



**Dosen Pengampu :**  
Freddy Wicaksono, M.Kom

# LAPORAN PRAKTIKUM PBO

**Aplikasi Perhitungan Dengan Metode :**  
PyQt5

---

**Disusun Oleh : (TIF22D / R4)**

Nova Subhan (220511170)  
Rahma Syifatun Nabilah (221511013)  
Raisya Farosyatul Ilmi (220511131)

## **KATA PENGANTAR**

Segala puji bagi Allah SWT yang telah memberikan kami kemudahan sehingga kami dapat menyelesaikan laporan ini dengan tepat waktu. Tanpa pertolongan-Nya tentunya kami tidak akan sanggup untuk menyelesaikan laporan ini dengan baik. Shalawat serta salam semoga terlimpah curahkan kepada baginda tercinta kita yaitu Nabi Muhammad SAW, yang kita nanti-nantikan syafa'atnya di akhirat nanti.

Laporan ini membahas implementasi PBO dalam konteks pengembangan aplikasi perhitungan menggunakan PyQt. Kami memandang PBO sebagai pilar utama dalam merancang aplikasi yang tidak hanya efektif secara fungsional tetapi juga mudah dipahami dan dikembangkan.

Penulis tentu menyadari bahwa laporan ini masih jauh dari kata sempurna dan masih banyak terdapat kesalahan serta kekurangan dalam penulisan ini. Untuk itu penulis berharap teman-teman semua bisa mengerti. Semoga makalah ini memberikan banyak manfaat bagi kita semua tentang pentingnya PBO dalam konteks pengembangan aplikasi perhitungan menggunakan PyQt.

Cirebon, 13 November 2023

Penulis

## DAFTAR ISI

	Hal
KATA PENGANTAR .....	ii
DAFTAR ISI .....	iii
BAB I    PENDAHULUAN .....	1
1.1   Latar Belakang.....	1
1.2   Rumusan Masalah.....	1
1.3   Tujuan.....	1
BAB II   PEMBAHASAN .....	2
2.1   Pengertian PyQt5 .....	2
2.2   Fungsi PyQt5.....	3
2.3   Kelebihan & Kekurangan PyQt5 .....	4
2.4   Instalasi PyQt5 .....	5
2.5   Tampilan Qt Designer .....	5
2.6   Source Code .....	6
2.7   Hasil Akhir .....	13
BAB III   PENUTUP .....	14
3.1   Kesimpulan .....	14
DAFTAR PUSTAKA .....	15

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

PyQt5 merupakan salah satu toolkit pengembangan GUI (Graphical User Interface) yang dibangun di atas Qt, sebuah toolkit dan framework pengembangan aplikasi lintas platform. PyQt5 memberikan akses Python ke fungsi-fungsi Qt, memungkinkan pengembang untuk membuat aplikasi dengan antarmuka pengguna yang menarik dan responsif.

Dengan menggunakan PyQt5, pengembang dapat membuat aplikasi dengan antarmuka pengguna yang kaya fitur dan responsif, serta mendukung pengembangan lintas platform yang efisien.

### **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, timbul beberapa masalah, diantaranya :

1. Pengertian PyQt5
2. Fungsi PyQt5
3. Membuat Aplikasi Perhitungan Menggunakan Metode PyQt5

### **1.3 Tujuan**

Berdasarkan rumusan masalah di atas, tujuan dari pembahasan makalah ini adalah :

1. Mengetahui Tentang PyQt5
2. Mengetahui Fungsi PyQt5
3. Mengetahui cara membuat Aplikasi Perhitungan Menggunakan Metode PyQt5

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Pengertian PyQt5**

PyQt adalah kerangka kerja GUI Python yang dibangun di sekitar kerangka kerja C++ Qt, yang dikembangkan dan dikelola oleh Perusahaan Qt. Ini memungkinkan kami membuat antarmuka modern yang terlihat nyaman di platform apa pun, termasuk Windows, macOS, Linux, dan bahkan Android. Ia juga memiliki perkakas yang solid, yang paling terkenal adalah Qt Creator , yang mencakup editor WYSIWYG untuk merancang antarmuka GUI dengan cepat dan mudah.

