**TECHNICAL REPORT**

**PEMROGRAMAN DESKTOP**

**MODUL 4**



**Disusun Oleh :**

TGL. PRAKTIKUM : Jum’at, 04 Desember 2020

NAMA : Achmad Farid Alfa Waid

NRP : 190411100073

KELOMPOK : 2

DOSEN : Moh. Kautsar Sophan, S.Kom., M.MT.

TELAH DISETUJUI TANGGAL :

...........................................

ASISTEN PRAKTIKUM

Nadia Asri

(180411100063)

**LABORATORIUM MULTIMEDIA COMPUTING**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**BAB I**

**TUJUAN DAN DASAR TEORI**

1. **TUJUAN**

Pada praktikum kali ini bertujuan untuk memahami tentang cara penggunaan class dari QMainWindow, QDialog, dan QWidget.

1. **DASAR TEORI**

QWidget terdiri dari beberapa komponen yang dapat ditampilkan dalam sebuah sebuah jendela window.

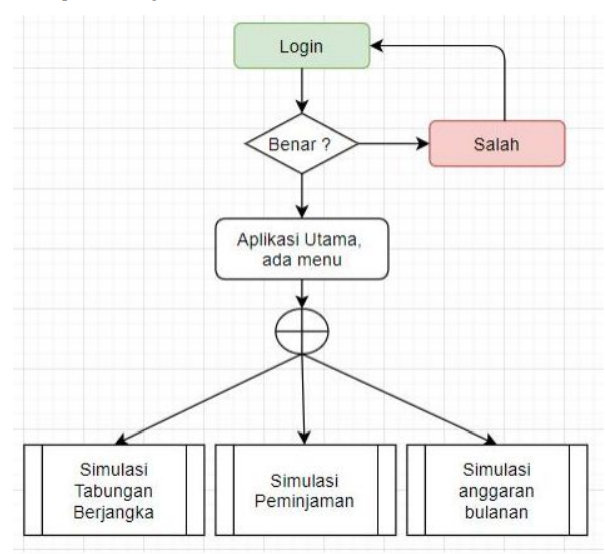
QMainWindow adalah sebuah window utama yang menyediakan struktur basic untuk membangun user interface pada aplikasi. Dapat berisi menu bar, QWidget, dll.

QDialog adalah window yang selalu berada di layer paling atas. Digunakan untuk menyelesaikan tugas-tugas jangka pendek seperti dialog (terima, tolak,dll).

**BAB II**

**PEMBAHASAN**

1. **SOAL**
2. Buat aplikasi dengan alur sebagai berikut

****

1. **JAWABAN**
2. **Login.py**

# -\*- coding: utf-8 -\*-

# Form implementation generated from reading ui file 'login.ui'

#

# Created by: PyQt5 UI code generator 5.15.1

#

# WARNING: Any manual changes made to this file will be lost when pyuic5 is

# run again. Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets

from mainwindow import Ui\_MainWindow

from PyQt5.Qt import \*

class Ui\_Dialog(object):

def setupUi(self, Dialog):

Dialog.setObjectName("Dialog")

Dialog.resize(395, 271)

Dialog.setInputMethodHints(QtCore.Qt.ImhNone)

self.label = QtWidgets.QLabel(Dialog)

self.label.setGeometry(QtCore.QRect(100, 20, 201, 41))

font = QtGui.QFont()

font.setFamily("Stencil")

font.setPointSize(18)

font.setBold(True)

font.setWeight(75)

self.label.setFont(font)

self.label.setObjectName("label")

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

self.label\_2.setGeometry(QtCore.QRect(10, 90, 131, 31))

font = QtGui.QFont()

font.setFamily("Arial Black")

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_2.setFont(font)

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

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

self.label\_3.setGeometry(QtCore.QRect(10, 150, 131, 31))

font = QtGui.QFont()

font.setFamily("Arial Black")

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_3.setFont(font)

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

self.username = QtWidgets.QLineEdit(Dialog)

self.username.setGeometry(QtCore.QRect(150, 90, 221, 31))

self.username.setObjectName("username")

self.password = QtWidgets.QLineEdit(Dialog)

self.password.setGeometry(QtCore.QRect(150, 150, 221, 31))

self.password.setObjectName("password")

self.button\_login = QtWidgets.QPushButton(Dialog)

self.button\_login.setGeometry(QtCore.QRect(140, 210, 111, 41))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.button\_login.setFont(font)

self.button\_login.setObjectName("button\_login")

self.button\_login.clicked.connect(self.check\_login)

self.retranslateUi(Dialog)

QtCore.QMetaObject.connectSlotsByName(Dialog)

def check\_login(self):

notif = QMessageBox()

username = str(self.username.text())

password = str(self.password.text())

if username == 'apak' and password == '123':

notif.setText("Selamat anda berhasil login!")

notif.exec()

Dialog.close()

self.MainWindow = QtWidgets.QMainWindow()

self.ui = Ui\_MainWindow()

self.ui.setupUi(self.MainWindow)

self.MainWindow.show()

else:

notif.setText("Anda gagal login!")

notif.exec()

def retranslateUi(self, Dialog):

\_translate = QtCore.QCoreApplication.translate

Dialog.setWindowTitle(\_translate("Dialog", "User Login"))

self.label.setText(\_translate("Dialog", "USER LOGIN!"))

self.label\_2.setText(\_translate("Dialog", "USERNAME"))

self.label\_3.setText(\_translate("Dialog", "PASSWORD"))

self.button\_login.setText(\_translate("Dialog", "LOGIN"))

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

import sys

app = QtWidgets.QApplication(sys.argv)

Dialog = QtWidgets.QDialog()

ui = Ui\_Dialog()

ui.setupUi(Dialog)

Dialog.show()

sys.exit(app.exec\_())

**mainwindow.py**

# -\*- coding: utf-8 -\*-

# Form implementation generated from reading ui file 'mainwindow.ui'

#

# Created by: PyQt5 UI code generator 5.15.1

#

# WARNING: Any manual changes made to this file will be lost when pyuic5 is

# run again. Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets

import tabungan\_berjangka, simulasi\_peminjaman,anggaran\_bulanan

class Ui\_MainWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(800, 600)

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

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

self.label.setGeometry(QtCore.QRect(170, 200, 461, 151))

font = QtGui.QFont()

font.setFamily("Arial Black")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.label.setFont(font)

self.label.setObjectName("label")

MainWindow.setCentralWidget(self.centralwidget)

self.menubar = QtWidgets.QMenuBar(MainWindow)

self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 26))

self.menubar.setObjectName("menubar")

self.menuMenu = QtWidgets.QMenu(self.menubar)

self.menuMenu.setObjectName("menuMenu")

