

Laporan Kelompok 1

Pemrograman II (PBO)

Penyusun :

Nur Iskandar S	220511009
Ridwan Nurjayadi	220511039
Salsa Dhiva Yolanda	220511181
Tri Wahyudin Juniar	220511061

KATA PENGANTAR

Pertama-tama kita panjatkan puji syukur kepada Tuhan Yang Maha Esa karena atas limpahan rahmat, taufik, hidayah serta inayah-Nya kami dapat menyelesaikan Laporan Praktikum Pemrograman II (PBO) ini tanpa menemui hambatan yang berarti. Tidak lupa pula kami ucapkan terima kasih kepada:

1. Freddy Wicaksono, S.Kom, M.Kom selaku dosen pengampu mata kuliah Pemrograman II (PBO)
2. Seluruh pihak yang telah membantu kami dalam menyusun laporan praktikum Pemrograman II (PBO)

Penulis sadar bahwa laporan yang penulis susun ini masih sangat jauh dari sempurna oleh karena itu penulis mengharapkan masukan dan kritikan yang bersifat membangun untuk sebagai acuan agar menjadi lebih baik lagi. Terima kasih.

Cirebon, 24 November 2023

Kelompok 1

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan	1
1.4 Manfaat	1
BAB II	2
PEMBAHASAN.....	2
2.1 Dasar Teori.....	2
A. Pengertian PYQT dan QTDESIGNER	2
B. Deskripsi Kalkulator pada PYQT dan QTDESIGNER	3
BAB III	19
PENUTUP	19
3.1 Kesimpulan	19
3.2 Saran	19

BAB I

PENDAHULUAN

1.1 Latar Belakang

Latar belakang makalah ini terfokus pada pemrograman berorientasi objek (PBO) dalam konteks pembuatan aplikasi Kalkulator PyQt. Pemrograman Berorientasi Objek merupakan paradigma yang mendasarkan struktur kode pada konsep objek, yang memungkinkan pengembangan perangkat lunak yang lebih modular, fleksibel, dan mudah dipahami. Kalkulator adalah alat untuk menghitung dari perhitungan sederhana seperti penjumlahan, pengurangan, perkalian dan pembagian sampai kepada kalkulator sains yang dapat menghitung rumus matematika tertentu. Pada perkembangannya sekarang ini, kalkulator sering dimasukkan sebagai fungsi tambahan dari pada komputer, handphone, bahkan sampai jam tangan, pembuatan aplikasi ini bertujuan untuk memberikan solusi yang menggabungkan keunggulan PBO dan kehandalan PyQt dalam pengembangan antarmuka pengguna.

Keberadaan aplikasi perhitungan kalkulator menggunakan PyQt menjadi penting karena memberikan kemudahan dalam proses perhitungan. Aplikasi kalkulator ini dapat mengoptimalkan efisiensi operasional, mengurangi potensi kesalahan perhitungan, dan secara keseluruhan. Melalui pemahaman konsep PBO. Dengan demikian, makalah ini akan mengeksplorasi implementasi konsep PBO dalam pembuatan aplikasi kalkulator menggunakan PyQt, serta manfaatnya dalam meningkatkan efektivitas proses manajemen sumber daya manusia di lingkungan kerja.

1.2 Rumusan Masalah

1. Apa itu PyQt dan Qt Designer?
2. Bagaimana cara membuat aplikasi kalkulator dengan metode PyQt?

1.3 Tujuan

1. Mengetahui pengertian PyQt dan Qt Designer
2. Mengetahui bagaimana cara membuat aplikasi kalkulator dengan menggunakan metode PyQt

1.4 Manfaat

1. Memberikan kemampuan untuk membuat aplikasi perhitungan menggunakan PyQt
2. Mempermudah perhitungan khususnya untuk pembelajaran matematika

BAB II

PEMBAHASAN

2.1 Dasar Teori

A. Pengertian PyQt dan Qt Designer

- **PyQt**
PyQt adalah lintas platform GUI perangkat Qt yang mengikat pada Python yang diimplementasikan sebagai plug-in. PyQt adalah perangkat lunak gratis yang dikembangkan oleh perusahaan Inggris bernama Riverbank Computing. PyQt tersedia di bawah persyaratan yang sama untuk Qt versi yang lebih tua dari 4,5; hal ini berarti berbagai bentuk termasuk lisensi GNU General Public License (GPL) dan lisensi komersial, tetapi tidak GNU Lesser General Public License (LGPL).[3] PyQt mendukung Microsoft Windows serta berbagai varian dari UNIX, termasuk Linux dan MacOS (atau Darwin OS).[4] PyQt adalah sebuah framework pengembangan aplikasi GUI (Graphical User Interface) untuk bahasa pemrograman Python. Singkatan dari "Python bindings for Qt," PyQt menyediakan binding atau pustaka Python untuk toolkit pengembangan Qt, yang merupakan toolkit GUI lintas platform yang populer. Qt sendiri dikembangkan oleh The Qt Company dan digunakan secara luas dalam industri perangkat lunak untuk membangun antarmuka pengguna yang kaya dan responsif. PyQt memungkinkan pengembang untuk membuat aplikasi dengan antarmuka pengguna yang menarik dan interaktif menggunakan Python, yang dikenal sebagai bahasa pemrograman yang mudah dipelajari dan kuat. Ini memungkinkan penggunaan fitur-fitur Qt, seperti widget GUI, manajemen layout, event handling, dan dukungan untuk pengembangan aplikasi lintas platform.