PyQt dikembangkan dan dikelola oleh Riverbank Computing dan pertama kali dirilis pada tahun 1998, empat tahun setelah Tkinter.

Versi PyQt yang dapat digunakan secara gratis dilisensikan di bawah GNU General Public License (GPL) v3 . Artinya, PyQt terbatas pada aplikasi berlisensi GPL kecuali Anda membeli lisensi komersialnya.

Perusahaan Qt memiliki pengikatan Python sendiri untuk Qt, yang disebut PySide. Library ini dirilis pada tahun 2009. Perbedaan utama antara PyQt dan PySide adalah pada perizinannya. PySide dilisensikan di bawah GNU Lesser General Public License (LGPL), yang berarti Anda menggunakan PySide dalam aplikasi non-GPL tanpa biaya tambahan apa pun.

Qt, dan dengan ekstensi PyQt, bukan hanya perpustakaan GUI. Ini adalah kerangka pengembangan aplikasi GUI yang lengkap. Selain komponen GUI standar, seperti widget dan tata letak, Qt menyediakan:

- Tampilan berbasis data MVC (model-view-controller) (spreadsheet, tabel)
- Antarmuka dan model basis data
- Plot grafik
- Visualisasi grafis vektor
- Pemutaran multimedia
- Efek suara dan daftar putar
- Antarmuka untuk perangkat keras, seperti perangkat pencetakan

Mekanisme sinyal dan slot Qt untuk pemrograman acara memungkinkan kami merancang aplikasi kompleks dengan tepat dari komponen yang dapat digunakan kembali dan diisolasi.

## 2.2 Fungsi PyQt5

PyQt paling sering digunakan untuk, dan sangat cocok untuk, membangun aplikasi desktop GUI berfitur lengkap. Seperti yang telah Anda pelajari, PyQt mendukung tampilan berbasis data (seperti MVC), grafik vektor, animasi & transisi, database, dan threading/konkurensi.

Qt Designer, pembuat GUI yang disediakan oleh Qt, memungkinkan Anda membuat perangkat lunak berkualitas profesional dalam waktu singkat. Mekanisme sinyal dan slot memungkinkan pemisahan komponen aplikasi dengan benar, memungkinkan arsitektur sistem yang kuat dan dapat dipelihara.

Anda juga dapat menggunakan PyQt untuk membuat antarmuka pengguna layar sentuh untuk perangkat keras yang didukung Raspberry Pi -- keduanya menggunakan Qt Widgets dan QML/Qt Quick API. Meskipun PyQt secara teknis dapat digunakan untuk membuat aplikasi untuk perangkat seluler, jenis aplikasi ini jarang terlihat di luar bidang penghobi.

## 2.3 Kelebihan & Kekurangan PyQt5

Ada sejumlah alasan Anda mungkin ingin memilih PyQt untuk proyek Anda dan alasan Anda mungkin tidak memilihnya. Mari kita lihat beberapa pro dan kontra menggunakan PyQt untuk membangun aplikasi GUI dengan Python.

Untuk memulai, mari kita mulai dengan kelebihannya:

- Ini adalah kerangka GUI yang kuat & modern yang cocok untuk membangun aplikasi profesional.
- Ini mencakup beberapa widget tingkat lanjut, termasuk tampilan berdasarkan data, bagan, antarmuka basis data, kanvas grafik vektor, pemutaran video, dan komponen browser web yang berfungsi penuh.
- Itu dapat memanfaatkan Qt Designer, yang memungkinkan Anda mendesain GUI menggunakan editor drag-and-drop grafis.
- Ini lintas platform dan dapat berjalan di Windows, Linux, macOS, dan perangkat seluler.
- Ini menyediakan komponen GUI yang modern dan tampak asli di semua platform utama. Komponen-komponen ini sebagian besar dapat disesuaikan jika diperlukan.
- Ini adalah perpustakaan yang dilengkapi baterai, yang berarti Anda dapat menyelesaikan banyak hal dengan PyQt secara langsung. Karakteristik ini berarti lebih sedikit kebutuhan akan ketergantungan pihak ketiga.
- Ini memiliki banyak dukungan dan sumber belajar online, termasuk tutorial lengkap PyQt .
- Ini memungkinkan pembuatan antarmuka layar sentuh dengan QML/Qt Quick API.