MainWindow.setMenuBar(self.menubar)

self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)

self.actionSimulasi\_Tabungan\_Berjangka = QtWidgets.QAction(MainWindow)

self.actionSimulasi\_Tabungan\_Berjangka.setObjectName("actionSimulasi\_Tabungan\_Berjangka")

self.actionSimulasi\_Anggaran\_Bulanan = QtWidgets.QAction(MainWindow)

self.actionSimulasi\_Anggaran\_Bulanan.setObjectName("actionSimulasi\_Anggaran\_Bulanan")

self.actionSimulasi\_Peminjaman = QtWidgets.QAction(MainWindow)

self.actionSimulasi\_Peminjaman.setObjectName("actionSimulasi\_Peminjaman")

self.menuMenu.addAction(self.actionSimulasi\_Tabungan\_Berjangka)

self.menuMenu.addAction(self.actionSimulasi\_Anggaran\_Bulanan)

self.menuMenu.addAction(self.actionSimulasi\_Peminjaman)

self.menubar.addAction(self.menuMenu.menuAction())

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

self.actionSimulasi\_Tabungan\_Berjangka.triggered.connect(Simulasi\_Tabungan\_Berjangka)

self.actionSimulasi\_Anggaran\_Bulanan.triggered.connect(Simulasi\_Anggaran\_Bulanan)

self.actionSimulasi\_Peminjaman.triggered.connect(Simulasi\_Peminjaman)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

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

self.label.setText(\_translate("MainWindow", "ANDA MASUK KE MAIN WINDOW"))

self.menuMenu.setTitle(\_translate("MainWindow", "Menu"))

self.actionSimulasi\_Tabungan\_Berjangka.setText(\_translate("MainWindow", "Simulasi Tabungan Berjangka"))

self.actionSimulasi\_Anggaran\_Bulanan.setText(\_translate("MainWindow", "Simulasi Anggaran Bulanan"))

self.actionSimulasi\_Peminjaman.setText(\_translate("MainWindow", "Simulasi Peminjaman"))

#membuat fungsi Simulasi\_Tabungan\_Berjangka

def Simulasi\_Tabungan\_Berjangka():

#memanggil/menampilkan aplikasi Simulasi\_Tabungan\_Berjangka

tabungan\_berjangka.Layout()

#membuat fungsi Simulasi\_Anggaran\_Bulanan

def Simulasi\_Anggaran\_Bulanan():

#memanggil/menampilkan aplikasi Simulasi\_Anggaran\_Bulanan

anggaran\_bulanan.Layout()

#membuat fungsi Simulasi\_Peminjaman

def Simulasi\_Peminjaman():

#memanggil/menampilkan aplikasi Simulasi\_Peminjaman

simulasi\_peminjaman.Layout()

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

import sys

app = QtWidgets.QApplication(sys.argv)

MainWindow = QtWidgets.QMainWindow()

ui = Ui\_MainWindow()

ui.setupUi(MainWindow)

MainWindow.show()

sys.exit(app.exec\_())

**anggaran\_bulanan.py**

import sys

from PyQt5.QtWidgets import \*

from PyQt5.Qt import \*

#GLOBAL

#Inisisalisai pyqt

app = QApplication(sys.argv)

#mengatur style di window menjadi style fusion

app.setStyle("fusion")

#membuat variabel ex yang berisi fungsi QWidget

ex = QWidget()

#variabel self berisi nilai ex yang berarti berisi QWidget

self = ex

#membuat QLineEdit dengan nama variabel anggaran, cal\_button, hasil

self.anggaran = QLineEdit(self)

#membuat sebuah button dengan nama variable cal\_button

self.cal\_button = QPushButton("Calculate", self)

#membuat sebuah button dengan nama variable hasil4

self.hasil = QPushButton("Hasil", self)

#membuat fungsi dengan nama Layout

def Layout():

#membuat grid layout dengan nama variabel grid

grid = QGridLayout()

#mengatur batas pada isi conten layout grid yaitu batas kana,kiri,atas,bawah bernilai 10

grid.setContentsMargins(10,10,10,10)

#membuat sebuag QTabWidget dengan nama variabel tab

tab = QTabWidget(self)

#mengatur semua isi konten yang ada di tab menjadi tulisan bold berwarna hitam dengan ukuran 15px

tab.setStyleSheet("font: bold 15px; color: black;")

#membuat QVBoxLayout dengan nama variabel vbox

vbox = QVBoxLayout()

#membuat groupbox

group = QGroupBox(self)

#membuat vbox menjadi layout

group.setLayout(vbox)

#memasukkan group kedalam variabel tab atau QTabWidget

tab.addTab(group,"Anngaran Bulanan")

#memasukkan tab kedalam layout utama grid

grid.addWidget(tab, 2, 1)

#------------------------- KELOMPOK 1 -------------------------#

#membuat label1 dan akan dimasukkan ke dalam layout vbox

label1 = QLabel(self)

label1.setText("Masukkan Jumlah Pendapatan Anda per Bulan:")

vbox.addWidget(label1)

#mengatur anggaran dengan border 0.5 px dan berwarna solid black

self.anggaran.setStyleSheet("border: 0.5px solid black;")

#memasukkan anggaran kedalam layout vbox

vbox.addWidget(self.anggaran)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch()

#mengatur cal\_button dengan background berwarna #33f8c7

self.cal\_button.setStyleSheet("background-color: #33f8c7;")

#memasukkan cal\_button kedalam layout vbox

vbox.addWidget(self.cal\_button)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch(1)

##mengatur hasil dengan font bold dan ukuran 20px dan berwarna hitam

self.hasil.setStyleSheet("font: bold 20px; color: black")

#mengatur button hasil dengan setDisabled agar tidak daat diklik

self.hasil.setDisabled(True)

#memasukkan button hasil ke dalam layout vbox

vbox.addWidget(self.hasil)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch(1)

#connect button to fungsi on\_click

self.cal\_button.clicked.connect(on\_click)

#------------------------- END OF KELOMPOK 1 -------------------------#

#membuat grid menjadi layout utama di window

ex.setLayout(grid)

#Menentukan ukuran window dan title untuk menampilkan

ex.setGeometry(100,100,500,300)

#membuat judul window

ex.setWindowTitle("Simulasi Anggaran Bulanan")

#menampilan isi dari variabel ex

ex.show()

#membuat system exit

sys.exit(app.exec\_())

#membuat fungsi on\_click yang berisi sistem untuk menghitung nilai Anggaran

def on\_click():

#membuat variabel Anggaran yang berisi inputan dari anggaran yang akan dirubah menjadi sebuah float

Anggaran = float(self.anggaran.text())