Berikut adalah beberapa poin penting tentang PyQt:

1. **Bahasa Pemrograman Python:** PyQt mengintegrasikan Python dengan Qt, memungkinkan pengembang menggunakan kekuatan dan fleksibilitas Python untuk pengembangan aplikasi GUI.
2. **Widget dan Antarmuka Pengguna:** PyQt menyediakan berbagai widget dan komponen GUI yang dapat digunakan untuk membangun antarmuka pengguna, seperti tombol, kotak teks, tabel, dan sebagainya.
3. **Manajemen Layout:** PyQt menyediakan mekanisme manajemen layout untuk mengatur tata letak widget secara otomatis, sehingga memudahkan pengembangan antarmuka yang responsif.
4. **Event Handling:** Dukungan yang kuat untuk menangani peristiwa (event handling), seperti merespons input pengguna atau perubahan status aplikasi.
5. **Dokumentasi dan Komunitas:** PyQt memiliki dokumentasi yang kuat dan aktif, serta didukung oleh komunitas pengembang yang besar. Ini

membuatnya lebih mudah untuk memahami dan menyelesaikan masalah yang mungkin muncul selama pengembangan.

6. Cross-Platform: Kode yang ditulis dengan PyQt dapat dijalankan di berbagai platform seperti Windows, Linux, dan macOS, tanpa perubahan yang signifikan.
7. Kompatibilitas dengan Qt: PyQt mencocokkan versi Qt yang ada, seperti PyQt4 yang mendukung Qt 4 dan PyQt5 yang mendukung Qt 5, sehingga memastikan kompatibilitas dengan versi Qt yang digunakan. PyQt memberikan solusi yang efisien dan efektif dalam pengembangan aplikasi berbasis GUI dengan menggunakan kekuatan Python dan fitur-fitur kaya dari Qt.

- QT Designer

Qt Designer adalah alat desain grafis yang disertakan dalam paket pengembangan Qt. Ini memungkinkan pengembang untuk merancang antarmuka pengguna (GUI) secara visual tanpa perlu menulis kode secara manual. Dengan menggunakan Qt Designer, pengguna dapat membuat dan mengatur elemen-elemen GUI dengan metode drag-and-drop, memudahkan proses desain.

Beberapa fitur dan konsep utama dari Qt Designer meliputi:

1. WYSIWYG (What You See Is What You Get)

Qt Designer bekerja berdasarkan prinsip WYSIWYG, yang memungkinkan pengguna melihat antarmuka pengguna sebagaimana adanya selama proses perancangan. Ini membuat pengalaman desain lebih intuitif, karena pengguna dapat langsung melihat hasil dari elemen-elemen yang ditempatkan di dalam formulir.

2. Drag-and-Drop

Desain antarmuka pengguna dapat dilakukan dengan metode drag-and-drop. Pengguna dapat mengambil elemen-elemen GUI seperti tombol, kolom teks, tabel, dan lainnya dari palet Qt Designer dan meletakkannya pada formulir sesuai kebutuhan.

3. Properti dan Layout

Qt Designer menyediakan antarmuka untuk mengatur properti dari setiap elemen GUI, seperti ukuran, warna, dan atribut-atribut lainnya. Selain itu, pengguna dapat mengelola layout dari formulir untuk mengatur posisi dan hubungan antar elemen.

4. Event Handling

Pengguna dapat menentukan dan mengelola penanganan peristiwa (event handling) pada elemen GUI. Ini mencakup menambahkan fungsi atau aksi yang akan dilakukan ketika suatu event tertentu terjadi, seperti saat tombol ditekan.

5. Integrasi dengan Kode Python

Desain yang dibuat dengan Qt Designer dapat diintegrasikan dengan kode Python menggunakan PyQt. Kode Python ini akan menghubungkan logika bisnis atau fungsi-fungsi tertentu dengan antarmuka pengguna yang dirancang.

6. Visualisasi dan Perubahan Dinamis

Pengguna dapat dengan mudah melihat perubahan dinamis dalam desain mereka saat menyesuaikan properti atau tata letak elemen. Hal ini memungkinkan untuk eksperimen dan optimalisasi desain.

Qt Designer membantu mempercepat proses pengembangan GUI dengan menyediakan alat visual yang mudah digunakan, mengurangi ketergantungan pada penulisan kode secara manual. Hasil desain dari Qt Designer dapat diekspor sebagai file XML yang dapat diintegrasikan langsung ke dalam proyek pengembangan menggunakan PyQt.