Meskipun PyQt memiliki banyak fitur dan kelebihan yang menarik, PyQt juga memiliki beberapa kekurangan:

- Ini bisa jadi menakutkan bagi pemula. Ukuran perpustakaan dan rangkaian fiturnya yang kompleks dapat membuatnya kewalahan.
- Dokumentasi Pythonnya buruk dan tidak lengkap . Sebagai alternatif, Anda dapat menggunakan dokumentasi resmi Qt. Namun, dokumentasi ini ditujukan untuk pustaka C++ dan mungkin sulit untuk diterjemahkan.
- Perlu waktu untuk mempelajari sepenuhnya kerangka kerja dan cara kerjanya.
- Ini memungkinkan untuk lisensi GPL atau komersial saja.

Hingga saat ini, Anda telah belajar banyak tentang Tkinter dan PyQt. Untuk membuat hidup Anda lebih menyenangkan, bagian berikut memberikan perbandingan singkat dan ringkas antara kedua alat GUI ini.

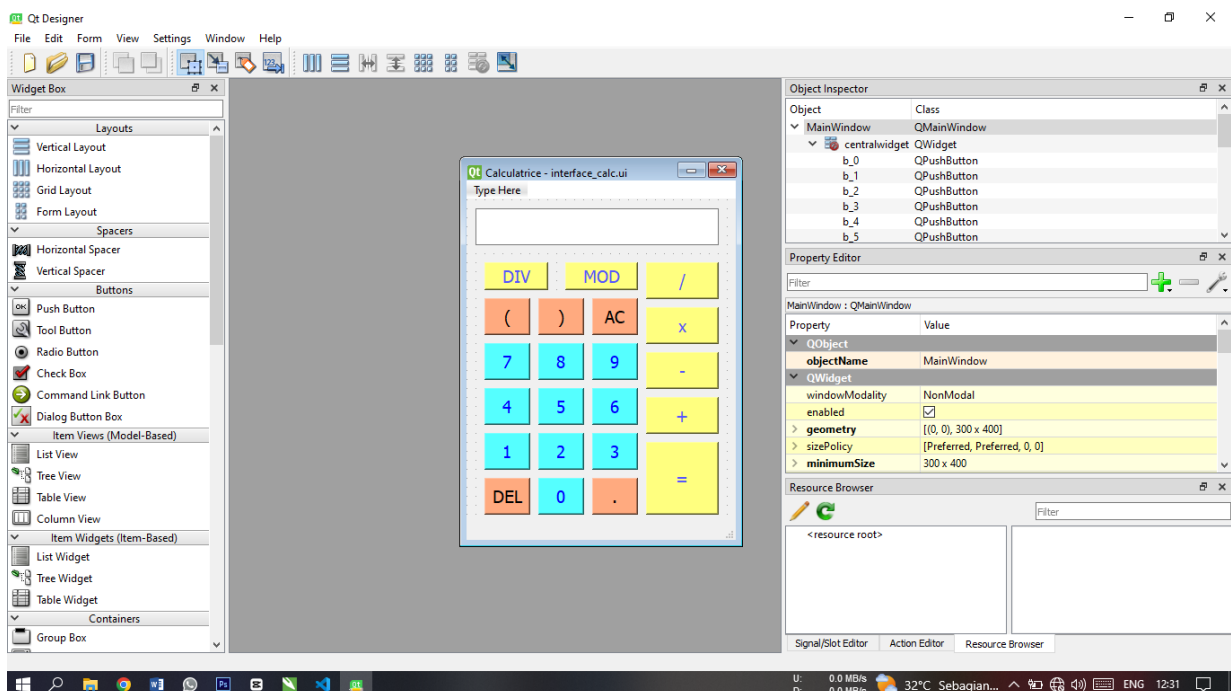
## 2.4 Instalasi PyQt5

### *Cara Install PyQt5 Melalui CMD*

Tampilan awal program, akan menampilkan tabel sesuai perintah program yang sudah dibuat/ dimasukkan. Setelah muncul tabel seperti dibawah, kita cek kembali di visual codenya. Ketikkan “ pip install PyQt5 ”

```
ERROR: No matching distribution found for PyQt5.uic
D:\Kampus\PBO\Pertemuan 3\Kalkulator>pip install PyQt5
Collecting PyQt5
  Downloading PyQt5-5.15.10-cp37-abi3-win_amd64.whl.metadata (2.2 kB)
Requirement already satisfied: PyQt5-sip<13,>=12.13 in c:\users\user\appdata\local\programs\python\python312\lib\site-packages (from PyQt5) (12.13.0)
Collecting PyQt5-Qt5>=5.15.2 (from PyQt5)
  Downloading PyQt5_Qt5-5.15.2-py3-none-win_amd64.whl (50.1 MB)
----- 50.1/50.1 MB 2.6 MB/s eta 0:00:00
Downloading PyQt5-5.15.10-cp37-abi3-win_amd64.whl (6.8 MB)
----- 6.8/6.8 MB 3.3 MB/s eta 0:00:00
Installing collected packages: PyQt5-Qt5, PyQt5
Successfully installed PyQt5-5.15.10 PyQt5-Qt5-5.15.2
D:\Kampus\PBO\Pertemuan 3\Kalkulator>
```

## 2.5 Tampilan Qt Designer





Untuk Instalasi PyQt5 di Visual code dapat mengeksekusi perintah berikut:

### ***Source Code (calculator.py)***

```
from PyQt5 import QtCore, QtGui, QtWidgets
from interface import Ui_MainWindow
ok=True
#Slots

def affiche1():
    global ok
    if(ok==True):
