

LAPORAN TUGAS AKHIR
MATA KULIAH PEMROGAMAN BERORIENTASI OBJEK



Di susun Oleh:

1. Ardan Putra Ramadhan 2113020213
2. Onkky Sandika Mahendra 2113020234
3. Satria Bagus Setya Budhi 2113020235

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NUSANTARA PGRI KEDIRI
TAHUN 2023

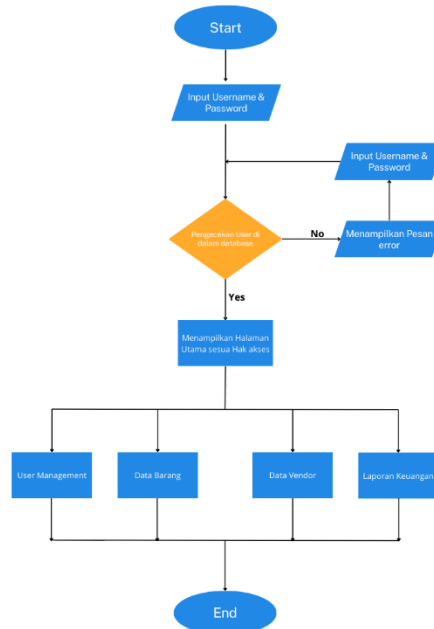
Daftar Isi

BAB I.....	3
Flowchart	3
Flwochart Bagian Admin	3
Flowchart Bagian Kasir	3
Kelas Diagram.....	4
Modul - Modul	5
1. PyQt5 Modules Import	5
2. Impor Widget PyQt5 Tambahan:	5
3. Impor Modul Koneksi Database:	5
4. Sys.....	5
Fitur Connect Database PhpMyAdmin.....	6
1. Import Modul MySQL Connector:	6
2. Kelas conndb:	6
4. Metode queryExecute:	6
Fitur – Fitur.....	7
1. Inisialisasi Kelas MainWindow:	7
5. Metode Login:.....	8
6. Metode-metode untuk Menangani Menu Aplikasi:.....	8
7. Instansiasi Objek Aplikasi dan Jendela:.....	8

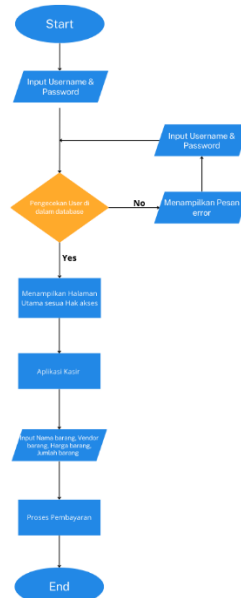
BAB I.