B. Deskripsi Implementasi

PyQt adalah satu set binding Python yang memungkinkan pengembang untuk membuat antarmuka pengguna grafis (GUI) yang sangat dapat disesuaikan dengan Python. Widget PyQt dapat dibuat dengan berbagai cara tergantung pada tingkat penyesuaian yang diperlukan. Qt Designer adalah alat Qt yang digunakan untuk merancang dan membangun GUI menggunakan editor apa yang Anda lihat adalah apa yang Anda dapatkan. Membangun widget menggunakan alat yang mudah digunakan ini memungkinkan pengembang untuk dengan cepat beralih pada desain sambil melihat umpan balik langsung. Posting ini akan membahas petunjuk langkah demi langkah tentang cara membuat widget PyQt menggunakan Qt Designer + Python.

Langkah Langkah implementasi :

A. Langkah 1

Untuk menginstall PyQt dan Qt Designer di Windows, Anda dapat mengikuti langkah-langkah berikut:

1. Instalasi Python

Pastikan telah menginstal Python di sistem Anda, Anda dapat mengunduh Python dari situs resmi python.org.

2. Pemasangan PyQt

Buka terminal atau command prompt dan jalankan perintah berikut untuk menginstal PyQt5 menggunakan pip:

```
```pip install pyqt5-tools```
```

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19041.3578]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>pip install pyqt5-tools
Collecting pyqt5-tools
 Downloading pyqt5-tools-5.15.9.3.3-py3-none-any.whl (29 kB)
Collecting click (from pyqt5-tools)
 Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
 Obtaining dependency information for click from https://files.pythonhosted.org/packages/00/2e/d53fa4b6fb2cfa713304affc7ca780ce4fc1fd871052771b58311a3229/click-8.1.7-py3-none-any.whl.metadata
 Downloading click-8.1.7-py3-none-any.whl (8.0 kB)
Collecting pyqt5==5.15.9 (from pyqt5-tools)
 Downloading PyQT5-5.15.9-cp37-abi3-win_amd64.whl (6.8 MB)
 0.0/0.0 MB 0.0/0.0 KB/s eta 0:00:00
Collecting pyqt5-plugins<5.15.9.3,>=5.15.9.2.2 (from pyqt5-tools)
 Downloading pyqt5_plugins-5.15.9.2.3-cp311-cp311-win_amd64.whl (66 kB)
 0.0/0.0 MB 0.0/0.0 KB/s eta 0:00:00
Collecting python-dotenv (from pyqt5-tools)
 Downloading python_dotenv-1.0.0-py3-none-any.whl (19 kB)
Collecting PyQT5-sip<13.0,>=12.11 (from pyqt5==5.15.9->pyqt5-tools)
 Obtaining dependency information for PyQT5-sip<13.0,>=12.11 from https://files.pythonhosted.org/packages/8e/e2/d296cb17bae19ba49137a2649934a70d5f48c99b7daa6ce9cf1aecdfeb/PyQT5_sip-12.13.0-cp311-cp311-win_amd64.whl.metadata
 Downloading PyQT5_sip-12.13.0-cp311-cp311-win_amd64.whl.metadata (524 bytes)
 Downloading PyQT5-QT5-5.15.2 (from pyqt5==5.15.9->pyqt5-tools)
 Downloading PyQT5_QT5-5.15.2-py3-none-win_amd64.whl (50.1 MB)
 0.0/50.1 MB 0.0/0.0 KB/s eta 0:00:00
Collecting qt5-tools<5.15.2.2,>=5.15.2.1.2 (from pyqt5-plugins<5.15.9.3,>=5.15.9.2.2->pyqt5-tools)
 Downloading qt5_tools-5.15.2.1.3-py3-none-any.whl (13 kB)
Collecting colorama (from click->pyqt5-tools)
 Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting qt5-applications<5.15.2.3,>=5.15.2.2.2 (from qt5-tools<5.15.2.2,>=5.15.2.1.2->pyqt5-plugins<5.15.9.3,>=5.15.9.2.2->pyqt5-tools)
 Downloading qt5_applications-5.15.2.2.3-py3-none-win_amd64.whl (64.5 MB)
 0.0/64.5 MB 0.0/0.0 KB/s eta 0:00:00
Download click-8.1.7-py3-none-any.whl (97 kB)
 0.0/97.9 kB 0.0/0.0 KB/s eta 0:00:00
Download PyQT5_sip-12.13.0-cp311-cp311-win_amd64.whl (78 kB)
 0.0/78.0 kB 0.0/0.0 KB/s eta 0:00:00
Installing collected packages: PyQT5-QT5, qt5-applications, python-dotenv, PyQT5-sip, colorama, pyqt5, click, qt5-tools, pyqt5-plugins, pyqt5-tools
WARNING: The script dotenv.exe is installed in 'C:\Users\User\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts pypylates.exe, pyqt5.exe and pyqt5.exe are installed in 'C:\Users\User\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The script qt5-tools.exe is installed in 'C:\Users\User\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed PyQT5-QT5-5.15.2 PyQT5-sip-12.13.0 click-8.1.7 colorama-0.4.6 pyqt5-5.15.9 pyqt5-plugins-5.15.9.2.3 pyqt5-tools-5.15.9.3 python-dotenv-1.0.0 qt5-applications-5.15.2.2.3 qt5-tools-5.15.2.1.3
```

### 3. Pemasangan Qt Designer

Qt Designer adalah alat desain visual untuk membuat antarmuka pengguna dengan mudah. Qt Designer biasanya disertakan dalam paket instalasi Qt. Jadi, Anda perlu mengunduh dan menginstal Qt dari situs resmi Qt.