        ui.lineEdit.setText(ui.lineEdit.text()+"1")
    else:
        ui.lineEdit.setText("1")
        ok=True
def affiche2():
    global ok
    if(ok==True):
        ui.lineEdit.setText(ui.lineEdit.text()+"2")
    else:
        ui.lineEdit.setText("2")
        ok=True
def affiche3():
    global ok
    if(ok==True):
        ui.lineEdit.setText(ui.lineEdit.text()+"3")
    else:
        ui.lineEdit.setText("3")
        ok=True
def affiche4():
    global ok
    if(ok==True):
        ui.lineEdit.setText(ui.lineEdit.text()+"4")
    else:
        ui.lineEdit.setText("4")
        ok=True
def affiche5():
    global ok
    if(ok==True):
        ui.lineEdit.setText(ui.lineEdit.text()+"5")
    else:
        ui.lineEdit.setText("5")
        ok=True
def affiche6():
    global ok
    if(ok==True):
        ui.lineEdit.setText(ui.lineEdit.text()+"6")
    else:
```

```

        ui.lineEdit.setText("6")
        ok=True
def affiche7():
    global ok
    if(ok==True):
        ui.lineEdit.setText(ui.lineEdit.text()+"7")
    else:
        ui.lineEdit.setText("7")
        ok=True
def affiche8():
    global ok
    if(ok==True):
        ui.lineEdit.setText(ui.lineEdit.text()+"8")
    else:
        ui.lineEdit.setText("8")
        ok=True
def affiche9():
    global ok
    if(ok==True):
        ui.lineEdit.setText(ui.lineEdit.text()+"9")
    else:
        ui.lineEdit.setText("9")
        ok=True
def affiche0():
    global ok
    if(ok==True):
        ui.lineEdit.setText(ui.lineEdit.text()+"0")
    else:
        ui.lineEdit.setText("0")
        ok=True
def affichePoint():
    ui.lineEdit.setText(ui.lineEdit.text()+".")
def afficheDivise():
    ui.lineEdit.setText(ui.lineEdit.text()+"/")
def afficheMult():
    ui.lineEdit.setText(ui.lineEdit.text()+"*")
def afficheMoins():
    ui.lineEdit.setText(ui.lineEdit.text()+"-")
def affichePlus():
    ui.lineEdit.setText(ui.lineEdit.text()+"+")
def afficheDiv():
    ui.lineEdit.setText(ui.lineEdit.text()+"//")
def afficheMod():
    ui.lineEdit.setText(ui.lineEdit.text()+"%")
def affichePo():
    ui.lineEdit.setText(ui.lineEdit.text()+"(")
def affichePf():
    ui.lineEdit.setText(ui.lineEdit.text()+")")