Flowchart

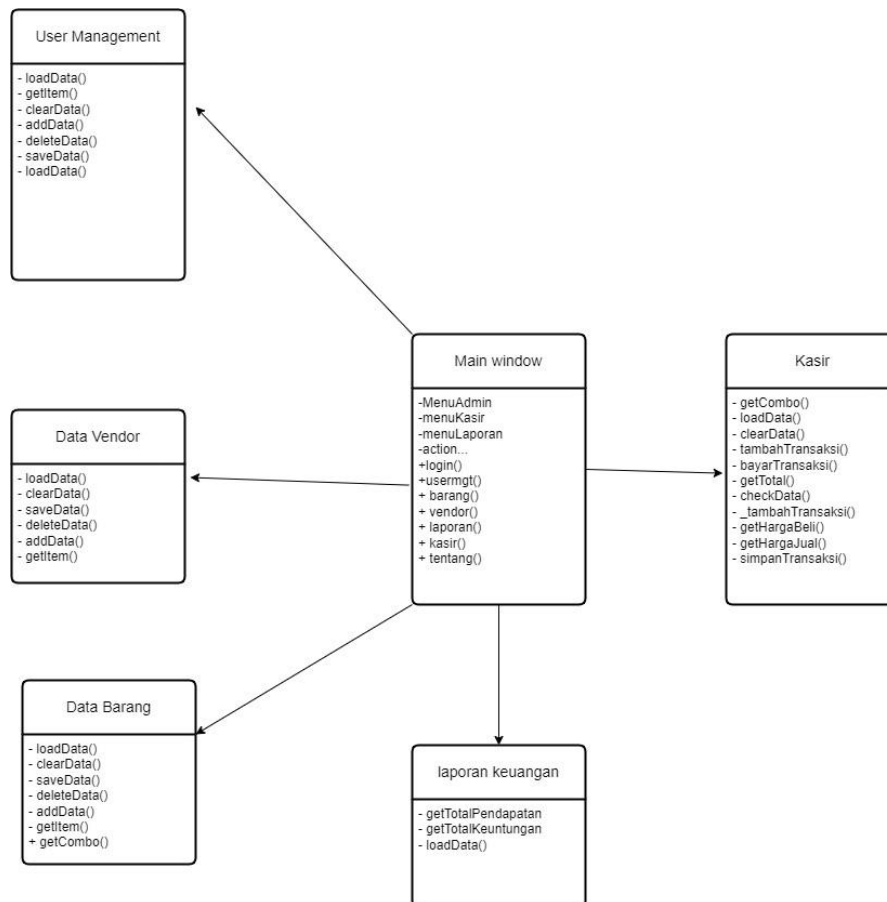
Flwochart Bagian Admin



Flowchart Bagian Kasir



Kelas Diagram



Modul - Modul

```
from PyQt5 import QtCore, QtGui
from PyQt5 import QtWidgets, uic
from PyQt5.QtWidgets import QMessageBox, QDesktopWidget

import conndb

import sys
from PyQt5 import QtWidgets, uic
import user_management as um
import login
import tentang
import barang as b
import vendor as v
import kasir as ks
import laporan as l
```

1. PyQt5 Modules Import

- A. QtCore: Menyediakan fungsionalitas inti yang tidak berhubungan dengan GUI.
- B. QtGui: Berisi fungsionalitas grafis.
- C. QtWidgets: Termasuk serangkaian elemen antarmuka pengguna untuk membuat aplikasi gaya desktop.
- D. uic: Merupakan singkatan dari User Interface Compiler, digunakan untuk memuat file UI yang dibuat dengan alat Qt Designer.

2. Impor Widget PyQt5 Tambahan:

- A. QMessageBox: Dialog untuk memberi tahu pengguna atau meminta konfirmasi.
- B. QDesktopWidget: Memberikan informasi tentang desktop pengguna, termasuk ukuran layar dan geometri.

3. Impor Modul Koneksi Database:

conndb. Modul ini berisi fungsi atau kelas terkait dengan koneksi database, tetapi implementasi sebenarnya tidak diberikan dalam potongan kode.

4. Sys :

Modul ini sudah ada pada saat python diinstall dan fungsinya untuk akses file local

Fitur Connect Database PhpMyAdmin

```
import mysql.connector

class conndb:
    def __init__(self):
        pass

    def queryResult(self, strsql):
        cnx = mysql.connector.connect(user='root', password='', host='127.0.0.1', database='db_toko')
        conn = cnx.cursor()
        conn.execute(strsql)
        result = conn.fetchall()
        return result
        pass

    def queryExecute(self, strsql):
        cnx = mysql.connector.connect(user='root', password='', host='127.0.0.1', database='db_toko')
        conn = cnx.cursor()
        conn.execute(strsql)
        cnx.commit()
        pass
```

1. Import Modul MySQL Connector:

Modul `mysql.connector` diimpor untuk memungkinkan interaksi dengan database MySQL.

2. Kelas `conndb`:

Kelas `conndb` didefinisikan dengan metode konstruktor (`__init__`) yang kosong.

3. Metode `queryResult`:

- A. Metode `queryResult` digunakan untuk menjalankan sebuah query SQL yang mengembalikan hasil (SELECT).
- B. Metode membuka koneksi ke database MySQL dengan menggunakan parameter koneksi (user, password, host, dan nama database).
- C. Mengeksekusi query SQL (`strsql`) menggunakan objek cursor.
- D. Mengambil hasil eksekusi dengan `fetchall()` dan mengembalikan hasil tersebut.
- E. Setelah selesai, koneksi ditutup.

4. Metode `queryExecute`:

- A. Metode `queryExecute` digunakan untuk menjalankan sebuah query SQL yang tidak mengembalikan hasil (INSERT, UPDATE, DELETE).
- B. Metode membuka koneksi ke database MySQL dengan menggunakan parameter koneksi (user, password, host, dan nama database).
- C. Mengeksekusi query SQL (`strsql`) menggunakan objek cursor.
- D. Melakukan commit untuk menyimpan perubahan ke dalam database.
- E. Setelah selesai, koneksi ditutup.