- Pilih versi Qt terbaru yang cocok dengan kebutuhan Anda dan unduh instalator.
- Jalankan instalator dan ikuti langkah-langkahnya. Pastikan untuk memilih komponen Qt Designer selama proses instalasi.

### 4. Konfigurasi PyQtTools (Opsional)

PyQt menyediakan beberapa alat tambahan, termasuk PyQtTools yang mencakup Qt Designer. Untuk mengaktifkan alat ini, Anda perlu menginstal PyQtTools dan menambahkannya ke instalasi PyQt.

```
```pip install PyQt5-tools```
```

Setelah instalasi, Anda dapat mengecek apakah Qt Designer telah diinstal dengan menjalankan perintah berikut:

```
```pyqt5-tools designer```
```

Ini akan membuka Qt Designer dan memastikan bahwa instalasinya berhasil.



Berikut adalah panduan langkah demi langkah untuk membuat kalkulator sederhana menggunakan Qt Designer dan PyQt di Windows:

### B. Langkah 2:

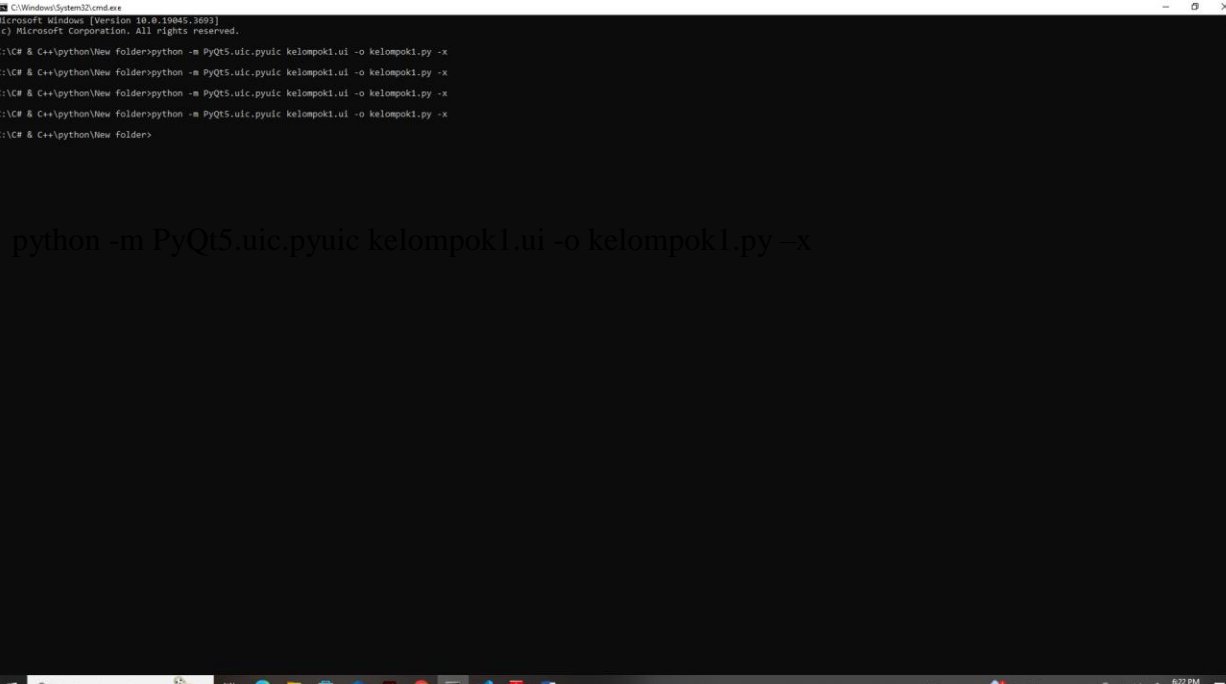
## Desain Antarmuka Kalkulator menggunakan Qt Designer

1. Buka Qt Designer dan buat desain antarmuka kalkulator Anda. Tambahkan tombol dan teks area sesuai kebutuhan Anda. Misalnya, Anda mungkin ingin menambahkan tombol angka 0-9, operator (+, -, \*, /), dan area teks untuk menampilkan hasil.
2. Setiap elemen antarmuka (tombol, teks area) perlu diberikan nama yang unik melalui "Property Editor". Ini akan mempermudah pengaksesan elemen tersebut dalam kode Python.
3. Simpan desain Anda dengan ekstensi file `.ui`. Misalnya, beri nama file Anda `calculator.ui`.

