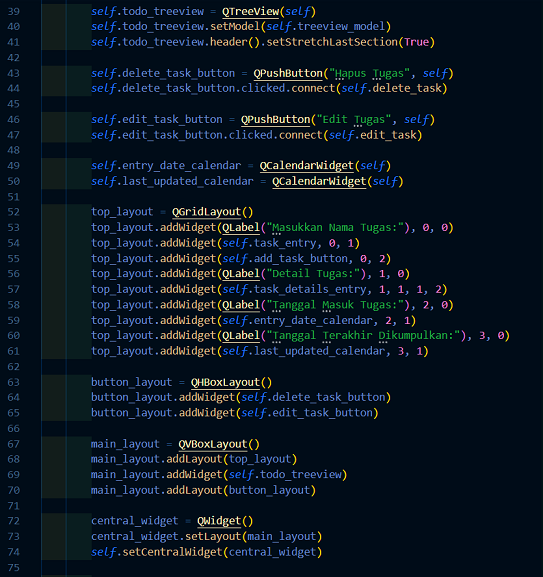
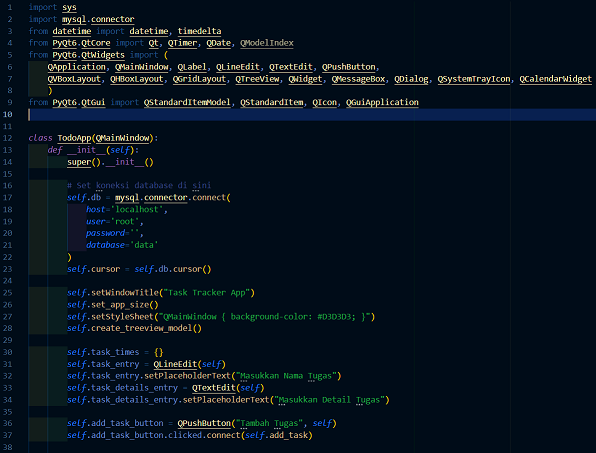
- Klik tombol "Hapus Tugas" untuk menghapus tugas dari daftar.

