LAPORAN
TUGAS AHIR

Mata Kuliah Pemrograman Berorientasi Objek

Disusun Oleh:

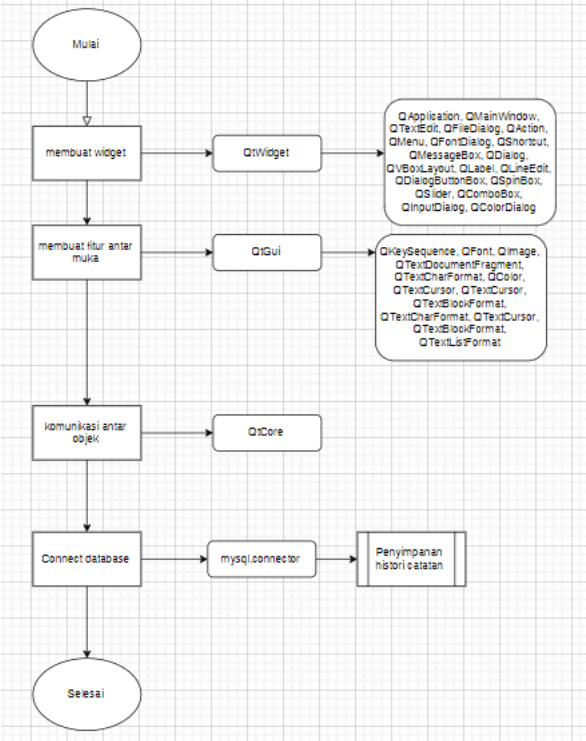
1. Syailendra Julian W
2. M. Hafid

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NUSANTARA PGRI KEDIRI

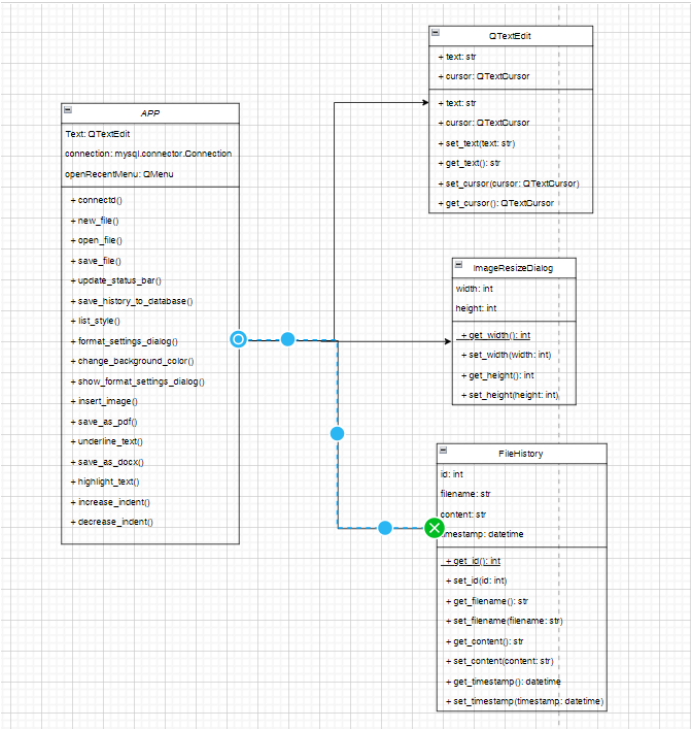
TAHUN 2023

Daftar Isi

Flowchart System



Class Diagram



Modul-modul

```
import sys
import mysql.connector
from PyQt5.QtWidgets import QApplication, QMainWindow, QTextEdit,
QFileDialog, QAction, QMenu, QFontDialog, QShortcut, QMessageBox,
QDialog, QVBoxLayout, QLabel, QLineEdit, QDialogButtonBox, QSpinBox,
QSlider, QComboBox, QInputDialog, QColorDialog
from PyQt5.QtGui import QKeySequence, QFont, QImage,
QTextDocumentFragment, QTextCharFormat, QColor, QTextCursor,
QTextCursor, QTextBlockFormat, QTextCharFormat, QTextCursor,
QTextBlockFormat, QTextListFormat
from PyQt5.QtCore import Qt
from reportlab.pdfgen import canvas
import docx as Document
```