Kelas `conndb` ini menyediakan dua metode utama: `queryResult` untuk menjalankan query SELECT dan mengembalikan hasilnya, serta `queryExecute` untuk menjalankan query lainnya yang tidak mengembalikan hasil. Namun, ada beberapa aspek yang perlu diperhatikan, seperti pengelolaan koneksi yang tidak optimal (karena setiap metode membuka koneksi baru).

Sebaiknya, koneksi dapat dikelola secara lebih efisien, misalnya, dengan menggunakan manajer konteks (with) atau mempertimbangkan pola desain lain untuk manajemen koneksi database.

Fitur – Fitur

```
class MainWindow(QMainWindow):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        uic.loadUi("main.ui", self)
        self.setWindowTitle("Andika Motor")
        self.logout()
        self.actionUser_Management.triggered.connect(self.usermgt)
        self.actionData_Barang.triggered.connect(self.barang)
        self.actionData_Vendor.triggered.connect(self.vendor)
        self.actionLogin.triggered.connect(self.login)
        self.actionLogout.triggered.connect(self.logout)
        self.actionAplikasi.triggered.connect(self.tentang)
        self.actionAplikasi_Kasir.triggered.connect(self.kasir)
        self.actionLaporan_Keuangan.triggered.connect(self.laporan)

    def logout(self):
        self.menuAdmin.setEnabled(False)
        self.menuKasir.setEnabled(False)
        self.menuLaporan.setEnabled(False)

    def login(self):
        self.lg = login.login()
        self.lg.exec()
        result = self.lg.result
        self.logout()
        if result == "ad":
            self.menuAdmin.setEnabled(True)
            self.menuLaporan.setEnabled(True)
        if result == "ks":
            self.menuKasir.setEnabled(True)
```

1. Inisialisasi Kelas MainWindow:

Kelas MainWindow diinisialisasi dan merupakan turunan dari kelas QtWidgets.QMainWindow.

2. Load UI dari File "main.ui":

Antarmuka pengguna (UI) dari file "main.ui" dimuat ke dalam objek self.

3. Pengaturan Judul Jendela:

Judul jendela diatur menjadi "Andika Motor".

4. Metode Logout:

Metode logout digunakan untuk menonaktifkan menu admin, menu kasir, dan menu laporan.

5. Metode Login:

Metode login membuka jendela login menggunakan objek `login.login()`. Jika login berhasil, menu yang sesuai diaktifkan berdasarkan peran pengguna (admin atau kasir).

```
def usermgt(self):
    self.umw = um.user_management()
    self.mdiArea.addSubWindow(self.umw)
    self.umw.show()

def barang(self):
    self.br = b.barang()
    self.mdiArea.addSubWindow(self.br)
    self.br.show()

def vendor(self):
    self.vn = v.vendor()
    self.mdiArea.addSubWindow(self.vn)
    self.vn.show()

def laporan(self):
    self.lp = l.laporan()
    self.mdiArea.addSubWindow(self.lp)
    self.lp.show()

def kasir(self):
    self.ksr = ks.kasir()
    self.mdiArea.addSubWindow(self.ksr)
    self.ksr.show()

def tentang(self):
    self.tn = tentang.tentang()
    self.tn.exec()
```

6. Metode-metode untuk Menangani Menu Aplikasi:

Metode-metode seperti `usermgt`, `barang`, `vendor`, `laporan`, `kasir`, dan `tentang` masing-masing digunakan untuk menangani tombol aksi yang terkait dengan fungsi-fungsi seperti manajemen pengguna, data barang, data vendor, laporan, kasir, dan informasi tentang aplikasi.

```
app = QtWidgets.QApplication(sys.argv)
window = MainWindow()
window.show()
app.exec_()
```

7. Instansiasi Objek Aplikasi dan Jendela:

Objek `QApplication` dibuat untuk menginisialisasi aplikasi PyQt5.

Objek `MainWindow` dibuat dan ditampilkan.

`app.exec_()` digunakan untuk menjalankan siklus utama aplikasi.

Kode ini menciptakan jendela utama aplikasi dengan berbagai fungsi dan menu yang terkait. Pengguna dapat melakukan login, dan berdasarkan peran pengguna, menu yang tersedia akan diaktifkan atau dinonaktifkan. Aplikasi ini menggunakan multiple document interface (MDI) untuk menangani sub-jendela seperti manajemen pengguna, data barang, dan sebagainya