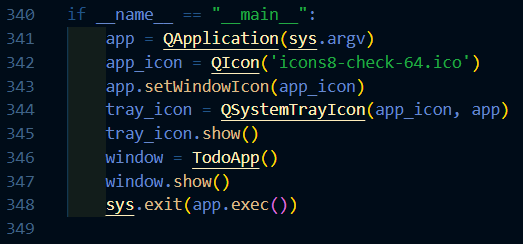
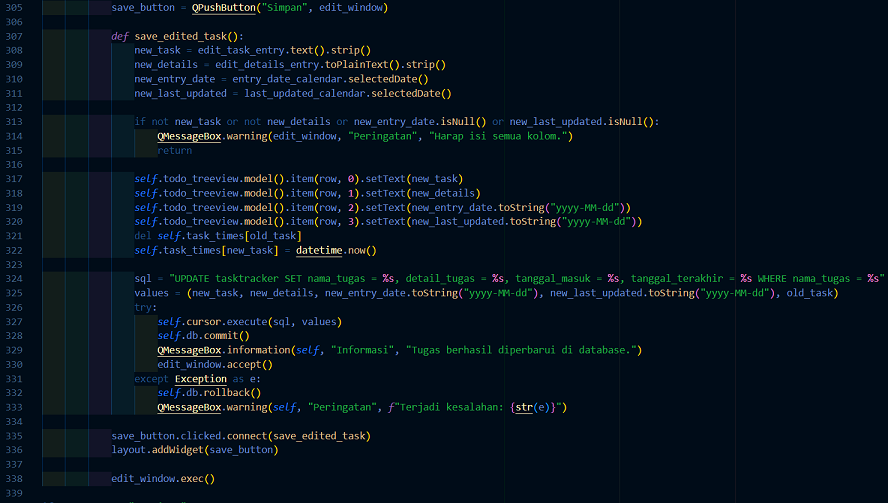
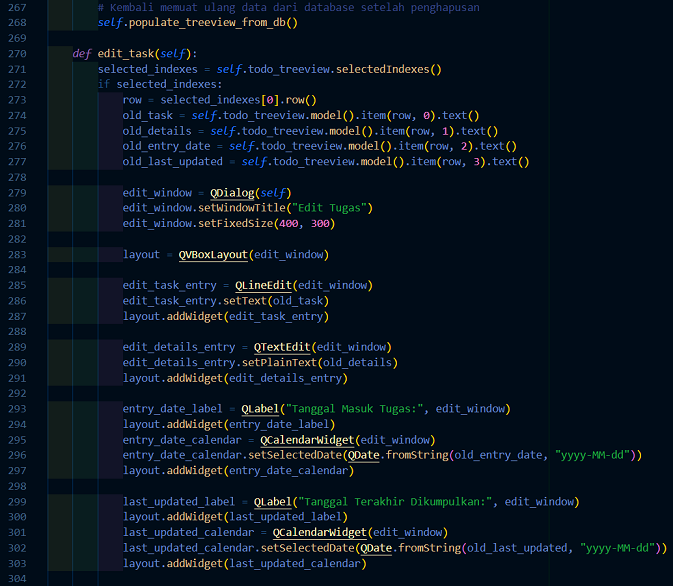
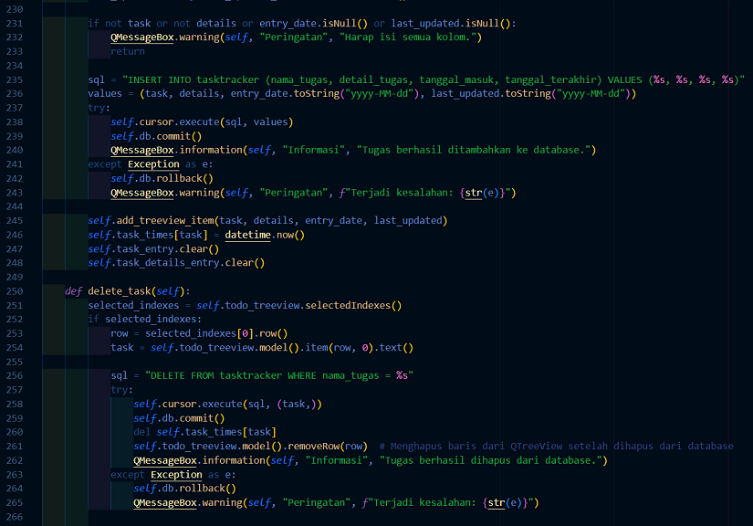
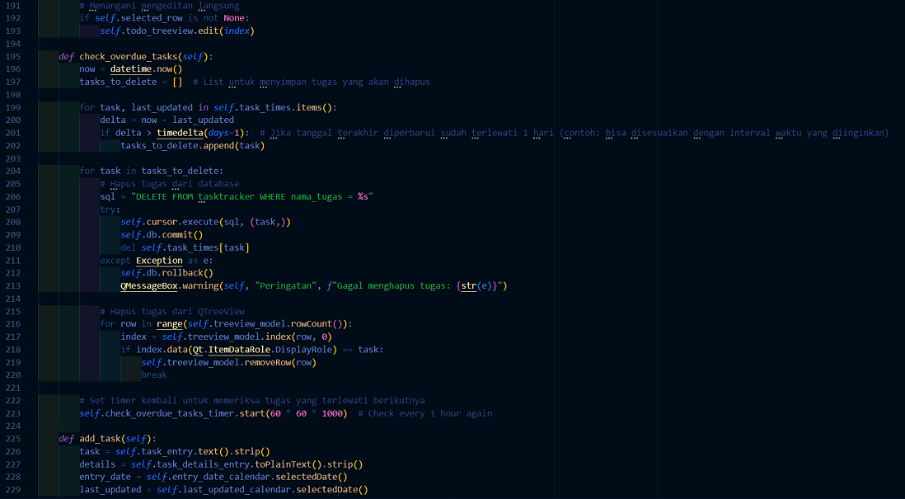
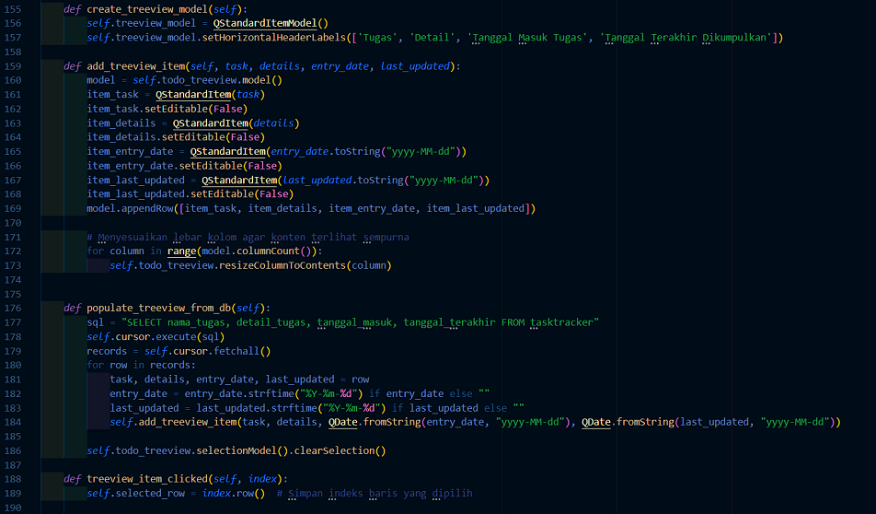
4. Pengingat Tugas yang Melewati Waktu :