### C. Langkah 3:

## Konversi Desain UI ke Kode Python

Setelah Anda menyimpan desain UI, konversi file `.ui` menjadi file Python menggunakan perintah berikut di terminal atau command prompt:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\CF & C++\python\New folder>python -m PyQt5.uic.pyuic kelompok1.ui -o kelompok1.py -x

C:\CF & C++\python\New folder>python -m PyQt5.uic.pyuic kelompok1.ui -o kelompok1.py -x

C:\CF & C++\python\New folder>python -m PyQt5.uic.pyuic kelompok1.ui -o kelompok1.py -x

C:\CF & C++\python\New folder>python -m PyQt5.uic.pyuic kelompok1.ui -o kelompok1.py -x

C:\CF & C++\python\New folder>python -m PyQt5.uic.pyuic kelompok1.ui -o kelompok1.py -x

C:\CF & C++\python\New folder>python -m PyQt5.uic.pyuic kelompok1.ui -o kelompok1.py -x

python -m PyQt5.uic.pyuic kelompok1.ui -o kelompok1.py -x
```

#### D. Langkah 4:

Setelah di konversi ke *python*, maka akan berubah menjadi seperti dibawah ini, dan jangan lupa menambah kan pseudocode (pseudocode yang disortir) untuk memunculkan angka/assignment/fungsi aritmatika pada aplikasi:

```
-*- coding: utf-8 -*-

Form implementation generated from reading ui file 'kelompok1.ui'
#
Created by: PyQt5 UI code generator 5.15.9
#
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_Kelompok1(object):
 def setupUi(self, Kelompok1):
 Kelompok1.setObjectName("Kelompok1")
 Kelompok1.resize(341, 476)
 Kelompok1.setStyleSheet("background-color: rgb(15, 15, 15);")
 Kelompok1.setFixedSize(Kelompok1.size())
 self.centralwidget = QtWidgets.QWidget(Kelompok1)
 self.centralwidget.setObjectName("centralwidget")
 self.gridLayout_2 = QtWidgets.QGridLayout(self.centralwidget)
 self.gridLayout_2.setObjectName("gridLayout_2")
 self.gridLayout = QtWidgets.QGridLayout()
 self.gridLayout.setObjectName("gridLayout")
 self.push_5 = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_5.sizePolicy().hasHeightForWidth(
))
 self.push_5.setSizePolicy(sizePolicy)
 self.push_5.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n")
```

```

"QPushButton:hover{\n"
"background-color: rgb(52, 157, 77);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(52, 157, 77);\n"
"}")
 self.push_5.setObjectName("push_5")
 self.gridLayout.addWidget(self.push_5, 3, 1, 1, 1)
 self.push_3 = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_3.sizePolicy().hasHeightForWidth(
))
 self.push_3.setSizePolicy(sizePolicy)
 self.push_3.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(52, 157, 77);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(52, 157, 77);\n"
"}")
 self.push_3.setObjectName("push_3")
 self.gridLayout.addWidget(self.push_3, 2, 2, 1, 1)
 self.push_equal = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_equal.sizePolicy().hasHeightForWi
dth())
 self.push_equal.setSizePolicy(sizePolicy)
 self.push_equal.setStyleSheet("QPushButton{\n"
"font: 75 36pt \"MS Shell Dlg 2\";\n"
" color: rgb(255, 255, 127);\n"
"border-radius:30px;\n"
"}\n"
"\n"

```

```

"QPushButton:hover{\n"
"color: rgb(255, 255, 255);\n"
" background-color: rgb(230, 230, 0);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"color: rgb(255, 255, 255);\n"
" background-color: rgb(230, 230, 0);\n"
"}")
 self.push_equal.setObjectName("push_equal")
 self.gridLayout.addWidget(self.push_equal, 5, 3, 1, 1)
 self.push_min = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_min.sizePolicy().hasHeightForWidth
h())
 self.push_min.setSizePolicy(sizePolicy)
 self.push_min.setStyleSheet("QPushButton{\n"
"font: 75 36pt \"MS Shell Dlg 2\";\n"
" color: rgb(255, 255, 127);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"color: rgb(255, 255, 255);\n"
" background-color: rgb(230, 230, 0);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"color: rgb(255, 255, 255);\n"
" background-color: rgb(230, 230, 0);\n"
"}")
 self.push_min.setObjectName("push_min")
 self.gridLayout.addWidget(self.push_min, 2, 3, 1, 1)
 self.push_del = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_del.sizePolicy().hasHeightForWidth
h())
 self.push_del.setSizePolicy(sizePolicy)
 self.push_del.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"

```

```

" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(185, 0, 0);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(185, 0, 0);\n"
"}\n"
""
 self.push_del.setObjectName("push_del")
 self.gridLayout.addWidget(self.push_del, 1, 2, 1, 1)
 self.push_point = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_point.sizePolicy().hasHeightForWidth())
 self.push_point.setSizePolicy(sizePolicy)
 self.push_point.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(52, 157, 77);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(52, 157, 77);\n"
"}")
 self.push_point.setObjectName("push_point")
 self.gridLayout.addWidget(self.push_point, 5, 2, 1, 1)
 self.push_1 = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_1.sizePolicy().hasHeightForWidth(
))
 self.push_1.setSizePolicy(sizePolicy)

```