#membuat variabel pokok, cicilan, nabung, lifestyle untuk menghitung nilai Anggaran

pokok = Anggaran \* 0.5

cicilan = Anggaran \* 0.3

nabung = Anggaran \* 0.15

lifestyle = Anggaran \* 0.05

#menampilkan nilai Anggaran pada button hasil

self.hasil.setText("Nilai Anggaran Pokok: Rp." + str("%.2f" %pokok) + "\n Nilai Anggaran Cicilan: Rp." + str("%.2f" %cicilan) + "\n Nilai Anggaran Menabung: Rp." + str("%.2f" %nabung) + "\n Nilai Anggaran Lifestyle: Rp." + str("%.2f" %lifestyle))

if \_\_name\_\_ == '\_\_main\_\_':

#memanggil fungsi Layout untuk menampilkan seluruh isi widget

Layout()

**tabungan\_berjangka.py**

import sys

from PyQt5.QtWidgets import \*

from PyQt5.Qt import \*

#GLOBAL

#Inisisalisai pyqt

app = QApplication(sys.argv)

#mengatur style di window menjadi style fusion

app.setStyle("fusion")

#membuat variabel ex yang berisi fungsi QWidget

ex = QWidget()

#variabel self berisi nilai ex yang berarti berisi QWidget

self = ex

#membuat QLineEdit dengan nama variabel awal, rutin, waktu bunga, admin

self.awal = QLineEdit(self)

self.rutin = QLineEdit(self)

self.waktu = QLineEdit(self)

self.bunga = QLineEdit(self)

self.pajak = QLineEdit(self)

#membuat sebuah button dengan nama variable cal\_button

self.cal\_button = QPushButton("Calculate", self)

#membuat sebuah button dengan nama variable hasil

self.hasil = QPushButton("Hasil", self)

#membuat fungsi dengan nama Layout

def Layout():

#membuat grid layout dengan nama variabel grid

grid = QGridLayout()

#mengatur batas pada isi conten layout grid yaitu batas kana,kiri,atas,bawah bernilai 10

grid.setContentsMargins(10,10,10,10)

#membuat sebuag QTabWidget dengan nama variabel tab

tab = QTabWidget(self)

#mengatur semua isi konten yang ada di tab menjadi tulisan bold berwarna hitam dengan ukuran 15px

tab.setStyleSheet("font: bold 15px; color: black;")

#membuat QVBoxLayout dengan nama variabel vbox

vbox = QVBoxLayout()

#membuat groupbox

group = QGroupBox(self)

#membuat vbox menjadi layout

group.setLayout(vbox)

#memasukkan group kedalam variabel tab atau QTabWidget

tab.addTab(group,"Tabungan Berjangka")

#memasukkan tab kedalam layout utama grid

grid.addWidget(tab, 2, 1)

#------------------------- KELOMPOK 1 -------------------------#

#membuat label1 dan akan dimasukkan ke dalam layout vbox

label1 = QLabel(self)

label1.setText("Masukkan Jumlah Saldo Awal:")

vbox.addWidget(label1)

#mengatur awal dengan border 0.5 px dan berwarna solid black

self.awal.setStyleSheet("border: 0.5px solid black;")

#memasukkan awal kedalam layout vbox

vbox.addWidget(self.awal)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch()

#membuat label1 dan akan dimasukkan ke dalam layout vbox

label1 = QLabel(self)

label1.setText("Masukkan Jumlah Saldo per Bulan:")

vbox.addWidget(label1)

#mengatur rutin dengan border 0.5 px dan berwarna solid black

self.rutin.setStyleSheet("border: 0.5px solid black;")

#memasukkan rutin kedalam layout vbox

vbox.addWidget(self.rutin)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch()

#membuat label1 dan akan dimasukkan ke dalam layout vbox

label1 = QLabel(self)

label1.setText("Masukkan Jangka Waktu (dalam bulan):")

vbox.addWidget(label1)

#mengatur waktu dengan border 0.5 px dan berwarna solid black

self.waktu.setStyleSheet("border: 0.5px solid black;")

#memasukkan waktu kedalam layout vbox

vbox.addWidget(self.waktu)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch()

#membuat label1 dan akan dimasukkan ke dalam layout vbox

label1 = QLabel(self)

label1.setText("Masukkan Bunga per Tahun:")

vbox.addWidget(label1)

#mengatur bunga dengan border 0.5 px dan berwarna solid black

self.bunga.setStyleSheet("border: 0.5px solid black;")

#memasukkan bunga kedalam layout vbox

vbox.addWidget(self.bunga)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch()

#membuat label1 dan akan dimasukkan ke dalam layout vbox

label1 = QLabel(self)

label1.setText("Masukkan Biaya Tarif Pajak:")

vbox.addWidget(label1)

#mengatur admin dengan border 0.5 px dan berwarna solid black

self.pajak.setStyleSheet("border: 0.5px solid black;")

#memasukkan admin kedalam layout vbox

vbox.addWidget(self.pajak)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch()

#mengatur cal\_button dengan background berwarna #33f8c7

self.cal\_button.setStyleSheet("background-color: #33f8c7;")

#memasukkan cal\_button kedalam layout vbox

vbox.addWidget(self.cal\_button)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch(1)

##mengatur hasil dengan font bold dan ukuran 20px dan berwarna hitam

self.hasil.setStyleSheet("font: bold 20px; color: black")

#mengatur button hasil dengan setDisabled agar tidak daat diklik

self.hasil.setDisabled(True)

#memasukkan button hasil ke dalam layout vbox

vbox.addWidget(self.hasil)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch(1)

#connect button to fungsi on\_click

self.cal\_button.clicked.connect(on\_click)

#------------------------- END OF KELOMPOK 1 -------------------------#

#membuat grid menjadi layout utama di window

ex.setLayout(grid)

#Menentukan ukuran window dan title untuk menampilkan

ex.setGeometry(100,100,500,300)

#membuat judul window

ex.setWindowTitle("Simulasi Tabungan Berjangka")

#menampilan isi dari variabel ex

ex.show()

#membuat system exit

#sys.exit(app.exec\_())

#membuat fungsi on\_click yang berisi sistem untuk menghitung nilai Tabungan Berjangka

def on\_click():

#membuat variabel Awal, Rutin, Waktu yang berisi inputan dari volume1 dan waktud1 yang akan dirubah menjadi sebuah float

Awal = float(self.awal.text())

Rutin = float(self.rutin.text())

Waktu = float(self.waktu.text())

TotalWaktu = (Waktu \* 30)

Bunga = (float(self.bunga.text()) / 100)

print (Bunga)

Pajak = (float(self.pajak.text()) / 100)