```

```

def afficheAc():
    ui.lineEdit.setText("")
def afficheDel():
    ui.lineEdit.setText(ui.lineEdit.text()[:-1])
def afficheEgale():
    global ok
    ui.lineEdit.setText(str(eval(ui.lineEdit.text())))
    ok=False
import sys
app = QtWidgets.QApplication(sys.argv)
MainWindow = QtWidgets.QMainWindow()
ui = Ui_MainWindow()
ui.setupUi(MainWindow)
MainWindow.show()

#Connections
ui.b_del.clicked.connect(afficheDel)
ui.b_ac.clicked.connect(afficheAc)
ui.b_egale.clicked.connect(afficheEgale)
ui.b_1.clicked.connect(affiche1)
ui.b_2.clicked.connect(affiche2)
ui.b_3.clicked.connect(affiche3)
ui.b_4.clicked.connect(affiche4)
ui.b_5.clicked.connect(affiche5)
ui.b_6.clicked.connect(affiche6)
ui.b_7.clicked.connect(affiche7)
ui.b_8.clicked.connect(affiche8)
ui.b_9.clicked.connect(affiche9)
ui.b_0.clicked.connect(affiche0)
ui.b_plus.clicked.connect(affichePlus)
ui.b_moins.clicked.connect(afficheMoins)
ui.b_mult.clicked.connect(afficheMult)
ui.b_divise.clicked.connect(afficheDivise)
ui.b_div.clicked.connect(afficheDiv)
ui.b_mod.clicked.connect(afficheMod)
ui.b_point.clicked.connect(affichePoint)
ui.b_po.clicked.connect(affichePo)
ui.b_pf.clicked.connect(affichePf)

sys.exit(app.exec_())