```

 self.push_1.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(52, 157, 77);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(52, 157, 77);\n"
"}")

 self.push_1.setObjectName("push_1")
 self.gridLayout.addWidget(self.push_1, 2, 0, 1, 1)
 self.push_2 = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_2.sizePolicy().hasHeightForWidth(
))

 self.push_2.setSizePolicy(sizePolicy)
 self.push_2.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(52, 157, 77);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(52, 157, 77);\n"
"}")

 self.push_2.setObjectName("push_2")
 self.gridLayout.addWidget(self.push_2, 2, 1, 1, 1)
 self.push_mult = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_mult.sizePolicy().hasHeightForWidth(
))

```

```

 self.push_mult.setSizePolicy(sizePolicy)
 self.push_mult.setStyleSheet("QPushButton{\n"
"font: 75 36pt \"MS Shell Dlg 2\";\n"
" color: rgb(255, 255, 127);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"color: rgb(255, 255, 255);\n"
" background-color: rgb(230, 230, 0);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"color: rgb(255, 255, 255);\n"
" background-color: rgb(230, 230, 0);\n"
"}")

 self.push_mult.setObjectName("push_mult")
 self.gridLayout.addWidget(self.push_mult, 3, 3, 1, 1)
 self.push_9 = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_9.sizePolicy().hasHeightForWidth(
))

 self.push_9.setSizePolicy(sizePolicy)
 self.push_9.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(52, 157, 77);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(52, 157, 77);\n"
"}")

 self.push_9.setObjectName("push_9")
 self.gridLayout.addWidget(self.push_9, 4, 2, 1, 1)
 self.push_8 = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)

```

```

 sizePolicy.setHeightForWidth(self.push_8.sizePolicy().hasHeightForWidth(
))

 self.push_8.setSizePolicy(sizePolicy)
 self.push_8.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(52, 157, 77);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(52, 157, 77);\n"
"}")

 self.push_8.setObjectName("push_8")
 self.gridLayout.addWidget(self.push_8, 4, 1, 1, 1)
 self.push_4 = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_4.sizePolicy().hasHeightForWidth(
))

 self.push_4.setSizePolicy(sizePolicy)
 self.push_4.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(52, 157, 77);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(52, 157, 77);\n"
"}")

 self.push_4.setObjectName("push_4")
 self.gridLayout.addWidget(self.push_4, 3, 0, 1, 1)
 self.push_zero = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)

```



```

 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_zero.sizePolicy().hasHeightForWidth(
th()))
 self.push_zero.setSizePolicy(sizePolicy)
 self.push_zero.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(52, 157, 77);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(52, 157, 77);\n"
"}")
 self.push_zero.setObjectName("push_zero")
 self.gridLayout.addWidget(self.push_zero, 5, 0, 1, 2)
 self.push_6 = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_6.sizePolicy().hasHeightForWidth(
))
 self.push_6.setSizePolicy(sizePolicy)
 self.push_6.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(52, 157, 77);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(52, 157, 77);\n"
"}")
 self.push_6.setObjectName("push_6")
 self.gridLayout.addWidget(self.push_6, 3, 2, 1, 1)
 self.push_div = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)

```

```

 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_div.sizePolicy().hasHeightForWidth())
 h())
 self.push_div.setSizePolicy(sizePolicy)
 self.push_div.setStyleSheet("QPushButton{\n"
"font: 75 36pt \"MS Shell Dlg 2\";\n"
" color: rgb(255, 255, 127);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"color: rgb(255, 255, 255);\n"
" background-color: rgb(230, 230, 0);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"color: rgb(255, 255, 255);\n"
" background-color: rgb(230, 230, 0);\n"
"}")
 self.push_div.setObjectName("push_div")
 self.gridLayout.addWidget(self.push_div, 4, 3, 1, 1)
 self.push_plus = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_plus.sizePolicy().hasHeightForWidth())
 th())
 self.push_plus.setSizePolicy(sizePolicy)
 self.push_plus.setStyleSheet("QPushButton{\n"
"font: 75 36pt \"MS Shell Dlg 2\";\n"
" color: rgb(255, 255, 127);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"color: rgb(255, 255, 255);\n"
" background-color: rgb(230, 230, 0);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"color: rgb(255, 255, 255);\n"
" background-color: rgb(230, 230, 0);\n"
"}")
 self.push_plus.setObjectName("push_plus")
 self.gridLayout.addWidget(self.push_plus, 1, 3, 1, 1)
 self.push_7 = QtWidgets.QPushButton(self.centralwidget)

```