#membuat variabel untuk menghitung nilai tabungan berjangka

TotalRutin = Rutin \* Waktu

Profit = TotalRutin \* Bunga \* TotalWaktu

TotalProfit = Profit / 365

TotalPajak = Pajak \* TotalProfit

TotalAkhir = Awal + TotalRutin + (TotalProfit - TotalPajak)

print(TotalProfit)

#menampilkan nilai tabungan berjangka pada button hasil

self.hasil.setText("Nilai Tabungan Ketika Jatuh Tempo: Rp." + str("%.2f" %TotalAkhir))

if \_\_name\_\_ == '\_\_main\_\_':

#memanggil fungsi Layout untuk menampilkan seluruh isi widget

Layout()

**simulasi\_peminjaman.py**

import sys

from PyQt5.QtWidgets import \*

from PyQt5.Qt import \*

#GLOBAL

#Inisisalisai pyqt

app = QApplication(sys.argv)

#mengatur style di window menjadi style fusion

app.setStyle("fusion")

#membuat variabel ex yang berisi fungsi QWidget

ex = QWidget()

#variabel self berisi nilai ex yang berarti berisi QWidget

self = ex

#membuat QLineEdit dengan nama variabel pinjaman, waktu, bunga

self.pinjaman = QLineEdit(self)

self.waktu = QLineEdit(self)

self.bunga = QLineEdit(self)

#membuat sebuah button dengan nama variable cal\_button

self.cal\_button = QPushButton("Calculate", self)

#membuat sebuah button dengan nama variable hasil

self.hasil = QPushButton("Hasil", self)

#membuat fungsi dengan nama Layout

def Layout():

#membuat grid layout dengan nama variabel grid

grid = QGridLayout()

#mengatur batas pada isi conten layout grid yaitu batas kana,kiri,atas,bawah bernilai 10

grid.setContentsMargins(10,10,10,10)

#membuat sebuag QTabWidget dengan nama variabel tab

tab = QTabWidget(self)

#mengatur semua isi konten yang ada di tab menjadi tulisan bold berwarna hitam dengan ukuran 15px

tab.setStyleSheet("font: bold 15px; color: black;")

#membuat QVBoxLayout dengan nama variabel vbox

vbox = QVBoxLayout()

#membuat groupbox

group = QGroupBox(self)

#membuat vbox menjadi layout

group.setLayout(vbox)

#memasukkan group kedalam variabel tab atau QTabWidget

tab.addTab(group,"Simulasi Peminjaman")

#memasukkan tab kedalam layout utama grid

grid.addWidget(tab, 2, 1)

#------------------------- KELOMPOK 1 -------------------------#

#membuat label1 dan akan dimasukkan ke dalam layout vbox

label1 = QLabel(self)

label1.setText("Masukkan Jumlah Peminjaman:")

vbox.addWidget(label1)

#mengatur pinjaman dengan border 0.5 px dan berwarna solid black

self.pinjaman.setStyleSheet("border: 0.5px solid black;")

#memasukkan volume1 kedalam layout vbox

vbox.addWidget(self.pinjaman)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch()

#membuat label1 dan akan dimasukkan ke dalam layout vbox

label1 = QLabel(self)

label1.setText("Masukkan Jangka Waktu Peminjaman (Dalam Bulan):")

vbox.addWidget(label1)

#mengatur waktu dengan border 0.5 px dan berwarna solid black

self.waktu.setStyleSheet("border: 0.5px solid black;")

#memasukkan waktu kedalam layout vbox

vbox.addWidget(self.waktu)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch()

#membuat label1 dan akan dimasukkan ke dalam layout vbox

label1 = QLabel(self)

label1.setText("Masukkan Bunga Peminjaman:")

vbox.addWidget(label1)

#mengatur bunga dengan border 0.5 px dan berwarna solid black

self.bunga.setStyleSheet("border: 0.5px solid black;")

#memasukkan bunga kedalam layout vbox

vbox.addWidget(self.bunga)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch()

#mengatur cal\_button dengan background berwarna #33f8c7

self.cal\_button.setStyleSheet("background-color: #33f8c7;")

#memasukkan cal\_button kedalam layout vbox

vbox.addWidget(self.cal\_button)

#menambahkan addStretch ke dalam vbox agar tertata rapi

vbox.addStretch(1)

##mengatur hasil dengan font bold dan ukuran 20px dan berwarna hitam

self.hasil.setStyleSheet("font: bold 20px; color: black")

#mengatur button hasil dengan setDisabled agar tidak daat diklik

self.hasil.setDisabled(True)

#memasukkan button hasil ke dalam layout vbox

vbox.addWidget(self.hasil)

#menambahkan addStretch ke dalam vbox4 agar tertata rapi

vbox.addStretch(1)

#connect button to fungsi on\_click

self.cal\_button.clicked.connect(on\_click)

#------------------------- END OF KELOMPOK 1 -------------------------#

#membuat grid menjadi layout utama di window

ex.setLayout(grid)

#Menentukan ukuran window dan title untuk menampilkan

ex.setGeometry(100,100,500,300)

#membuat judul window

ex.setWindowTitle("Simulasi Peminjaman")

#menampilan isi dari variabel ex

ex.show()

#membuat system exit

sys.exit(app.exec\_())

#membuat fungsi on\_click yang berisi sistem untuk menghitung nilai Simulasi Pemnjaman

def on\_click():

#membuat variabel Pinjaman, Waktu, Bunga yang berisi inputan dari pinjaman, waktu, bunga yang akan dirubah menjadi sebuah float

Pinjaman = float(self.pinjaman.text())

Waktu = float(self.waktu.text())

Bunga = (float(self.bunga.text()) / 100)

Total = Pinjaman + (Pinjaman + Waktu + Bunga)

#menampilkan nilai simulasi peminjaman pada button hasil

self.hasil.setText("Nilai Peminjaman yang harus di bayarkan: Rp." + str("%.2f" %Total))

if \_\_name\_\_ == '\_\_main\_\_':

#memanggil fungsi Layout untuk menampilkan seluruh isi widget

Layout()

1. **Penjelasan Kode Program**

* *class Ui\_Dialog(object):*

*def setupUi(self, Dialog):*

*Dialog.setObjectName("Dialog")*

*Dialog.resize(395, 271)*

*Dialog.setInputMethodHints(QtCore.Qt.ImhNone)*

Membuat sebuah class dialog dan sebuah fungsi yang berisi beberapa widgets didalamnya