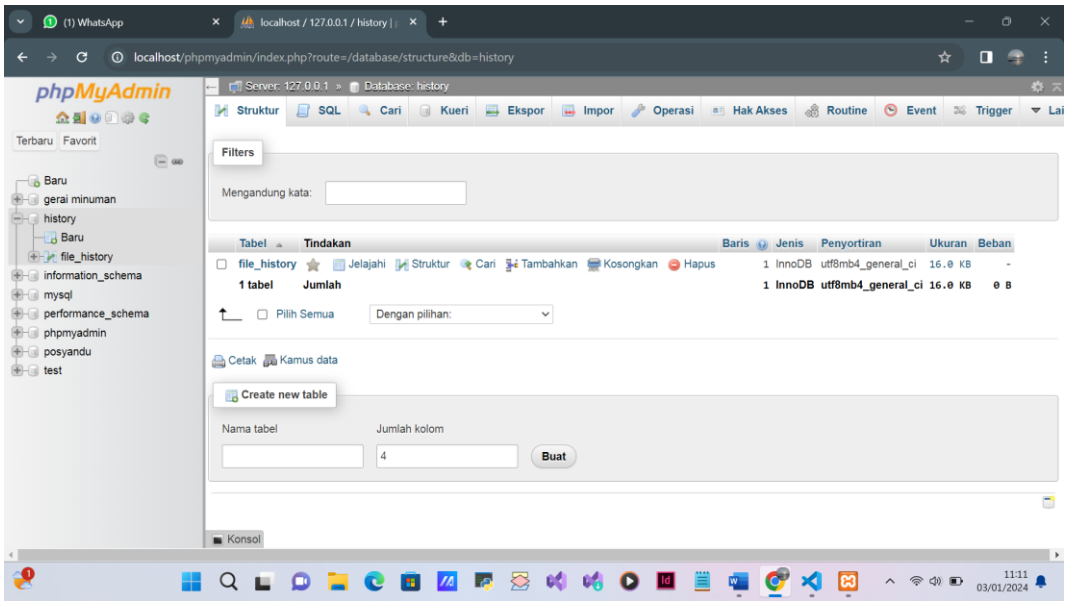
- Sys
  Modul ini sudah ada pada saat python diinstall dan fungsinya untuk akses file local
- Mysql.connector
  Bagian ini digunakan untuk menghubungkan source code dengan basis data mysql
- Pyqt5
  - QtWidgets
    Digunakan untuk membuat jendela pada aplikasi GUI jadi semua turunan modulnya berfungsi untuk tampilan luar aplikasi
  - QtGui
    Dimodul ini disimpan komponen-komponen interface dengan pengguna
  - QtCore
    Inti dari non-GUI, digunakan oleh modul lain
- Reportlab
  Bagian ini untuk mencetak file bertipe pdf maupun grafis
- Python-docx
  Digunakan untuk membaca, membuat dan mengupdate file ms.word

Fitur

- Mysql.connect
  Aplikasi/GUI ini menggunakan mysql untuk penyimpanan, walau sebenarnya lebih baik menggunakan path yang nantinya disimpan langsung di local

```
def connectd(self):
        self.connection = mysql.connector.connect(
            host="localhost",
            user="root",
            password="",
            database="history"
        )
        self.cursor = self.connection.cursor()

        create_table_query = """
        CREATE TABLE IF NOT EXISTS file_history (
            id INT AUTO_INCREMENT PRIMARY KEY,
            filename VARCHAR(255),
            content TEXT,
            timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
        )
        """
        self.cursor.execute(create_table_query)
        self.connection.commit()
```



- Change background color
  Notepad ini memiliki fitur mengganti warna background sesuai yang diinginkan pengguna

```
def change_background_color(self):
     color =
QColorDialog.getColor(self.palette().color(self.backgroundRole()),
self, "Select Background Color")
     if color.isValid():
        self.set_background_color(color)

    def set_background_color(self, color):
     palette = self.palette()
     palette.setColor(self.backgroundRole(), color)
     self.setPalette(palette)
```

- Save doc & pdf
  Fitur ini digunakan untuk merubah jenis file .text ke pdf/docx

```
def save_as_pdf(self):
        filename, _ = QFileDialog.getSaveFileName(self, "Save as
PDF", "", "PDF Files (*.pdf);;All Files (*)")
        if filename:
            try:
                pdf_canvas = canvas.Canvas(filename)
                pdf_canvas.setFont("Helvetica", 12)  # You can
change the font and size
                pdf_canvas.drawString(72, 800,
self.text.toPlainText())
                pdf_canvas.save()
            except Exception as e:
                QMessageBox.critical(self, "Error", f"An error
occurred while saving as PDF:\n{str(e)}")

    def save_as_docx(self):
        filename, _ = QFileDialog.getSaveFileName(self, "Save as
DOCX", "", "Word Documents (*.docx);;All Files (*)")
        if filename:
            try:
                if not filename.lower().endswith(".docx"):
                    filename += ".docx"

                document = Document()
                document.add_paragraph(self.text.toPlainText())
                document.save(filename)
            except Exception as e:
                QMessageBox.critical(self, "Error", f"An error
occurred while saving as DOCX:\n{str(e)}")
```

- Font
  Terinspirasi dari ms.word, fitur ini digunakan untuk mengganti font penulisan dalam notepad

```
def add_font_actions_to_menu(self):
        font_list = ["Arial", "Times New Roman", "Courier New",
"Verdana", "Serif", "Sans Serif", "Monospace", "Cursive", "Fantasy"]
        for font_name in font_list:
            action = QAction(font_name, self)
            action.triggered.connect(lambda _, font_name=font_name:
self.change_font_for_selected_word(font_name))
            self.context_menu.addAction(action)

    def change_font_for_selected_word(self, font_name):
        cursor = self.text.textCursor()
        selected_text = cursor.selectedText()

        if selected_text:
            char_format = QTextCharFormat()
            char_format.setFont(QFont(font_name, 12))
            cursor.mergeCharFormat(char_format)
```