- Aplikasi akan memberikan pengingat jika terdapat tugas yang belum selesai atau diedit dalam waktu 24 jam terakhir. Akan Tetapi fitur ini masih dalam tahap pengembangan yang lebih lanjut.

Kesimpulan

Aplikasi "Task Tracker App" adalah aplikasi sederhana untuk melacak dan mengelola tugas-tugas yang harus dilakukan. Dibuat menggunakan PyQt6 dan menghubungkan ke database MySQL untuk menyimpan data tugas. Aplikasi ini memiliki fitur dasar untuk menambah, mengedit, dan menghapus tugas, serta memberikan pengingat untuk tugas-tugas yang melewati batas waktu yang ditentukan.



Input:

1. Nama Tugas dan Detail Tugas:

- Pengguna diminta untuk memasukkan nama tugas dan detail tugas ke dalam aplikasi menggunakan `QLineEdit` untuk nama tugas dan `QTextEdit` untuk detail tugas.

2. Tanggal Masuk dan Tanggal Terakhir Diperbarui:

- Pengguna memilih tanggal masuk dan tanggal terakhir diperbarui untuk setiap tugas menggunakan `QCalendarWidget`.

3. Pencarian Tugas:

- Terdapat fitur pencarian yang memungkinkan pengguna memasukkan teks pencarian ke dalam `QLineEdit` untuk mencari tugas berdasarkan nama.

Proses:

1. Menambah Tugas Baru:

- Pengguna memasukkan informasi tugas (nama, detail, tanggal masuk, dan tanggal terakhir) ke dalam formulir input.

- Setelah pengguna menekan tombol "Tambah Tugas", informasi tersebut akan dimasukkan ke dalam database MySQL dengan bantuan fungsi `add\_task()` dalam aplikasi.

- Informasi tugas juga ditambahkan ke dalam QTreeView yang menampilkan daftar tugas.

2. Mengedit Tugas yang Ada:

- Pengguna dapat mengklik tugas yang ada di QTreeView untuk mengedit langsung.

- Ketika pengguna mengklik tombol "Edit Tugas", aplikasi akan membuka jendela dialog yang memungkinkan pengguna mengubah informasi tugas (nama, detail, tanggal masuk, dan tanggal terakhir).

- Setelah pengguna menekan tombol "Simpan" dalam jendela dialog, informasi tugas akan diperbarui di database dan QTreeView menggunakan fungsi `edit\_task()`.

3. Menghapus Tugas:

- Pengguna memilih tugas yang ingin dihapus dari QTreeView.

- Setelah pengguna menekan tombol "Hapus Tugas", aplikasi akan menghapus tugas yang dipilih dari database dan QTreeView menggunakan fungsi `delete\_task()`.