* *self.label = QtWidgets.QLabel(Dialog)*

*self.label.setGeometry(QtCore.QRect(100, 20, 201, 41))*

*font = QtGui.QFont()*

*font.setFamily("Stencil")*

*font.setPointSize(18)*

*font.setBold(True)*

*font.setWeight(75)*

*self.label.setFont(font)*

*self.label.setObjectName("label")*

*self.label\_2 = QtWidgets.QLabel(Dialog)*

*self.label\_2.setGeometry(QtCore.QRect(10, 90, 131, 31))*

*font = QtGui.QFont()*

*font.setFamily("Arial Black")*

*font.setPointSize(12)*

*font.setBold(True)*

*font.setWeight(75)*

*self.label\_2.setFont(font)*

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

*self.label\_3 = QtWidgets.QLabel(Dialog)*

*self.label\_3.setGeometry(QtCore.QRect(10, 150, 131, 31))*

*font = QtGui.QFont()*

*font.setFamily("Arial Black")*

*font.setPointSize(12)*

*font.setBold(True)*

*font.setWeight(75)*

*self.label\_3.setFont(font)*

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

*self.username = QtWidgets.QLineEdit(Dialog)*

*self.username.setGeometry(QtCore.QRect(150, 90, 221, 31))*

*self.username.setObjectName("username")*

*self.password = QtWidgets.QLineEdit(Dialog)*

*self.password.setGeometry(QtCore.QRect(150, 150, 221, 31))*

*self.password.setObjectName("password")*

*self.button\_login = QtWidgets.QPushButton(Dialog)*

*self.button\_login.setGeometry(QtCore.QRect(140, 210, 111, 41))*

*font = QtGui.QFont()*

*font.setPointSize(12)*

*font.setBold(True)*

*font.setWeight(75)*

*self.button\_login.setFont(font)*

*self.button\_login.setObjectName("button\_login")*

Membuat sebuah line edit dan beberapa label serta push button, sekaligus mengatur posisinya.

* *self.button\_login.clicked.connect(self.check\_login)*

*self.retranslateUi(Dialog)*

*QtCore.QMetaObject.connectSlotsByName(Dialog)*

Membuat sebuah signal pada push button dan yang akan terhubung ke fungsi check\_login.

* *def check\_login(self):*

*notif = QMessageBox()*

*username = str(self.username.text())*

*password = str(self.password.text())*

*if username == 'apak' and password == '123':*

*notif.setText("Selamat anda berhasil login!")*

*notif.exec()*

*Dialog.close()*

*self.MainWindow = QtWidgets.QMainWindow()*

*self.ui = Ui\_MainWindow()*

*self.ui.setupUi(self.MainWindow)*

*self.MainWindow.show()*

*else:*

*notif.setText("Anda gagal login!")*

*notif.exec()*

Membuat sebuah slot berupa fungsi yang menerima signal daributton yang sudah klik, dan ada sebuah validasi ketika inputan username dan password sudah benar, maka akan muncul notif sebagai QMessageBox, kemudian akan menutup dialog dan beralih ke file mainwindow. Sementara jika username dan password salah makan akan muncul notifikasi dan tidak akan beralih ke file mainwindow.

* *def retranslateUi(self, Dialog):*

*\_translate = QtCore.QCoreApplication.translate*

*Dialog.setWindowTitle(\_translate("Dialog", "User Login"))*

*self.label.setText(\_translate("Dialog", "USER LOGIN!"))*

*self.label\_2.setText(\_translate("Dialog", "USERNAME"))*

*self.label\_3.setText(\_translate("Dialog", "PASSWORD"))*

*self.button\_login.setText(\_translate("Dialog", "LOGIN"))*

Menamai kemebali tampilan yang ada pada masing – masing widget sesuai dengan keinginan kita.

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

*import sys*

*app = QtWidgets.QApplication(sys.argv)*

*Dialog = QtWidgets.QDialog()*

*ui = Ui\_Dialog()*

*ui.setupUi(Dialog)*

*Dialog.show()*

*sys.exit(app.exec\_())*

Mendeklarasikan QApplication di dalam variable app, memasukkan value yang ada di class Ui\_Form ke dalam variable ui. Kemudian menampilkan variable tersebut dengan fungsi show(), dan membuat system exit.

* *class Ui\_MainWindow(object):*

*def setupUi(self, MainWindow):*

*MainWindow.setObjectName("MainWindow")*

*MainWindow.resize(800, 600)*

*self.centralwidget = QtWidgets.QWidget(MainWindow)*

*self.centralwidget.setObjectName("centralwidget")*

Membuat sebuah class mainwindow dan sebuah fungsi yang berisi beberapa widgets didalamnya

* self.label = QtWidgets.QLabel(self.centralwidget)

*self.label.setGeometry(QtCore.QRect(170, 200, 461, 151))*

*font = QtGui.QFont()*

*font.setFamily("Arial Black")*

*font.setPointSize(14)*

*font.setBold(True)*

*font.setWeight(75)*

*self.label.setFont(font)*

*self.label.setObjectName("label")*

*MainWindow.setCentralWidget(self.centralwidget)*

*self.menubar = QtWidgets.QMenuBar(MainWindow)*

*self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 26))*

*self.menubar.setObjectName("menubar")*

*self.menuMenu = QtWidgets.QMenu(self.menubar)*

*self.menuMenu.setObjectName("menuMenu")*

*MainWindow.setMenuBar(self.menubar)*

*self.statusbar = QtWidgets.QStatusBar(MainWindow)*

*self.statusbar.setObjectName("statusbar")*

*MainWindow.setStatusBar(self.statusbar)*

*self.actionSimulasi\_Tabungan\_Berjangka = QtWidgets.QAction(MainWindow)*

*self.actionSimulasi\_Tabungan\_Berjangka.setObjectName("actionSimulasi\_Tabungan\_Berjangka")*

*self.actionSimulasi\_Anggaran\_Bulanan = QtWidgets.QAction(MainWindow)*

*self.actionSimulasi\_Anggaran\_Bulanan.setObjectName("actionSimulasi\_Anggaran\_Bulanan")*

*self.actionSimulasi\_Peminjaman = QtWidgets.QAction(MainWindow)*

*self.actionSimulasi\_Peminjaman.setObjectName("actionSimulasi\_Peminjaman")*

*self.menuMenu.addAction(self.actionSimulasi\_Tabungan\_Berjangka)*

*self.menuMenu.addAction(self.actionSimulasi\_Anggaran\_Bulanan)*

*self.menuMenu.addAction(self.actionSimulasi\_Peminjaman)*

*self.menubar.addAction(self.menuMenu.menuAction())*

Membuat sebuah label serta menubar yang bersisi beberapa QAction, sekaligus mengatur posisi label.