```

 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_7.sizePolicy().hasHeightForWidth(
))

 self.push_7.setSizePolicy(sizePolicy)
 self.push_7.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(52, 157, 77);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(52, 157, 77);\n"
"}")

 self.push_7.setObjectName("push_7")
 self.gridLayout.addWidget(self.push_7, 4, 0, 1, 1)
 self.label = QtWidgets.QLabel(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(13)
 sizePolicy.setHeightForWidth(self.label.sizePolicy().hasHeightForWidth()
)

 self.label.setSizePolicy(sizePolicy)
 self.label.setStyleSheet("font: 75 36pt \"MS Shell Dlg 2\";\n"
"\n"
"color: rgb(255, 255, 255);")
 self.label.setText("")
 self.label.setAlignment(QtCore.Qt.AlignRight|QtCore.Qt.AlignTrailing|QtC
ore.Qt.AlignVCenter)
 self.label.setObjectName("label")
 self.gridLayout.addWidget(self.label, 0, 0, 1, 4)
 self.push_clear = QtWidgets.QPushButton(self.centralwidget)
 sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
QtWidgets.QSizePolicy.Preferred)
 sizePolicy.setHorizontalStretch(50)
 sizePolicy.setVerticalStretch(10)
 sizePolicy.setHeightForWidth(self.push_clear.sizePolicy().hasHeightForWi
dth())

```

```

 self.push_clear.setSizePolicy(sizePolicy)
 self.push_clear.setStyleSheet("QPushButton{\n"
"font: 75 20pt \"MS Shell Dlg 2\";\n"
" \n"
" \n"
" color: rgb(255, 255, 255);\n"
"border-radius:30px;\n"
"}\n"
"\n"
"QPushButton:hover{\n"
"background-color: rgb(185, 0, 0);\n"
"}\n"
"\n"
"QPushButton:pressed{\n"
"background-color: rgb(185, 0, 0);\n"
"}")

 self.push_clear.setObjectName("push_clear")
 self.gridLayout.addWidget(self.push_clear, 1, 0, 1, 2)
 self.gridLayout_2.addLayout(self.gridLayout, 0, 0, 1, 1)
 Kelompok1.setCentralWidget(self.centralwidget)
 self.menubar = QtWidgets.QMenuBar(Kelompok1)
 self.menubar.setGeometry(QtCore.QRect(0, 0, 341, 21))
 self.menubar.setObjectName("menubar")
 Kelompok1.setMenuBar(self.menubar)
 self.statusbar = QtWidgets.QStatusBar(Kelompok1)
 self.statusbar.setObjectName("statusbar")
 Kelompok1.setStatusBar(self.statusbar)

 self.retranslateUi(Kelompok1)
 QtCore.QMetaObject.connectSlotsByName(Kelompok1)

 self.push_1.clicked.connect(self.method_1)
 self.push_2.clicked.connect(self.method_2)
 self.push_3.clicked.connect(self.method_3)
 self.push_4.clicked.connect(self.method_4)
 self.push_5.clicked.connect(self.method_5)
 self.push_6.clicked.connect(self.method_6)
 self.push_7.clicked.connect(self.method_7)
 self.push_8.clicked.connect(self.method_8)
 self.push_9.clicked.connect(self.method_9)
 self.push_zero.clicked.connect(self.method_zero)
 self.push_point.clicked.connect(self.method_point)
 self.push_plus.clicked.connect(self.method_plus)
 self.push_min.clicked.connect(self.method_min)
 self.push_mult.clicked.connect(self.method_mult)
 self.push_div.clicked.connect(self.method_div)
 self.push_equal.clicked.connect(self.method_equal)
 self.push_clear.clicked.connect(self.method_clear)
 self.push_del.clicked.connect(self.method_del)

```

```
def method_1(self):
 text=self.label.text()
 self.label.setText(text+"1")
```

```
def method_2(self):
 text=self.label.text()
 self.label.setText(text+"2")
```

```
def method_3(self):
 text=self.label.text()
 self.label.setText(text+"3")
```

```
def method_4(self):
 text=self.label.text()
 self.label.setText(text+"4")
```

```
def method_5(self):
 text=self.label.text()
 self.label.setText(text+"5")
```

```
def method_6(self):
 text=self.label.text()
 self.label.setText(text+"6")
```

```
def method_7(self):
 text=self.label.text()
 self.label.setText(text+"7")
```

```
def method_8(self):
 text=self.label.text()
 self.label.setText(text+"8")
```

```
def method_9(self):
 text=self.label.text()
 self.label.setText(text+"9")
```

```
def method_zero(self):
 text=self.label.text()
 self.label.setText(text+"0")
```

```
def method_point(self):
 text=self.label.text()
 self.label.setText(text+".")
```

```
def method_plus(self):
 text=self.label.text()
 self.label.setText(text+"+")
```

```

def method_min(self):
 text=self.label.text()
 self.label.setText(text+"-")

def method_mult(self):
 text=self.label.text()
 self.label.setText(text+"*")

def method_div(self):
 text=self.label.text()
 self.label.setText(text+"/")

def method_clear(self):
 self.label.setText("")

def method_del(self):
 text=self.label.text()
 self.label.setText(text[:len(text)-1])

def method_equal(self):
 text=self.label.text()

 try:
 ans=eval(text)
 self.label.setText(str(ans))
 except:
 self.label.setText("Wrong Input")

def retranslateUi(self, Kelompok1):
 _translate = QtCore.QCoreApplication.translate
 Kelompok1.setWindowTitle(_translate("Kelompok1", "Kalkulator Kelompok
1"))

