LAPORAN TUGAS AHIR

**APLIKASI MOBILE BANKING**

Mata Kuliah Pemrograman Berorientasi Objek



Disusun Oleh:

1. Yoga prasetyo (2213020106)
2. Alvin Arya (2213020222)
3. Frizky Wahyu Andika (2213020178)

PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNIK DAN ILMU KOMPUTER UNIVERSITAS NUSANTARA PGRI KEDIRI TAHUN 2023

DAFTAR ISI

[DAFTAR ISI 2](file:///C:\Users\user\Downloads\Laporan%20PBO%20(1)%20(1).docx#_Toc155120597)

[Flowchat sistem Mobile Banking 3](file:///C:\Users\user\Downloads\Laporan%20PBO%20(1)%20(1).docx#_Toc155120598)

[Class diagram 4](file:///C:\Users\user\Downloads\Laporan%20PBO%20(1)%20(1).docx#_Toc155120599)

[Hasil program dan penjelasan 5](file:///C:\Users\user\Downloads\Laporan%20PBO%20(1)%20(1).docx#_Toc155120600)

Daftar pustaka……………………………………………………………………………………………………………………………………19

Flowchat sistem Mobile Banking

USER

LOGIN

START

HOME

PAGE

DEPOSIT

SALDO

TRANSFER

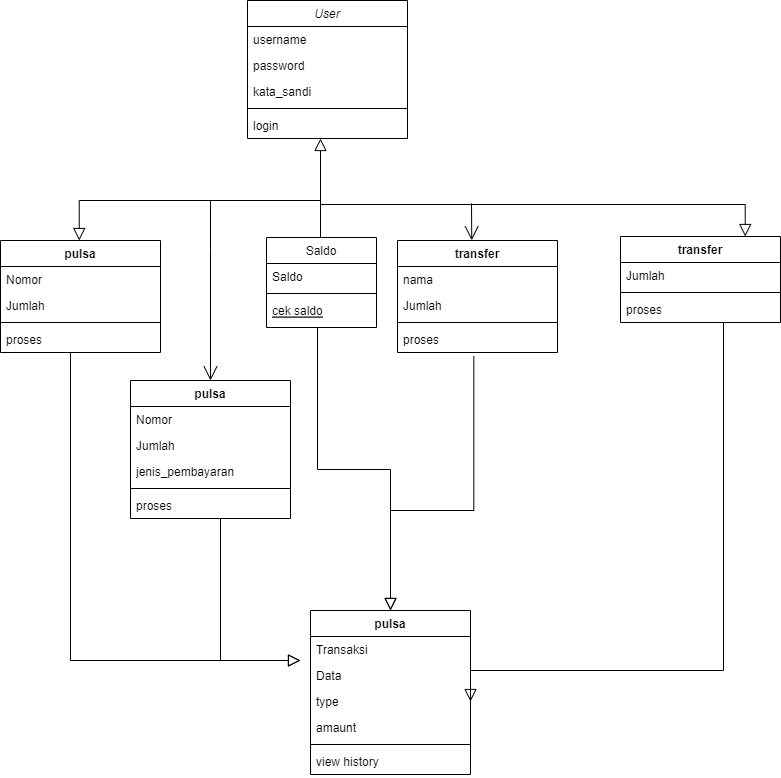
PULSA

DOMPET DIGITAL

RIWAYAT

END

Class diagram



Hasil program dan penjelasan

from PyQt5 import QtCore, QtGui, QtWidgets

import mysql.connector

import sys

class HalamanUtama(QtWidgets.QWidget):

    def \_\_init\_\_(self, username, user\_id, cursor, connection):

        super().\_\_init\_\_()

        self.setWindowTitle("Mobile Banking")

        self.setGeometry(100, 100, 400, 300)

        self.username = username

        self.user\_id = user\_id

        self.cursor = cursor

        self.connection = connection

        layout = QtWidgets.QVBoxLayout(self)

        # Menambahkan widget ke dalam layout

        label\_gambar = QtWidgets.QLabel(self)

        pixmap = QtGui.QPixmap("bank.png")  # Ganti dengan path ke gambar Anda

        pixmap\_terukur = pixmap.scaled(200, 200, QtCore.Qt.KeepAspectRatio)

        label\_gambar.setPixmap(pixmap\_terukur)

        label\_gambar.setAlignment(QtCore.Qt.AlignCenter)

        layout.addWidget(label\_gambar)

        label\_selamat\_datang = QtWidgets.QLabel(f"Selamat datang di Mobile Banking, {username}!", self)

        label\_selamat\_datang.setAlignment(QtCore.Qt.AlignCenter)

        layout.addWidget(label\_selamat\_datang)

        tombol\_saldo = QtWidgets.QPushButton("Saldo", self)

        tombol\_saldo.clicked.connect(self.tampilkan\_saldo)

        layout.addWidget(tombol\_saldo)

        tombol\_transfer = QtWidgets.QPushButton("Transfer", self)

        tombol\_transfer.clicked.connect(self.tampilkan\_menu\_transfer)

        layout.addWidget(tombol\_transfer)

        tombol\_deposit = QtWidgets.QPushButton("Deposit", self)

        tombol\_deposit.clicked.connect(self.tampilkan\_menu\_deposit)

        layout.addWidget(tombol\_deposit)

        tombol\_pulsa = QtWidgets.QPushButton("Pulsa", self)

        tombol\_pulsa.clicked.connect(self.tampilkan\_menu\_pulsa)

        layout.addWidget(tombol\_pulsa)

        tombol\_dompet\_digital = QtWidgets.QPushButton("Dompet Digital", self)

        tombol\_dompet\_digital.clicked.connect(self.tampilkan\_menu\_dompet\_digital)

        layout.addWidget(tombol\_dompet\_digital)

        tombol\_riwayat = QtWidgets.QPushButton("Riwayat", self)

        tombol\_riwayat.clicked.connect(self.tampilkan\_menu\_riwayat)

        layout.addWidget(tombol\_riwayat)

def tampilkan\_menu\_riwayat(self):