* *self.retranslateUi(MainWindow)*

*QtCore.QMetaObject.connectSlotsByName(MainWindow)*

*self.actionSimulasi\_Tabungan\_Berjangka.triggered.connect(Simulasi\_Tabungan\_Berjangka)*

*self.actionSimulasi\_Anggaran\_Bulanan.triggered.connect(Simulasi\_Anggaran\_Bulanan)*

*self.actionSimulasi\_Peminjaman.triggered.connect(Simulasi\_Peminjaman)*

Membuat sebuah signal pada menubar QAction dan akan terhubung ke fungsi masing – masing.

* *def retranslateUi(self, MainWindow):*

*\_translate = QtCore.QCoreApplication.translate*

*MainWindow.setWindowTitle(\_translate("MainWindow", "MainWindow"))*

*self.label.setText(\_translate("MainWindow", "ANDA MASUK KE MAIN WINDOW"))*

*self.menuMenu.setTitle(\_translate("MainWindow", "Menu"))*

*self.actionSimulasi\_Tabungan\_Berjangka.setText(\_translate("MainWindow", "Simulasi Tabungan Berjangka"))*

*self.actionSimulasi\_Anggaran\_Bulanan.setText(\_translate("MainWindow", "Simulasi Anggaran Bulanan"))*

*self.actionSimulasi\_Peminjaman.setText(\_translate("MainWindow", "Simulasi Peminjaman"))*

Menamai kemebali tampilan yang ada pada masing – masing widget sesuai dengan keinginan kita.

* *#membuat fungsi Simulasi\_Tabungan\_Berjangka*

*def Simulasi\_Tabungan\_Berjangka():*

*#memanggil/menampilkan aplikasi Simulasi\_Tabungan\_Berjangka*

*tabungan\_berjangka.Layout()*

*#membuat fungsi Simulasi\_Anggaran\_Bulanan*

*def Simulasi\_Anggaran\_Bulanan():*

*#memanggil/menampilkan aplikasi Simulasi\_Anggaran\_Bulanan*

*anggaran\_bulanan.Layout()*

*#membuat fungsi Simulasi\_Peminjaman*

*def Simulasi\_Peminjaman():*

*#memanggil/menampilkan aplikasi Simulasi\_Peminjaman*

*simulasi\_peminjaman.Layout()*

Membuat sebuah slot berupa fungsi yang menerima signal dari QAction yang sudah klik, dan akan ketika fungsi sudah di jalankan maka akan beralih ke file lain seperti file simulasi\_peminjaman, dll.

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

*import sys*

*app = QtWidgets.QApplication(sys.argv)*

*MainWindow = QtWidgets.QMainWindow()*

*ui = Ui\_MainWindow()*

*ui.setupUi(MainWindow)*

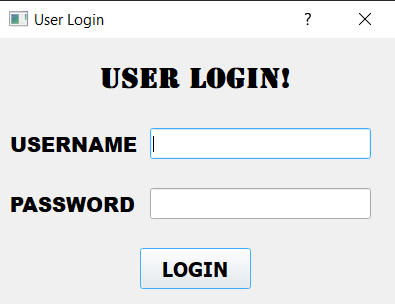
*MainWindow.show()*

*sys.exit(app.exec\_())*

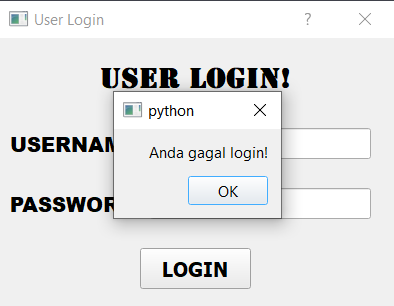
Mendeklarasikan QApplication di dalam variable app, memasukkan value yang ada di class Ui\_Form ke dalam variable ui. Kemudian menampilkan variable tersebut dengan fungsi show(), dan membuat system exit.

1. **Hasil Running Program**

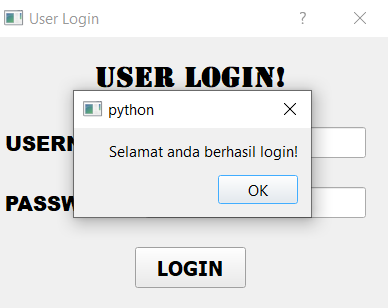
Tampilan pada window login.py

****

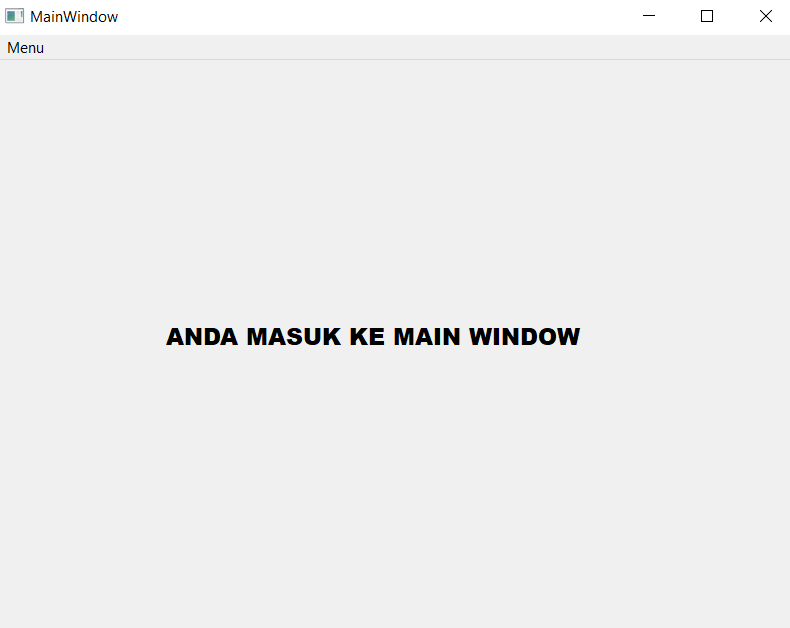
Jika username dan password salah, maka akan menampilkan window seperti gambar berikut.

****

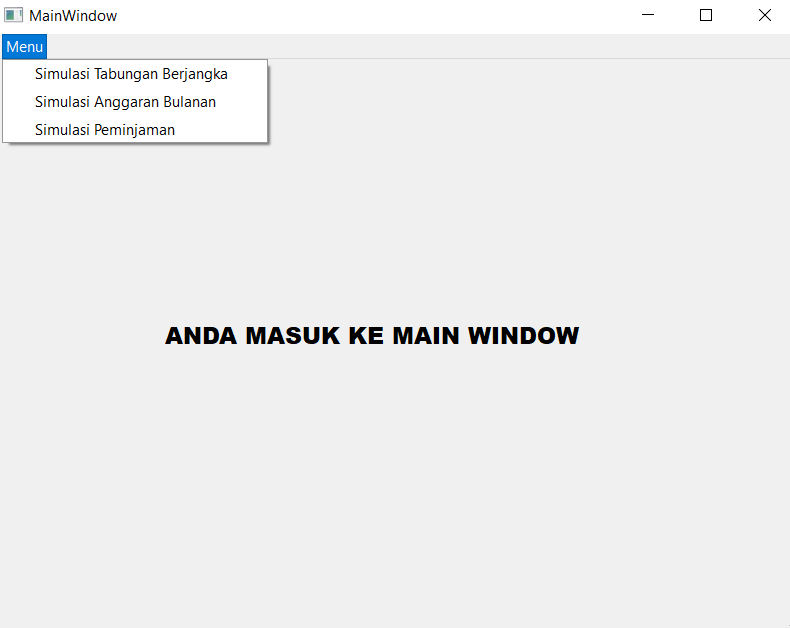
Jika username dan password benar, maka akan menampilkan window seperti gambar berikut.