```

## Source Code (Interface.py)

```
# Form implementation generated from reading ui file 'interface_calc.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

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(300, 400)
        MainWindow.setMinimumSize(QtCore.QSize(300, 400))
        MainWindow.setMaximumSize(QtCore.QSize(300, 400))
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.lineEdit = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit.setGeometry(QtCore.QRect(10, 10, 271, 41))
        self.lineEdit.setStyleSheet("background-color: rgb(255, 255, 255);\n"
"color: rgb(255, 0, 255);\n"
"font: 16pt \"MS Shell Dlg 2\";\n"
"border-color: rgb(0, 0, 0);")
        self.lineEdit.setReadOnly(True)
        self.lineEdit.setObjectName("lineEdit")
        self.b_8 = QtWidgets.QPushButton(self.centralwidget)
        self.b_8.setGeometry(QtCore.QRect(80, 160, 51, 41))
        self.b_8.setStyleSheet("color: rgb(0, 0, 255);\n"
"background-color: rgb(85, 255, 255);\n"
"font: 14pt \"MS Shell Dlg 2\";")
        self.b_8.setObjectName("b_8")
        self.b_pf = QtWidgets.QPushButton(self.centralwidget)
        self.b_pf.setGeometry(QtCore.QRect(80, 110, 51, 41))
        self.b_pf.setStyleSheet("background-color: rgb(255, 170, 127);\n"
"font: 14pt \"MS Shell Dlg 2\";")
        self.b_pf.setObjectName("b_pf")
        self.b_7 = QtWidgets.QPushButton(self.centralwidget)
        self.b_7.setGeometry(QtCore.QRect(20, 160, 51, 41))
        self.b_7.setStyleSheet("color: rgb(0, 0, 255);\n"
"background-color: rgb(85, 255, 255);\n"
"font: 14pt \"MS Shell Dlg 2\";")
        self.b_7.setObjectName("b_7")
        self.b_1 = QtWidgets.QPushButton(self.centralwidget)
        self.b_1.setGeometry(QtCore.QRect(20, 260, 51, 41))
        self.b_1.setStyleSheet("color: rgb(0, 0, 255);\n"
```

```

"background-color: rgb(85, 255, 255);font: 14pt \"MS Shell Dlg 2\";")
    self.b_1.setObjectName("b_1")
    self.b_0 = QtWidgets.QPushButton(self.centralwidget)
    self.b_0.setGeometry(QtCore.QRect(80, 310, 51, 41))
    self.b_0.setStyleSheet("font: 14pt \"MS Shell Dlg 2\";color: rgb(0,
170, 255);\n"
"color: rgb(0, 0, 255);\n"
"background-color: rgb(85, 255, 255);")
    self.b_0.setObjectName("b_0")
    self.b_egale = QtWidgets.QPushButton(self.centralwidget)
    self.b_egale.setGeometry(QtCore.QRect(200, 270, 81, 81))
    self.b_egale.setStyleSheet("font: 14pt \"MS Shell Dlg 2\";color:
rgb(85, 85, 255);\n"
"background-color: rgb(255, 255, 127);")
    self.b_egale.setObjectName("b_egale")
    self.b_mod = QtWidgets.QPushButton(self.centralwidget)
    self.b_mod.setGeometry(QtCore.QRect(110, 70, 81, 31))
    self.b_mod.setStyleSheet("font: 14pt \"MS Shell Dlg 2\";color: rgb(85,
85, 255);\n"
"background-color: rgb(255, 255, 127);")
    self.b_mod.setObjectName("b_mod")
    self.b_3 = QtWidgets.QPushButton(self.centralwidget)
    self.b_3.setGeometry(QtCore.QRect(140, 260, 51, 41))
    self.b_3.setStyleSheet("color: rgb(0, 0, 255);\n"
"background-color: rgb(85, 255, 255);font: 14pt \"MS Shell Dlg 2\";")
    self.b_3.setObjectName("b_3")
    self.b_moins = QtWidgets.QPushButton(self.centralwidget)
    self.b_moins.setGeometry(QtCore.QRect(200, 170, 81, 41))
    self.b_moins.setStyleSheet("font: 14pt \"MS Shell Dlg 2\";color:
rgb(85, 85, 255);\n"
"background-color: rgb(255, 255, 127);")
    self.b_moins.setObjectName("b_moins")
    self.b_mult = QtWidgets.QPushButton(self.centralwidget)
    self.b_mult.setGeometry(QtCore.QRect(200, 120, 81, 41))
    self.b_mult.setStyleSheet("font: 14pt \"MS Shell Dlg 2\";color: rgb(85,
85, 255);\n"
"background-color: rgb(255, 255, 127);")
    self.b_mult.setObjectName("b_mult")
    self.b_divise = QtWidgets.QPushButton(self.centralwidget)
    self.b_divise.setGeometry(QtCore.QRect(200, 70, 81, 41))
    self.b_divise.setStyleSheet("font: 14pt \"MS Shell Dlg 2\";color:
rgb(85, 85, 255);\n"
"background-color: rgb(255, 255, 127);")
    self.b_divise.setObjectName("b_divise")
    self.b_6 = QtWidgets.QPushButton(self.centralwidget)
    self.b_6.setGeometry(QtCore.QRect(140, 210, 51, 41))
    self.b_6.setStyleSheet("color: rgb(0, 0, 255);\n"
"background-color: rgb(85, 255, 255);font: 14pt \"MS Shell Dlg 2\";")