    # Ambil transaksi terbaru setiap kali menu riwayat ditampilkan

    transaksi = self.ambil\_transaksi\_terbaru()

    menu\_riwayat = QtWidgets.QDialog(self)

    menu\_riwayat.setWindowTitle("Riwayat Transaksi")

    menu\_riwayat.setGeometry(200, 200, 600, 400)

    tabel = QtWidgets.QTableWidget(menu\_riwayat)

    tabel.setColumnCount(3)

    tabel.setHorizontalHeaderLabels(["Tanggal", "Jenis", "Jumlah"])

    for baris, transaksi in enumerate(transaksi):

        tabel.insertRow(baris)

        tabel.setItem(baris, 0, QtWidgets.QTableWidgetItem(transaksi["tanggal"]))

        tabel.setItem(baris, 1, QtWidgets.QTableWidgetItem(transaksi["jenis"]))

        tabel.setItem(baris, 2, QtWidgets.QTableWidgetItem(str(transaksi["jumlah"])))

    tombol\_keluar = QtWidgets.QPushButton("Kembali", menu\_riwayat)

    tombol\_keluar.clicked.connect(menu\_riwayat.reject)

    layout\_grid = QtWidgets.QGridLayout(menu\_riwayat)

    layout\_grid.addWidget(tabel, 0, 0, 1, 2)

    layout\_grid.addWidget(tombol\_keluar, 1, 0, 1, 2)

    menu\_riwayat.exec\_()

def fetch\_latest\_transactions(self):

        try:

            query = "SELECT date, type, amount FROM transactions\_history WHERE user\_id = %s ORDER BY date DESC"

            self.cursor.execute(query, (self.user\_id,))

            latest\_transactions = self.cursor.fetchall()

            # Print for debugging

            print("Latest Transactions:", latest\_transactions)

            transactions\_list = []

            for transaction in latest\_transactions:

                # Convert date to a human-readable format

                formatted\_date = datetime.strftime(transaction[0], "%Y-%m-%d")

                transactions\_list.append({

                    "date": formatted\_date,

                    "type": transaction[1],

                    "amount": transaction[2]

                })

            return transactions\_list

        except Exception as e:

            print(f"Error fetching transactions\_history: {e}")

            return []

def tampilkan\_menu\_pulsa(self):

    menu\_pulsa = QtWidgets.QDialog(self)

    menu\_pulsa.setWindowTitle("Menu Pulsa")

    menu\_pulsa.setGeometry(200, 200, 400, 300)

    # Tambahkan QLabel untuk ikon pulsa

    label\_icon\_pulsa = QtWidgets.QLabel(menu\_pulsa)

    pixmap\_icon\_pulsa = QtGui.QPixmap("pulsa.png")  # Ganti dengan path ke ikon pulsa Anda

    pixmap\_icon\_pulsa\_terukur = pixmap\_icon\_pulsa.scaled(100, 100, QtCore.Qt.KeepAspectRatio)

    label\_icon\_pulsa.setPixmap(pixmap\_icon\_pulsa\_terukur)

    label\_icon\_pulsa.setAlignment(QtCore.Qt.AlignCenter)

    label\_nomor\_hp = QtWidgets.QLabel("Nomor HP:", menu\_pulsa)

    label\_pembelian = QtWidgets.QLabel("Pilihan Pembelian:", menu\_pulsa)

    self.entry\_nomor\_hp = QtWidgets.QLineEdit(menu\_pulsa)

    # Buat tombol untuk setiap opsi pembelian dengan label seragam

    tombol\_5000 = QtWidgets.QPushButton("Rp. 5.000", menu\_pulsa)

    tombol\_10000 = QtWidgets.QPushButton("Rp. 10.000", menu\_pulsa)

    tombol\_15000 = QtWidgets.QPushButton("Rp. 15.000", menu\_pulsa)

    tombol\_20000 = QtWidgets.QPushButton("Rp. 20.000", menu\_pulsa)

    tombol\_25000 = QtWidgets.QPushButton("Rp. 25.000", menu\_pulsa)

    tombol\_30000 = QtWidgets.QPushButton("Rp. 30.000", menu\_pulsa)

    # Hubungkan setiap tombol ke fungsi beli\_pulsa dengan jumlah yang sesuai

    tombol\_5000.clicked.connect(lambda: self.beli\_pulsa(5000))

    tombol\_10000.clicked.connect(lambda: self.beli\_pulsa(10000))

    tombol\_15000.clicked.connect(lambda: self.beli\_pulsa(15000))

    tombol\_20000.clicked.connect(lambda: self.beli\_pulsa(20000))

    tombol\_25000.clicked.connect(lambda: self.beli\_pulsa(25000))

    tombol\_30000.clicked.connect(lambda: self.beli\_pulsa(30000))

    tombol\_keluar = QtWidgets.QPushButton("Kembali", menu\_pulsa)

    tombol\_keluar.clicked.connect(menu\_pulsa.reject)

    layout\_grid = QtWidgets.QGridLayout(menu\_pulsa)

    layout\_grid.addWidget(label\_nomor\_hp, 0, 0)

    layout\_grid.addWidget(self.entry\_nomor\_hp, 0, 1)

    layout\_grid.addWidget(label\_pembelian, 1, 0)

    # Tambahkan tombol ke dalam layout

    layout\_grid.addWidget(tombol\_5000, 1, 1)

    layout\_grid.addWidget(tombol\_10000, 1, 2)

    layout\_grid.addWidget(tombol\_15000, 2, 1)

    layout\_grid.addWidget(tombol\_20000, 2, 2)

    layout\_grid.addWidget(tombol\_25000, 3, 1)

    layout\_grid.addWidget(tombol\_30000, 3, 2)

    # Tambahkan QLabel untuk ikon pulsa

    layout\_grid.addWidget(label\_icon\_pulsa, 0, 3, 4, 1)

    layout\_grid.addWidget(tombol\_keluar, 4, 0, 1, 4)

    menu\_pulsa.exec\_()

def beli\_pulsa(self, jumlah\_beli):

    try:

        # Dapatkan informasi pulsa dari field input

        nomor\_hp = self.entry\_nomor\_hp.text()

        # Lakukan validasi dan operasi pembelian pulsa

        if nomor\_hp and jumlah\_beli > 0:

            # Panggil fungsi untuk menangani pembelian pulsa

            self.beli\_pulsa(nomor\_hp, jumlah\_beli)

            QtWidgets.QMessageBox.information(

                self,

                "Pembelian Pulsa Berhasil",

                f"Pembelian pulsa sejumlah {jumlah\_beli} untuk {nomor\_hp} berhasil."