****

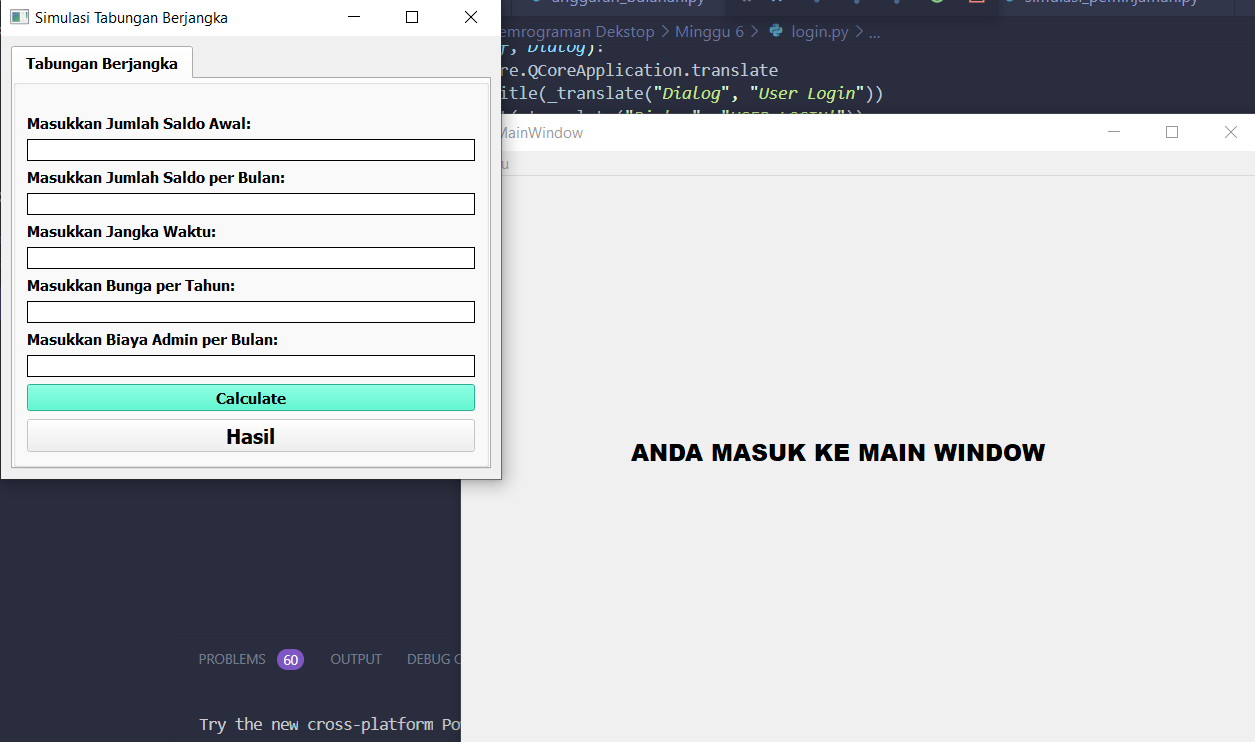
Setelah klik ok, maka akan dialihkan ke window mainwindow.py dengan tampilan seperti gambar berikut.

****

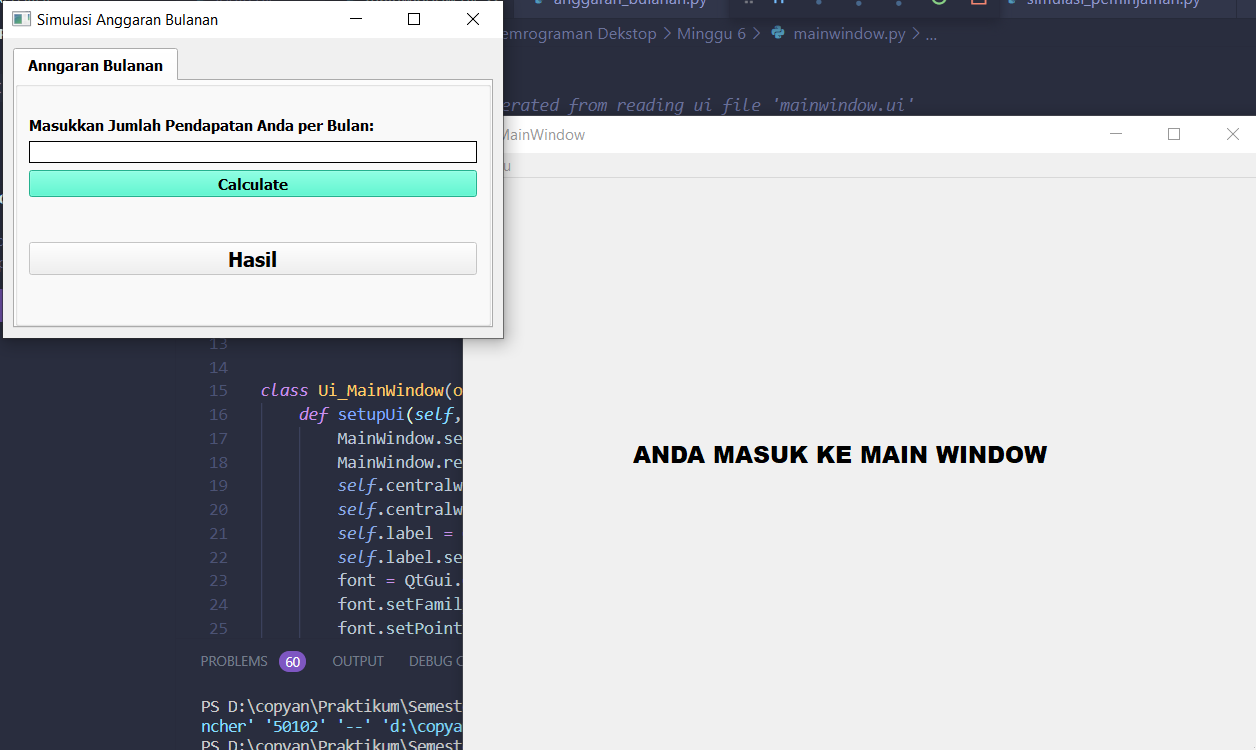
Pada mainwindow.py terdapat menu bar yang terdiri dari beberapa QAction dengan tampilan seperti gambar berikut.