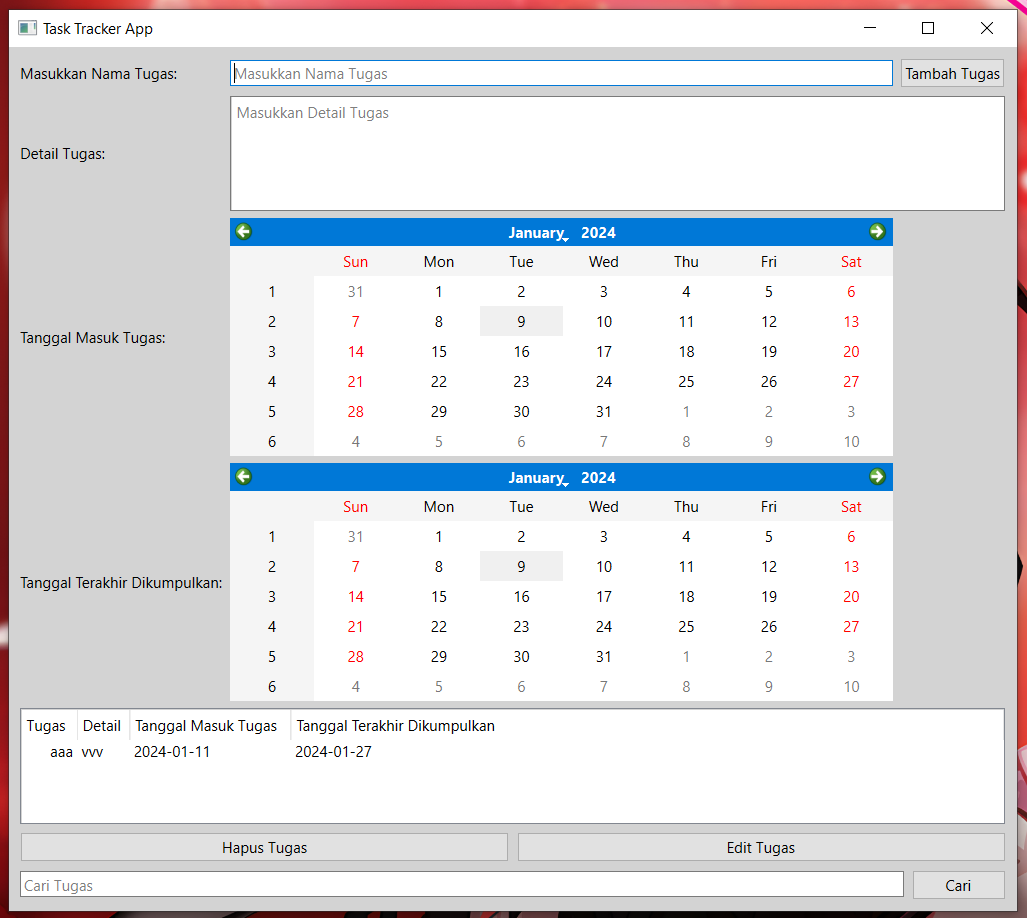
4. Pencarian Tugas:

- Saat pengguna memasukkan teks pencarian ke dalam `QLineEdit` untuk mencari tugas berdasarkan nama, aplikasi akan melakukan pencarian dalam database menggunakan fungsi `search\_task()` atau `search\_task\_auto()`.

5. Pemeriksaan Tugas Terlewat:

- Aplikasi secara otomatis memeriksa tugas yang sudah melewati batas waktu tertentu (contohnya: setiap satu jam).

- Jika ada tugas yang sudah melewati batas waktu, tugas tersebut akan dihapus dari database dan QTreeView menggunakan fungsi `check\_overdue\_tasks()`.



Output:

1. Daftar Tugas:

- Daftar tugas ditampilkan dalam format tabel menggunakan `QTreeView` di antarmuka pengguna.

- Setiap tugas ditampilkan dengan kolom-kolom yang mencakup nama tugas, detail tugas, tanggal masuk, dan tanggal terakhir diperbarui.

2. Informasi Tambahan:

- Pesan informasi dan peringatan ditampilkan menggunakan `QMessageBox` ketika tugas berhasil ditambahkan, dihapus, atau diperbarui, atau ketika ada kesalahan dalam proses tersebut.

- Jendela dialog akan muncul saat pengguna ingin mengedit tugas untuk memungkinkan pengeditan informasi tugas yang lebih detail.

3. Pencarian Hasil:

- Hasil pencarian tugas berdasarkan nama akan ditampilkan dalam QTreeView setelah pengguna melakukan pencarian menggunakan fitur pencarian.

4. Pemberitahuan Sistem Tray (Tray Icon):

- Aplikasi menampilkan ikon di sistem tray untuk memberi tahu pengguna bahwa aplikasi sedang berjalan.

5. Interaksi Pengguna:

- Pengguna dapat berinteraksi dengan aplikasi melalui input keyboard, mouse, dan tombol yang disediakan di antarmuka aplikasi.

Dalam aplikasi TodoApp, input melibatkan pengisian informasi tugas, prosesnya termasuk manajemen data tugas dan interaksi pengguna, dan outputnya adalah tampilan daftar tugas yang dikelola dan informasi terkait dari proses yang dilakukan oleh pengguna pada antarmuka aplikasi.