            )

        else:

            QtWidgets.QMessageBox.warning(

                self,

                "Input Tidak Valid",

                "Harap masukkan informasi yang valid untuk pembelian pulsa."

            )

    except ValueError:

        QtWidgets.QMessageBox.warning(

            self,

            "Input Tidak Valid",

            "Harap masukkan jumlah numerik yang valid untuk pembelian pulsa."

        )

def beli\_pulsa(self, nomor\_hp, jumlah\_beli):

    try:

        # Output debugging untuk mencetak Nomor HP yang diberikan

        print(f"Nomor HP yang Diberikan: {nomor\_hp}")

        # Dapatkan user\_id berdasarkan nomor\_hp yang diberikan

        query\_user\_id = "SELECT id FROM users WHERE LOWER(TRIM(nomor\_hp)) = LOWER(TRIM(%s))"

        self.cursor.execute(query\_user\_id, (nomor\_hp,))

        hasil\_user\_id = self.cursor.fetchone()

        if hasil\_user\_id is not None:

            user\_id = hasil\_user\_id[0]

            # Lakukan logika untuk menangani pembelian pulsa, sebagai contoh, mengurangi saldo

            # Untuk kesederhanaan, mari anggap pembelian pulsa hanya mengurangkan saldo

            query\_saldo = "SELECT saldo FROM users WHERE id = %s"

            self.cursor.execute(query\_saldo, (user\_id,))

            hasil\_saldo = self.cursor.fetchone()

            if hasil\_saldo is not None:

                saldo\_sekarang = hasil\_saldo[0]

     # Kurangkan jumlah dari saldo sekarang

                saldo\_baru = saldo\_sekarang - jumlah\_beli

     # Perbarui saldo di database

                query\_perbarui = "UPDATE users SET saldo = %s WHERE id = %s"

                self.cursor.execute(query\_perbarui, (saldo\_baru, user\_id))

                self.connection.commit()

     # Masukkan informasi pembelian pulsa ke dalam tabel pulsa\_purchases

                query\_masukkan\_pembelian\_pulsa = "INSERT INTO pulsa\_purchases (user\_id, nomor\_hp, jumlah\_beli) VALUES (%s, %s, %s)"

                self.cursor.execute(query\_masukkan\_pembelian\_pulsa, (user\_id, nomor\_hp, jumlah\_beli))

                self.connection.commit()

                print(f"Pembelian pulsa berhasil untuk pengguna {user\_id}. Saldo baru: {saldo\_baru}")

            else:

                print("Pengguna tidak ditemukan.")

        else:

            print("Nomor HP tidak ditemukan.")

    except Exception as e:

        print(f"Error dalam beli\_pulsa: {e}")

    def tampilkan\_menu\_dompet\_digital(self):

        menu\_dompet\_digital = QtWidgets.QDialog(self)

        menu\_dompet\_digital.setWindowTitle("Dompet Digital")

        menu\_dompet\_digital.setGeometry(200, 200, 400, 300)

    # Tambahkan QLabel untuk gambar dompet digital

        label\_gambar\_dompet\_digital = QtWidgets.QLabel(menu\_dompet\_digital)

        pixmap\_gambar\_dompet\_digital = QtGui.QPixmap("wallet.png")

        pixmap\_dompet\_digital\_terukur = pixmap\_gambar\_dompet\_digital.scaled(100, 100, QtCore.Qt.KeepAspectRatio)

        label\_gambar\_dompet\_digital.setPixmap(pixmap\_dompet\_digital\_terukur)

        label\_gambar\_dompet\_digital.setAlignment(QtCore.Qt.AlignCenter)

        label\_nomor = QtWidgets.QLabel("Nomor:", menu\_dompet\_digital)

        label\_jumlah = QtWidgets.QLabel("Jumlah:", menu\_dompet\_digital)

        label\_jenis\_dompet = QtWidgets.QLabel("Jenis Dompet:", menu\_dompet\_digital)

    # Gunakan atribut instance di sini

        self.entry\_nomor = QtWidgets.QLineEdit(menu\_dompet\_digital)

        self.entry\_jumlah = QtWidgets.QLineEdit(menu\_dompet\_digital)

        self.combo\_jenis\_dompet = QtWidgets.QComboBox(menu\_dompet\_digital)

        self.combo\_jenis\_dompet.addItems(["OVO", "Shopee", "Dana"])

        tombol\_top\_up = QtWidgets.QPushButton("Top Up", menu\_dompet\_digital)

        tombol\_top\_up.clicked.connect(self.top\_up\_amount)

        tombol\_kembali = QtWidgets.QPushButton("Kembali", menu\_dompet\_digital)

        tombol\_kembali.clicked.connect(menu\_dompet\_digital.reject)

        layout\_grid = QtWidgets.QGridLayout(menu\_dompet\_digital)

        layout\_grid.addWidget(label\_gambar\_dompet\_digital, 0, 0, 1, 2)  # Tambahkan label gambar

        layout\_grid.addWidget(label\_nomor, 1, 0)

        layout\_grid.addWidget(self.entry\_nomor, 1, 1)

        layout\_grid.addWidget(label\_jumlah, 2, 0)

        layout\_grid.addWidget(self.entry\_jumlah, 2, 1)

        layout\_grid.addWidget(label\_jenis\_dompet, 3, 0)

        layout\_grid.addWidget(self.combo\_jenis\_dompet, 3, 1)

        layout\_grid.addWidget(tombol\_top\_up, 4, 0, 1, 2)

        layout\_grid.addWidget(tombol\_kembali, 5, 0, 1, 2)

        menu\_dompet\_digital.exec\_()

    def top\_up\_amount(self):

     try:

        # Dapatkan informasi top-up dari field input

        nomor = self.entry\_nomor.text()

        jumlah = float(self.entry\_jumlah.text())

        jenis\_dompet = self.combo\_jenis\_dompet.currentText()

        # Lakukan validasi dan operasi top-up

        if nomor and jumlah > 0:

            # Top-up saldo (panggil fungsi yang benar)

            self.top\_up\_balance(jumlah)

            # Hubungkan dompet digital ke akun pengguna

            self.connect\_digital\_wallet(nomor, jenis\_dompet)

            QtWidgets.QMessageBox.information(

                self,

                "Top Up Berhasil",

                f"Top Up sejumlah {jumlah} ke {jenis\_dompet} dengan Nomor {nomor} berhasil.")

        else:

            QtWidgets.QMessageBox.warning(

                self,"Input Tidak Valid","Harap masukkan informasi yang valid untuk top-up.")

     except ValueError:

        QtWidgets.QMessageBox.warning(

            self,"Input Tidak Valid","Harap masukkan jumlah numerik yang valid untuk top-up.")

def connect\_digital\_wallet(self, nomor\_dompet, jenis\_dompet):

    try:

        # Periksa apakah pengguna sudah memiliki dompet digital

        query\_periksa = "SELECT \* FROM digital\_wallets WHERE user\_id = %s"

        self.cursor.execute(query\_periksa, (self.user\_id,))

        dompet\_digital\_ada = self.cursor.fetchone()

        if dompet\_digital\_ada:

            # Perbarui dompet yang sudah ada

            query\_perbarui = "UPDATE digital\_wallets SET nomor\_dompet = %s, jenis\_dompet = %s WHERE user\_id = %s"

            self.cursor.execute(query\_perbarui, (nomor\_dompet, jenis\_dompet, self.user\_id))

        else:

            # Masukkan entri dompet baru

            query\_masukkan = "INSERT INTO digital\_wallets (user\_id, nomor\_dompet, jenis\_dompet) VALUES (%s, %s, %s)"

            self.cursor.execute(query\_masukkan, (self.user\_id, nomor\_dompet, jenis\_dompet))

            self.connection.commit()

        print(f"Dompet digital terhubung dengan sukses. Nomor Dompet: {nomor\_dompet}, Jenis Dompet: {jenis\_dompet}")

    except Exception as e:

        print(f"Error dalam connect\_digital\_wallet: {e}")

def top\_up\_balance(self, jumlah):

    try:

        # Query saldo sekarang dari database

        query\_saldo = "SELECT saldo FROM users WHERE id = %s"

        self.cursor.execute(query\_saldo, (self.user\_id,))

        hasil\_saldo = self.cursor.fetchone()

        if hasil\_saldo is not None:

            saldo\_sekarang = hasil\_saldo[0]

        # Top-up sejumlah ke saldo sekarang

            saldo\_baru = saldo\_sekarang + jumlah

        # Perbarui saldo di database

            query\_perbarui = "UPDATE users SET saldo = %s WHERE id = %s"

            self.cursor.execute(query\_perbarui, (saldo\_baru, self.user\_id))

            self.connection.commit()

            print(f"Top-up berhasil. Saldo baru: {saldo\_baru}")

        else:

            print("Pengguna tidak ditemukan.")

    except Exception as e:

        print(f"Error dalam top\_up\_balance: {e}")

def tampilkan\_saldo(self):

    dialog\_saldo = QtWidgets.QDialog(self)

    dialog\_saldo.setWindowTitle("Saldo Anda")

    dialog\_saldo.setGeometry(200, 200, 400, 250)

    query\_saldo = "SELECT saldo FROM users WHERE id = %s"

    self.cursor.execute(query\_saldo, (self.user\_id,))

    hasil\_saldo = self.cursor.fetchone()

    if hasil\_saldo is not None:

        jumlah\_saldo = hasil\_saldo[0]

        saldo\_terformat = "{:,}".format(jumlah\_saldo)

        label\_saldo = QtWidgets.QLabel(f"Saldo Anda: Rp {saldo\_terformat}", dialog\_saldo)

        label\_saldo.setAlignment(QtCore.Qt.AlignCenter)

        # Menambahkan gambar ATM

        label\_gambar = QtWidgets.QLabel(dialog\_saldo)

        pixmap = QtGui.QPixmap("kartu.png")  # Ganti dengan path ke gambar Anda

        pixmap\_terukur = pixmap.scaledToWidth(200)

        label\_gambar.setPixmap(pixmap\_terukur)

        label\_gambar.setAlignment(QtCore.Qt.AlignCenter)

        tombol\_ok = QtWidgets.QPushButton("OK", dialog\_saldo)

        tombol\_ok.clicked.connect(dialog\_saldo.accept)

        layout\_grid = QtWidgets.QGridLayout(dialog\_saldo)

        layout\_grid.addWidget(label\_gambar, 0, 0, 1, 2)  # Tambahkan gambar ATM

        layout\_grid.addWidget(label\_saldo, 1, 0, 1, 2)

        layout\_grid.addWidget(tombol\_ok, 2, 0, 1, 2)

        dialog\_saldo.exec\_()

    else:

        QtWidgets.QMessageBox.warning(

            self,"Error","Gagal mendapatkan informasi saldo. Silakan coba lagi.")

def tampilkan\_menu\_transfer(self):

        menu\_transfer = QtWidgets.QDialog(self)

        menu\_transfer.setWindowTitle("Menu Transfer")

        menu\_transfer.setGeometry(200, 200, 400, 300)