```

```

self.b_6.setObjectName("b_6")

self.b_ac = QtWidgets.QPushButton(self.centralwidget)
self.b_ac.setGeometry(QtCore.QRect(140, 110, 51, 41))
self.b_ac.setStyleSheet("background-color: rgb(255, 170, 127);font:
14pt \"MS Shell Dlg 2\";")
self.b_ac.setObjectName("b_ac")
self.b_2 = QtWidgets.QPushButton(self.centralwidget)
self.b_2.setGeometry(QtCore.QRect(80, 260, 51, 41))
self.b_2.setStyleSheet("color: rgb(0, 0, 255);\n"
"background-color: rgb(85, 255, 255);font: 14pt \"MS Shell Dlg 2\";")
self.b_2.setObjectName("b_2")
self.b_9 = QtWidgets.QPushButton(self.centralwidget)
self.b_9.setGeometry(QtCore.QRect(140, 160, 51, 41))
self.b_9.setStyleSheet("color: rgb(0, 0, 255);\n"
"background-color: rgb(85, 255, 255);\n"
"font: 14pt \"MS Shell Dlg 2\";")
self.b_9.setObjectName("b_9")
self.b_div = QtWidgets.QPushButton(self.centralwidget)
self.b_div.setGeometry(QtCore.QRect(20, 70, 71, 31))
self.b_div.setStyleSheet("color: rgb(85, 85, 255);\n"
"background-color: rgb(255, 255, 127);\n"
"font: 14pt \"MS Shell Dlg 2\";")
self.b_div.setObjectName("b_div")
self.b_del = QtWidgets.QPushButton(self.centralwidget)
self.b_del.setGeometry(QtCore.QRect(20, 310, 51, 41))
self.b_del.setStyleSheet("font: 14pt \"MS Shell Dlg 2\";background-
color: rgb(255, 170, 127);")
self.b_del.setObjectName("b_del")
self.b_plus = QtWidgets.QPushButton(self.centralwidget)
self.b_plus.setGeometry(QtCore.QRect(200, 220, 81, 41))
self.b_plus.setStyleSheet("font: 14pt \"MS Shell Dlg 2\";color: rgb(85,
85, 255);\n"
"background-color: rgb(255, 255, 127);")
self.b_plus.setObjectName("b_plus")
self.b_po = QtWidgets.QPushButton(self.centralwidget)
self.b_po.setGeometry(QtCore.QRect(20, 110, 51, 41))
self.b_po.setStyleSheet("background-color: rgb(255, 170, 127);\n"
"font: 14pt \"MS Shell Dlg 2\";")
self.b_po.setObjectName("b_po")
self.b_4 = QtWidgets.QPushButton(self.centralwidget)
self.b_4.setGeometry(QtCore.QRect(20, 210, 51, 41))
self.b_4.setStyleSheet("color: rgb(0, 0, 255);\n"
"background-color: rgb(85, 255, 255);\n"
"font: 14pt \"MS Shell Dlg 2\";")
self.b_4.setObjectName("b_4")
self.b_5 = QtWidgets.QPushButton(self.centralwidget)
self.b_5.setGeometry(QtCore.QRect(80, 210, 51, 41))
self.b_5.setStyleSheet("color: rgb(0, 0, 255);\n"