- Cut,copy,paste, select all
  Untuk yang satu ini adalah fitur dasar yang Sebagian besar aplikasi untuk menulis memilikinya

```
def cut(self):
        self.text.cut()

    def copy(self):
        self.text.copy()

    def paste(self):
        clipboard = QApplication.clipboard()
        mime_data = clipboard.mimeData()

        if mime_data.hasText():
            text_fragment =
QTextDocumentFragment.fromHtml(mime_data.text())
            cursor = self.text.textCursor()
            cursor.insertFragment(text_fragment)
```

- Insert image
  Notepad ini juga bisa digunakan untuk menginsertkan gambar

```
def insert_image(self):
        options = QFileDialog.Options()
        options |= QFileDialog.ReadOnly

        file_name, _ = QFileDialog.getOpenFileName(self, "Open Image
File", "", "Images (*.png *.jpg *.bmp *.gif);;All Files (*)",
options=options)

        if file_name:
            try:
                dialog = ImageResizeDialog(self)
                if dialog.exec_() == QDialog.Accepted:
                    width, height = dialog.get_size()
                    image = QImage(file_name)
                    if width > 0 and height > 0:
                        image = image.scaled(width, height,
Qt.KeepAspectRatio)
                    cursor = self.text.textCursor()
                    cursor.insertImage(image)
            except Exception as e:
                QMessageBox.critical(self, "Error", f"An error
occurred while inserting the image:\n{str(e)}")
```

- Spacing
  Fitur ini berfungsi untuk mengatur jarak antar baris dalam paragraf

```
def show_format_settings_dialog(self):
     dialog = FormatSettingsDialog(self)
     result = dialog.exec_()
     if result == QDialog.Accepted:
        line_spacing, paragraph_spacing, background_color =
dialog.get_format_settings()
        self.set_text_formatting(line_spacing, paragraph_spacing)
        self.set_background_color(background_color)

def set_text_formatting(self, line_spacing, paragraph_spacing):
        cursor = self.text.textCursor()
        block_format = cursor.blockFormat()

        block_format.setLineHeight(line_spacing,
QTextBlockFormat.ProportionalHeight)
        block_format.setBottomMargin(paragraph_spacing)
        block_format.setTopMargin(paragraph_spacing)
```

```
        cursor.setBlockFormat(block_format)
        self.text.setTextCursor(cursor)

        char_format = cursor.charFormat()
        char_format.setFontPointSize('font_size')
        cursor.setCharFormat(char_format)

        self.text.setTextCursor(cursor)
```

- Symbol & emoji
  Fitur tambahan untuk mempercantik karya tulis

```
def show_symbol_dialog(self):
        symbol_list = ["©", "®", "™", "€", "£", "¥", "§"]
        symbol, ok = QInputDialog.getItem(self, "Insert Symbol",
"Select a symbol:", symbol_list, 0, False)
        if ok and symbol:
            cursor = self.text.textCursor()
            cursor.insertText(symbol)

    def show_emoji_dialog(self):
        emoji_list = ["😊", "😍", "👍", "🎉", "💖", "🚀"]
        emoji, ok = QInputDialog.getItem(self, "Insert Emoji",
"Select an emoji:", emoji_list, 0, False)
        if ok and emoji:
            cursor = self.text.textCursor()
            cursor.insertText(emoji)
```

- Count (word, space)
  Fitur ini digunakan untuk menghitung jumlah kata dan spasi

```
def update_status_bar(self):
        text = self.text.toPlainText()

        word_count = len(text.split())
        space_count = text.count(' ')

        status_text = f"Words: {word_count}, Spaces: {space_count}"
        self.statusBar.showMessage(status_text)
```

- Bullets & Numbering
  Biasanya digunakan untuk daftar nama benda atau lainnya

```
def set_list_style(self, list_style):
     cursor = self.text.textCursor()
     block_format = cursor.blockFormat()

     list_format = QTextListFormat()
     list_format.setStyle(list_style)

     cursor.mergeBlockFormat(block_format)
     cursor.createList(list_format)
```

- Alignment
  Digunakan untuk mengatur letak paragraf

```
def set_alignment(self, alignment):
        cursor = self.text.textCursor()
        block_format = cursor.blockFormat()
        block_format.setAlignment(alignment)
        cursor.setBlockFormat(block_format)
```

- Increase & decrease indent
  Fungsinya seperti tombol tab pada keyboard

```
def increase_indent(self):
        cursor = self.text.textCursor()
        cursor.insertText("\t")

    def decrease_indent(self):
        cursor = self.text.textCursor()
        cursor.movePosition(QTextCursor.StartOfBlock)
        cursor.movePosition(QTextCursor.NextCharacter,
QTextCursor.KeepAnchor, 1)
        selected_text = cursor.selectedText()
        if selected_text == "\t":
            cursor.removeSelectedText()
```

Daftar Pustaka

doc.qt.io. (n.d.). QWidget Class.

docs.reportlab.com. (n.d.). QtGui.

tinkq.blogspot.com. (n.d.). Fungsi Decrease indent dan Increase indent pada Microsoft Word.