    # Tambahkan widget untuk gambar transfer

        label\_gambar\_transfer = QtWidgets.QLabel(menu\_transfer)

        pixmap\_gambar\_transfer = QtGui.QPixmap("transfer.jpg")

        pixmap\_terukur\_transfer = pixmap\_gambar\_transfer.scaled(100, 100, QtCore.Qt.KeepAspectRatio)

        label\_gambar\_transfer.setPixmap(pixmap\_terukur\_transfer)

        label\_gambar\_transfer.setAlignment(QtCore.Qt.AlignCenter)

    # Tambahkan widget untuk detail transfer

        label\_pengirim = QtWidgets.QLabel(f"Pengirim: {self.username}", menu\_transfer)

        label\_penerima = QtWidgets.QLabel("Penerima:", menu\_transfer)

        label\_jumlah = QtWidgets.QLabel("Jumlah:", menu\_transfer)

        entry\_penerima = QtWidgets.QLineEdit(menu\_transfer)

        entry\_jumlah = QtWidgets.QLineEdit(menu\_transfer)

        tombol\_transfer = QtWidgets.QPushButton("Transfer", menu\_transfer)

        tombol\_transfer.clicked.connect(lambda: self.transfer\_amount(entry\_penerima.text(), entry\_jumlah.text()))

        tombol\_kembali = QtWidgets.QPushButton("Kembali", menu\_transfer)

        tombol\_kembali.clicked.connect(menu\_transfer.reject)

    # Atur tata letak untuk menu transfer

        layout\_grid = QtWidgets.QGridLayout(menu\_transfer)

        layout\_grid.addWidget(label\_gambar\_transfer, 0, 0, 1, 2)

        layout\_grid.addWidget(label\_pengirim, 1, 0, 1, 2)

        layout\_grid.addWidget(label\_penerima, 2, 0)

        layout\_grid.addWidget(entry\_penerima, 2, 1)

        layout\_grid.addWidget(label\_jumlah, 3, 0)

        layout\_grid.addWidget(entry\_jumlah, 3, 1)

        layout\_grid.addWidget(tombol\_transfer, 4, 0, 1, 2)

        layout\_grid.addWidget(tombol\_kembali, 5, 0, 1, 2)

        menu\_transfer.exec\_()

def transfer\_amount(self, username\_penerima, jumlah\_str):

    try:

        # Dapatkan ID pengirim

        query\_id\_pengirim = "SELECT id FROM users WHERE username = %s"

        self.cursor.execute(query\_id\_pengirim, (self.username,))

        hasil\_pengirim = self.cursor.fetchone()

        if hasil\_pengirim:

            id\_pengirim = hasil\_pengirim[0]

            # Dapatkan ID penerima

            query\_id\_penerima = "SELECT id FROM users WHERE username = %s"

            self.cursor.execute(query\_id\_penerima, (username\_penerima,))

            hasil\_penerima = self.cursor.fetchone()

            if hasil\_penerima:

                id\_penerima = hasil\_penerima[0]

            # Konversi jumlah ke integer

                jumlah = int(jumlah\_str)

            # Periksa apakah pengirim memiliki saldo yang cukup

                if jumlah > 0:

            # Perbarui saldo pengirim

                    query\_perbarui\_saldo\_pengirim = "UPDATE users SET saldo = saldo - %s WHERE id = %s"

                    self.cursor.execute(query\_perbarui\_saldo\_pengirim, (jumlah, id\_pengirim))

            # Perbarui saldo penerima

                    query\_perbarui\_saldo\_penerima = "UPDATE users SET saldo = saldo + %s WHERE id = %s"

                    self.cursor.execute(query\_perbarui\_saldo\_penerima, (jumlah, id\_penerima))

            # Masukkan rekam transfer

                    query\_masukkan\_transfer = "INSERT INTO transfers (sender\_id, receiver\_id, amount) VALUES (%s, %s, %s)"

                    self.cursor.execute(query\_masukkan\_transfer, (id\_pengirim, id\_penerima, jumlah))

            # Commit perubahan ke database

                    self.connection.commit()

                    # Tampilkan pesan sukses (sesuaikan jika diperlukan)

                    QtWidgets.QMessageBox.information(

                        self, "Transfer Berhasil",f"Berhasil mentransfer {jumlah} ke {username\_penerima}." )

                else:

                    # Tampilkan pesan jumlah tidak valid (sesuaikan jika diperlukan)

                    QtWidgets.QMessageBox.warning(

                        self,"Transfer Gagal","Jumlah tidak valid. Silakan coba lagi.")

            else:

                # Tampilkan pesan penerima tidak ditemukan (sesuaikan jika diperlukan)

                QtWidgets.QMessageBox.warning(

                    self,"Transfer Gagal",f"Penerima dengan nama pengguna {username\_penerima} tidak ditemukan.")

        else:

            # Tampilkan pesan pengirim tidak ditemukan (sesuaikan jika diperlukan)

            QtWidgets.QMessageBox.warning(

                self,"Transfer Gagal",f"Pengirim dengan nama pengguna {self.username} tidak ditemukan.")

    except mysql.connector.Error as err:

        print("Error MySQL: {}".format(err))

        QtWidgets.QMessageBox.critical(

            self,"Error",f"Terjadi kesalahan saat melakukan transfer: {err}")

        import traceback

        traceback.print\_exc()

    def tampilkan\_menu\_deposit(self):

        menu\_deposit = QtWidgets.QDialog(self)

        menu\_deposit.setWindowTitle("Menu Deposit")

        menu\_deposit.setGeometry(200, 200, 400, 300)

    # Tambahkan widget untuk detail deposit

        label\_jumlah = QtWidgets.QLabel("Jumlah:", menu\_deposit)

        entry\_jumlah = QtWidgets.QLineEdit(menu\_deposit)

        tombol\_deposit = QtWidgets.QPushButton("Deposit", menu\_deposit)

        tombol\_deposit.clicked.connect(lambda: self.deposit\_amount(entry\_jumlah.text()))

        tombol\_kembali = QtWidgets.QPushButton("Kembali", menu\_deposit)

        tombol\_kembali.clicked.connect(menu\_deposit.reject)