****

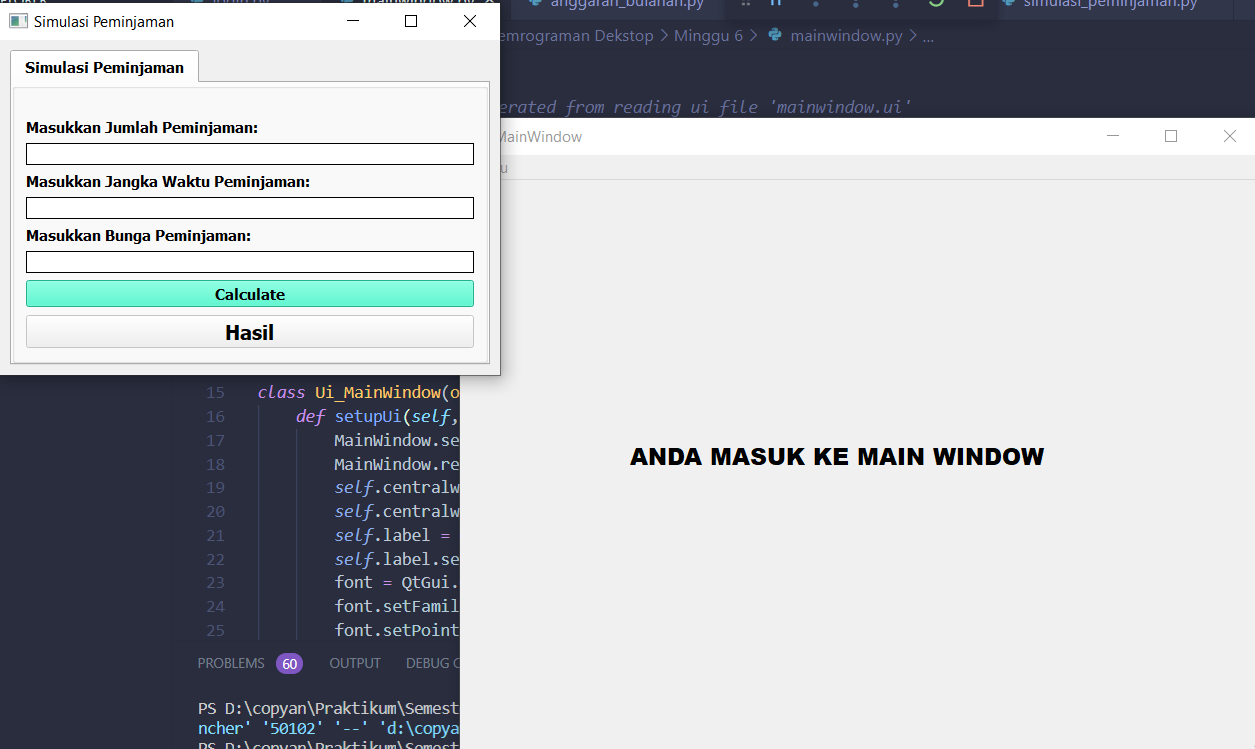
Masing – masing QAction tersebut jika di klik akan membuka file yang sudah disiapkan, seperti gambar berikut setelah mengklik QAction simulasi tabungan berjangka.

****

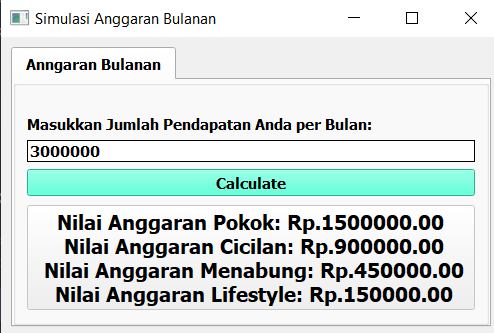
Setelah mengklik QAction simulasi anggaran bulanan.

****

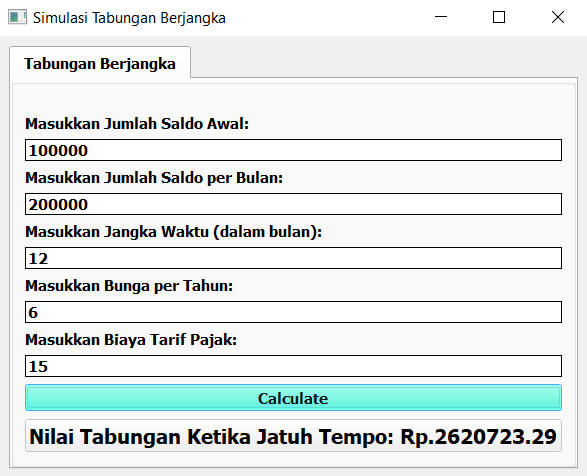
Setelah mengklik QAction simulasi peminjaman.

****

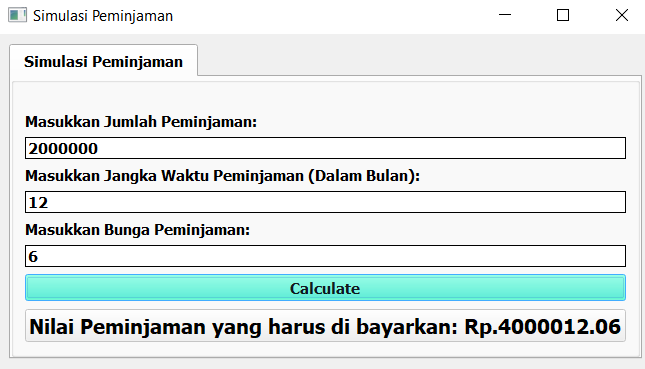
Simulasi Anggaran Bulanan

****

Simulasi Tabungan Berjangka



Simulasi Peminjaman



**BAB II**

**PENUTUP**

1. **Kesimpulan**
2. Kita bisa mendapatkan value dari inputan user menggunakan line edit.
3. Dengan adanya signal dan slot dapat membuat aplikasi yang di buat berfungsi lebih baik.
4. Dengan menggunakan menu bar, user menjadi terbantu dengan fitur – fitur yang dimiliki aplikasi
5. **Saran**

Banyak mencoba dan mengekplorasi widget yang lain agar lebih paham