```

```

"background-color: rgb(85, 255, 255);\n"
"font: 14pt \"MS Shell Dlg 2\";")
    self.b_5.setObjectName("b_5")
    self.b_point = QtWidgets.QPushButton(self.centralwidget)
    self.b_point.setGeometry(QtCore.QRect(140, 310, 51, 41))
    self.b_point.setStyleSheet("font: 14pt \"MS Shell Dlg 2\";background-
color: rgb(255, 170, 127);")
    self.b_point.setObjectName("b_point")
    MainWindow.setCentralWidget(self.centralwidget)
    self.menubar = QtWidgets.QMenuBar(MainWindow)
    self.menubar.setGeometry(QtCore.QRect(0, 0, 300, 21))
    self.menubar.setObjectName("menubar")
    MainWindow.setMenuBar(self.menubar)
    self.statusbar = QtWidgets.QStatusBar(MainWindow)
    self.statusbar.setObjectName("statusbar")
    MainWindow.setStatusBar(self.statusbar)

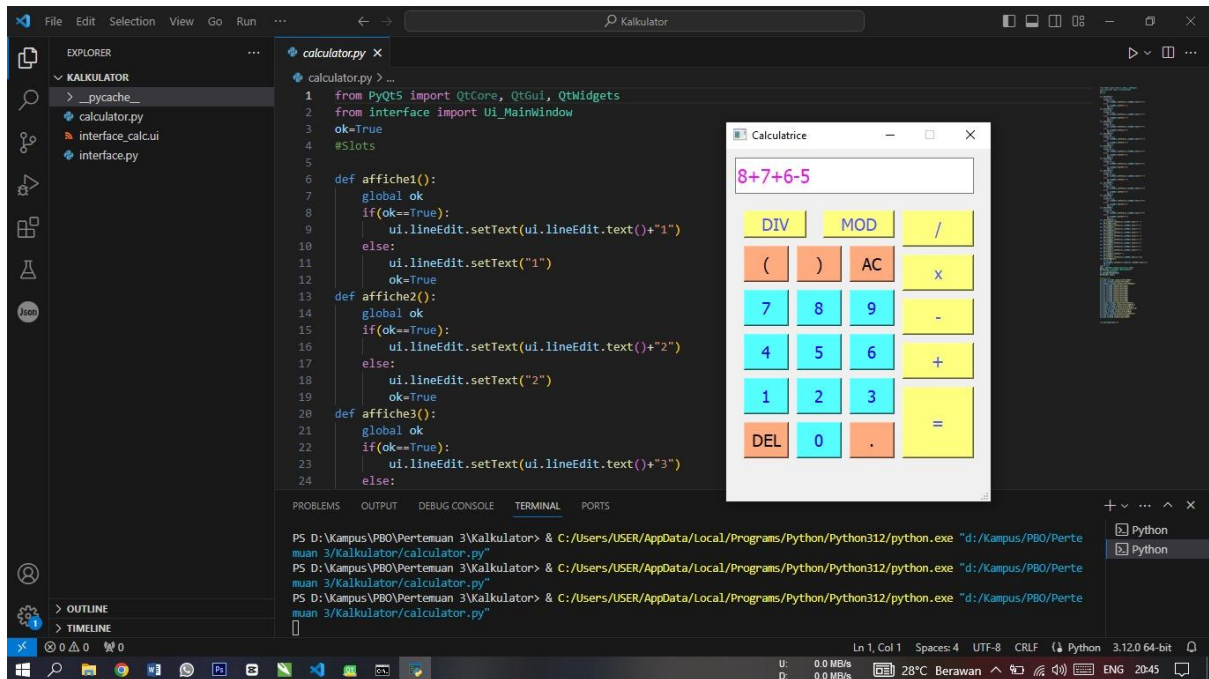
    self.retranslateUi(MainWindow)
    QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "Calculatrice"))
    self.b_8.setText(_translate("MainWindow", "8"))
    self.b_pf.setText(_translate("MainWindow", ""))
    self.b_7.setText(_translate("MainWindow", "7"))
    self.b_1.setText(_translate("MainWindow", "1"))
    self.b_0.setText(_translate("MainWindow", "0"))
    self.b_egale.setText(_translate("MainWindow", "="))
    self.b_mod.setText(_translate("MainWindow", "MOD"))
    self.b_3.setText(_translate("MainWindow", "3"))
    self.b_moins.setText(_translate("MainWindow", "-"))
    self.b_mult.setText(_translate("MainWindow", "x"))
    self.b_divise.setText(_translate("MainWindow", "/"))
    self.b_6.setText(_translate("MainWindow", "6"))
    self.b_ac.setText(_translate("MainWindow", "AC"))
    self.b_2.setText(_translate("MainWindow", "2"))
    self.b_9.setText(_translate("MainWindow", "9"))
    self.b_div.setText(_translate("MainWindow", "DIV"))
    self.b_del.setText(_translate("MainWindow", "DEL"))
    self.b_plus.setText(_translate("MainWindow", "+"))
    self.b_po.setText(_translate("MainWindow", "("))
    self.b_4.setText(_translate("MainWindow", "4"))
    self.b_5.setText(_translate("MainWindow", "5"))
    self.b_point.setText(_translate("MainWindow", "."))

```

## 2.7 Hasil Akhir

*Jika sudah muncul tabel calculatornya, berarti program yang kita masukkan itu benar, Jika tabel calculatornya tidak muncul berarti ada kesalahan dalam memasukkan code programnya.*





## **BAB III**

### **PENUTUP**

#### **3.1 KESIMPULAN**

Bagi Sebagian orang lebih memilih mengetikkan langsung kode-kode terkait komponen Qt di editor, tanpa melalui membuat desain antarmuka. Tapi ada juga yang lebih memilih membuat desain antarmuka di Qt Designer terlebih dahulu, kemudian mengkonversinya ke bentuk source code.

## DAFTAR PUSTAKA

Riverbank Computing. (n.d.). *What is PyQt?*. Retrieved October 07, 2021, from <https://riverbankcomputing.com/software/pyqt/intro>

Python Package Index. (2021, March 21). *Python bindings for the Qt cross platform application toolkit*. Retrieved October 07, 2021 from <https://pypi.org/project/PyQt5/>

calculatrice avec Python et PyQt5 (2020, November 24)  
[https://youtu.be/tCQ\\_bdvdZHY?si=PFGXWw2NPD8yHIOo](https://youtu.be/tCQ_bdvdZHY?si=PFGXWw2NPD8yHIOo)