    # Tambahkan widget untuk gambar deposito

        label\_gambar\_deposito = QtWidgets.QLabel(menu\_deposit)

        pixmap = QtGui.QPixmap("deposito.png")

        pixmap\_terukur = pixmap.scaled(100, 100, QtCore.Qt.KeepAspectRatio)

        label\_gambar\_deposito.setPixmap(pixmap\_terukur)

        label\_gambar\_deposito.setAlignment(QtCore.Qt.AlignCenter)

    # Atur tata letak untuk menu deposit

        layout\_grid = QtWidgets.QGridLayout(menu\_deposit)

        layout\_grid.addWidget(label\_gambar\_deposito, 0, 0, 1, 2)

        layout\_grid.addWidget(label\_jumlah, 1, 0)

        layout\_grid.addWidget(entry\_jumlah, 1, 1)

        layout\_grid.addWidget(tombol\_deposit, 2, 0, 1, 2)

        layout\_grid.addWidget(tombol\_kembali, 3, 0, 1, 2)

        menu\_deposit.exec\_()

    def deposito\_jumlah(self, jumlah\_str):

     try:

        # Konversi jumlah menjadi integer

        jumlah = int(jumlah\_str)

        # Periksa apakah jumlah deposito valid

        if jumlah > 0:

         # Perbarui saldo pengguna

            query\_perbarui\_saldo = "UPDATE users SET saldo = saldo + %s WHERE id = %s"

            self.cursor.execute(query\_perbarui\_saldo, (jumlah, self.user\_id))

        # Masukkan catatan deposito

            query\_masukkan\_deposito = "INSERT INTO deposits (user\_id, amount) VALUES (%s, %s)"

            self.cursor.execute(query\_masukkan\_deposito, (self.user\_id, jumlah))

        # Komit perubahan ke database

            self.connection.commit()

        # Tampilkan pesan sukses (dapat disesuaikan)

            QtWidgets.QMessageBox.information(

                self,"Deposit Berhasil",f"Berhasil melakukan deposito sejumlah {jumlah}.")

        else:

            # Tampilkan pesan jumlah deposito tidak valid (dapat disesuaikan)

            QtWidgets.QMessageBox.warning(

                self, "Deposit Gagal", "Jumlah deposito tidak valid. Silakan coba lagi.")

     except mysql.connector.Error as err:

        print("Error MySQL: {}".format(err))

        QtWidgets.QMessageBox.critical(

            self, "Error", f"Terjadi kesalahan saat melakukan deposito: {err}")

        import traceback

        traceback.print\_exc()

class Ui\_Form(object):

    def setupUi(self, Form):

        Form.setObjectName("Form")

        Form.resize(330, 442)

        Form.setWindowFlags(QtCore.Qt.FramelessWindowHint)

        Form.setAttribute(QtCore.Qt.WA\_TranslucentBackground)

        Form.setStyleSheet("QPushButton#pushButton{\n""background-color:rgba(2, 65, 118, 255);\n""color:rgba(255, 255, 255, 200);\n""border-radius:5px;\n"

                           "}\n""QPushButton#pushButton:pressed{\n""padding-left:5px;\n""padding-top:5px;\n""background-color:rgba(2, 65, 118, 100);\n"

                           "background-position:calc(100% - 10px)center;\n""}\n""QPushButton#pushButton:hover{\n""background-color:rgba(2, 65, 118, 200);\n""}")

        self.widget = QtWidgets.QWidget(Form)

        self.widget.setGeometry(QtCore.QRect(10, 10, 290, 410))

        self.widget.setObjectName("widget")

        self.label = QtWidgets.QLabel(self.widget)

        self.label.setGeometry(QtCore.QRect(0, 0, 290, 410))

        self.label.setStyleSheet("background-color:rgba(16, 30, 41, 240);\n""border-radius:10px;")

        self.label.setText("")

        self.label.setObjectName("label")

        self.lineEdit = QtWidgets.QLineEdit(self.widget)

        self.lineEdit.setGeometry(QtCore.QRect(20, 210, 250, 30))

        font = QtGui.QFont()

        font.setPointSize(10)

        self.lineEdit.setFont(font)

        self.lineEdit.setStyleSheet("background-color:rgba(0, 0, 0, 0);\n""border:1px solid rgba(0, 0, 0, 0);\n""border-bottom-color:rgba(46, 82, 101, 255);\n"

                                   "color:rgb(255, 255, 255);\n""padding-bottom:7px")

        self.lineEdit.setObjectName("lineEdit")

        self.lineEdit\_2 = QtWidgets.QLineEdit(self.widget)

        self.lineEdit\_2.setGeometry(QtCore.QRect(20, 260, 250, 30))

        font = QtGui.QFont()

        font.setPointSize(10)

        self.lineEdit\_2.setFont(font)

        self.lineEdit\_2.setStyleSheet("background-color:rgba(0, 0, 0, 0);\n" "border:1px solid rgba(0, 0, 0, 0);\n""border-bottom-color:rgba(46, 82, 101, 255);\n"

                                     "color:rgb(255, 255, 255);\n""padding-bottom:7px")

        self.lineEdit\_2.setEchoMode(QtWidgets.QLineEdit.Password)

        self.lineEdit\_2.setObjectName("lineEdit\_2")

        self.pushButton = QtWidgets.QPushButton(self.widget)

        self.pushButton.setGeometry(QtCore.QRect(20, 320, 250, 40))

        font = QtGui.QFont()

        font.setPointSize(10)

        font.setBold(True)

        font.setWeight(75)

        self.pushButton.setFont(font)

        self.pushButton.setObjectName("pushButton")

        self.label\_2 = QtWidgets.QLabel(self.widget)

        self.label\_2.setGeometry(QtCore.QRect(60, 30, 180, 150))

        original\_pixmap = QtGui.QPixmap("icon orang.png")  # Ganti dengan path logo Anda

        scaled\_pixmap = original\_pixmap.scaled(180, 150, QtCore.Qt.KeepAspectRatio)

        self.label\_2.setPixmap(scaled\_pixmap)

        self.label\_2.setAlignment(QtCore.Qt.AlignCenter)

        self.label\_2.setObjectName("label\_2")

        self.label\_3 = QtWidgets.QLabel(self.widget)

        self.label\_3.setGeometry(QtCore.QRect(50, 365, 211, 16))

        self.label\_3.setStyleSheet("color:rgba(255, 255, 255, 150);")

        self.label\_3.setObjectName("label\_3")

        # Menu setup

        self.menuBar = QtWidgets.QMenuBar(Form)

        self.menuBar.setGeometry(QtCore.QRect(0, 0, 330, 21))

        self.menuBar.setObjectName("menuBar")

        self.menuBar.setStyleSheet("background-color: rgba(16, 30, 41, 240); color: white;")

        self.menuBar.setGeometry(QtCore.QRect(0, 0, 330, 25))

        Form.setMenuBar(self.menuBar)

        self.retranslateUi(Form)

        QtCore.QMetaObject.connectSlotsByName(Form)

    def retranslateUi(self, Form):

        \_translate = QtCore.QCoreApplication.translate

        Form.setWindowTitle(\_translate("Form", "Form"))

        self.lineEdit.setPlaceholderText(\_translate("Form", "  Nama Pengguna"))

        self.lineEdit\_2.setPlaceholderText(\_translate("Form", "  Kata Sandi"))

        self.pushButton.setText(\_translate("Form", "L o g   I n"))

        self.label\_3.setText(\_translate("Form", "Lupa Nama Pengguna atau Kata Sandi?"))

    def show\_home\_page(self):

        username = self.lineEdit.text()

        password = self.lineEdit\_2.text()

        try:

            connection = mysql.connector.connect(

                host="localhost",

                user="root",

                password="",

                database="bankpbo")

            cursor = connection.cursor()

            query = "SELECT \* FROM users WHERE username = %s AND user\_password = %s"

            cursor.execute(query, (username, password))

            result = cursor.fetchone()

            if result:

                user\_id = result[0]

                self.home\_page = HomePage(username, user\_id, cursor, connection)

                self.home\_page.show()

            else:

                QtWidgets.QMessageBox.warning(

                    self.widget,"Login Gagal",

                    "Nama pengguna atau kata sandi tidak valid. Silakan coba lagi.")

        except mysql.connector.Error as err:

            print("Error MySQL: {}".format(err))

            import traceback

            traceback.print\_exc()

if \_\_name\_\_ == "\_\_main\_\_":

    app = QtWidgets.QApplication(sys.argv)

    Form = QtWidgets.QMainWindow()

    ui = Ui\_Form()

    ui.setupUi(Form)

    ui.pushButton.clicked.connect(ui.show\_home\_page)

    Form.setCentralWidget(ui.widget)

    Form.show()

    sys.exit(app.exec\_())

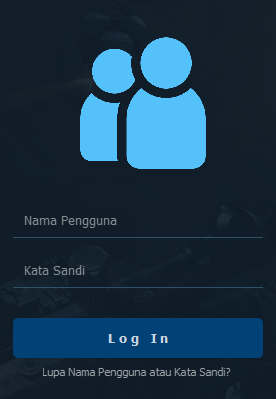
Alur program ini adalah sebagai berikut

1. **Login Page (Ui\_Form)**:
   * Aplikasi dimulai dengan menampilkan halaman login (**Ui\_Form**), yang mencakup elemen seperti **QLineEdit** untuk memasukkan nama pengguna dan kata sandi, serta tombol **QPushButton** untuk masuk.
   * Gambar logo dan latar belakang juga ditampilkan pada halaman login.
2. **Login Process**:
   * Ketika pengguna mengklik tombol "Login", fungsi **show\_home\_page** dijalankan.
   * Dalam **show\_home\_page**, program mencoba melakukan koneksi ke database MySQL menggunakan **mysql.connector**.
   * Setelah koneksi berhasil, program mengeksekusi query SQL untuk mencocokkan nama pengguna dan kata sandi yang dimasukkan oleh pengguna dengan entri dalam database.
   * Jika kombinasi nama pengguna dan kata sandi ditemukan, halaman utama (**HomePage**) ditampilkan.
3. **Home Page (HalamanUtama)**:
   * Halaman utama (**HomePage**) menampilkan layar utama aplikasi mobile banking setelah login berhasil.
   * Ini mencakup elemen seperti gambar logo bank, ucapan selamat datang, dan beberapa tombol untuk fungsi utama seperti cek saldo, transfer, deposit, pembelian pulsa, dompet digital, dan riwayat transaksi.
4. **Menu Transfer**:
   * Saat pengguna mengklik tombol "Transfer" di halaman utama, fungsi **tampilkan\_menu\_transfer** dijalankan.
   * Ini membuka dialog transfer (**menu\_transfer**) yang meminta informasi pengirim, penerima, dan jumlah transfer.
   * Fungsi **transfer\_amount** dipanggil untuk mengeksekusi transfer dana antar pengguna.
5. **Transfer Process**:
   * Dalam **transfer\_amount**, program mencari ID pengirim dan penerima berdasarkan nama pengguna.
   * Jumlah transfer dikonversi menjadi integer, dan saldo pengirim diperbarui di database.
   * Saldo penerima juga diperbarui, dan rekam transfer dimasukkan ke dalam tabel transaksi.
   * Seluruh proses dijalankan dalam satu transaksi database untuk menjaga konsistensi data.
6. **Menu Deposit**:
   * Saat pengguna mengklik tombol "Deposit" di halaman utama, fungsi **tampilkan\_menu\_deposit** dijalankan.
   * Ini membuka dialog deposit (**menu\_deposit**) yang meminta informasi jumlah deposit.
7. **Deposit Process**:
   * Dalam **deposito\_jumlah**, jumlah deposit dikonversi menjadi integer, dan saldo pengguna diperbarui di database.
   * Rekam deposit dimasukkan ke dalam tabel deposit untuk tujuan pelacakan.
8. **Menu Pulsa dan Dompet Digital**:
   * Fungsi **tampilkan\_menu\_pulsa** dan **tampilkan\_menu\_dompet\_digital** masing-masing menangani pembelian pulsa dan top-up dompet digital.
   * Informasi seperti nomor HP, jumlah pembelian, dan jenis dompet diminta dari pengguna, kemudian diproses dan diperbarui di database.
9. **Menu Riwayat**:
   * Fungsi **tampilkan\_menu\_riwayat** menampilkan riwayat transaksi pengguna dalam bentuk tabel menggunakan **QTableWidget**.
10. **Saldo**:

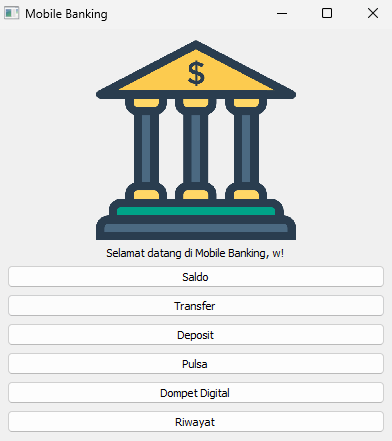
* Fungsi **tampilkan\_saldo** menampilkan dialog yang menampilkan saldo pengguna.

1. **Error Handling**:
   * Terdapat beberapa mekanisme penanganan kesalahan dan pesan peringatan untuk memberi informasi kepada pengguna jika ada masalah selama eksekusi program atau proses transaksi.

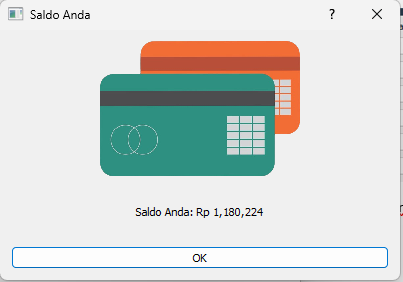
Semua fungsi terkait dengan interaksi pengguna dan pemrosesan transaksi telah diimplementasikan di dalam kelas **HalamanUtama**.



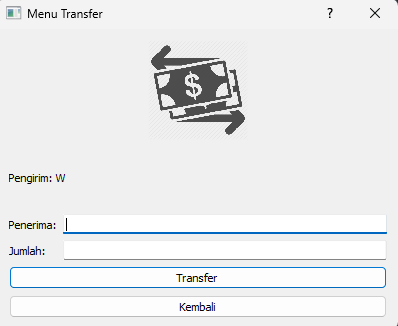
1. tampilan pertam adalah menampilkan page log in untuk melakukan transaksi
2. tampilan ke dua adalh menu menu yang terdapat pada mobile banking



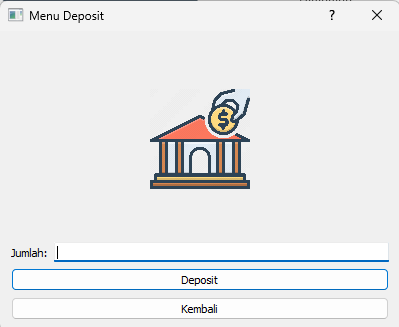
1. tampilan ke tiga menampilkan saldo



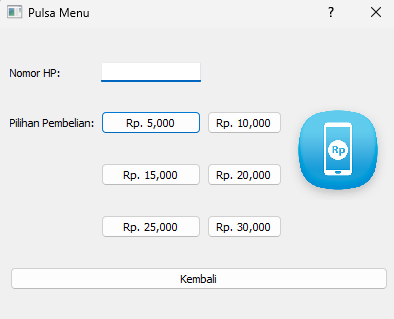
1. tampilan ke empat adalah tampilan transfer



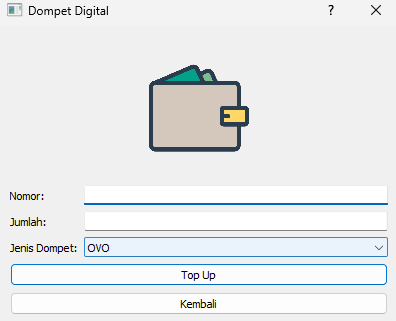
1. tampilan ke lima adalah tampilan deposit



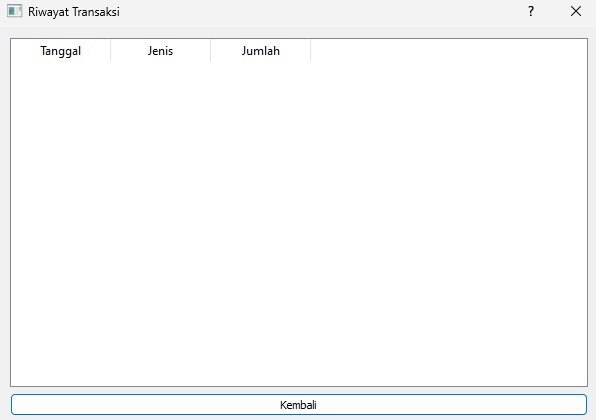
1. tampilan ke enam adalah tampilan pembelian pulsa



1. tampilan ke tuju adalah tampilan dompet digital



1. tampilan ke delapan adalah tampilan Riwayat



‘

**Daftar Pustaka**

1. Sathye, M. (1999). Adoption of Internet banking by Australian consumers: an empirical investigation. International Journal of Bank Marketing, 17(7), 324-334.
2. Lee, Y., Lee, J., & Lee, T. (2003). An empirical study on the wireless Internet acceptance model: a focus on motivation and habit. International Journal of Information Management, 23(6), 347-356.
3. Liao, Z., & Cheung, M. T. (2002). Internet-based e-banking and consumer attitudes: an empirical study. Information & Management, 39(4), 283-295.
4. Agarwal, R., & Prasad, J. (1997). The role of innovation characteristics and perceived voluntariness in the acceptance of information technologies. Decision Sciences, 28(3), 557-582.
5. Laukkanen, T. (2007). Internet vs mobile banking: comparing customer value perceptions. Business Process Management Journal, 13(6), 788-797.