 self.push_5.setText(_translate("Kelompok1", "5"))
 self.push_5.setShortcut(_translate("Kelompok1", "5"))
 self.push_3.setText(_translate("Kelompok1", "3"))
 self.push_3.setShortcut(_translate("Kelompok1", "3"))
 self.push_equal.setText(_translate("Kelompok1", "="))
 self.push_equal.setShortcut(_translate("Kelompok1", "="))
 self.push_min.setText(_translate("Kelompok1", "-"))
 self.push_min.setShortcut(_translate("Kelompok1", "-"))
 self.push_del.setText(_translate("Kelompok1", "Del"))
 self.push_del.setShortcut(_translate("Kelompok1", "Del"))
 self.push_point.setText(_translate("Kelompok1", "."))
 self.push_point.setShortcut(_translate("Kelompok1", "."))
 self.push_1.setText(_translate("Kelompok1", "1"))
 self.push_1.setShortcut(_translate("Kelompok1", "1"))
 self.push_2.setText(_translate("Kelompok1", "2"))
 self.push_2.setShortcut(_translate("Kelompok1", "2"))
 self.push_mult.setText(_translate("Kelompok1", "x"))

```

```

self.push_mult.setShortcut(_translate("Kelompok1", "*"))
self.push_9.setText(_translate("Kelompok1", "9"))
self.push_9.setShortcut(_translate("Kelompok1", "9"))
self.push_8.setText(_translate("Kelompok1", "8"))
self.push_8.setShortcut(_translate("Kelompok1", "8"))
self.push_4.setText(_translate("Kelompok1", "4"))
self.push_4.setShortcut(_translate("Kelompok1", "4"))
self.push_zero.setText(_translate("Kelompok1", "0"))
self.push_zero.setShortcut(_translate("Kelompok1", "0"))
self.push_6.setText(_translate("Kelompok1", "6"))
self.push_6.setShortcut(_translate("Kelompok1", "6"))
self.push_div.setText(_translate("Kelompok1", "÷"))
self.push_div.setShortcut(_translate("Kelompok1", "/"))
self.push_plus.setText(_translate("Kelompok1", "+"))
self.push_plus.setShortcut(_translate("Kelompok1", "+"))
self.push_7.setText(_translate("Kelompok1", "7"))
self.push_7.setShortcut(_translate("Kelompok1", "7"))
self.push_clear.setText(_translate("Kelompok1", "Clear"))
self.push_clear.setShortcut(_translate("Kelompok1", "Backspace"))

```

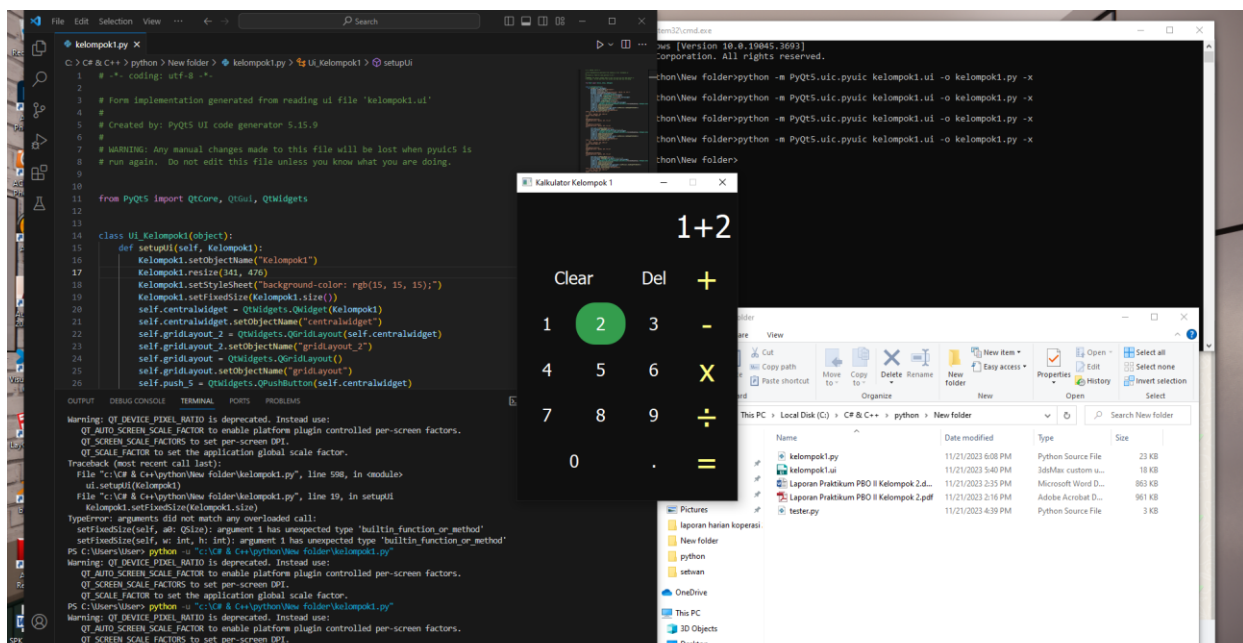
```

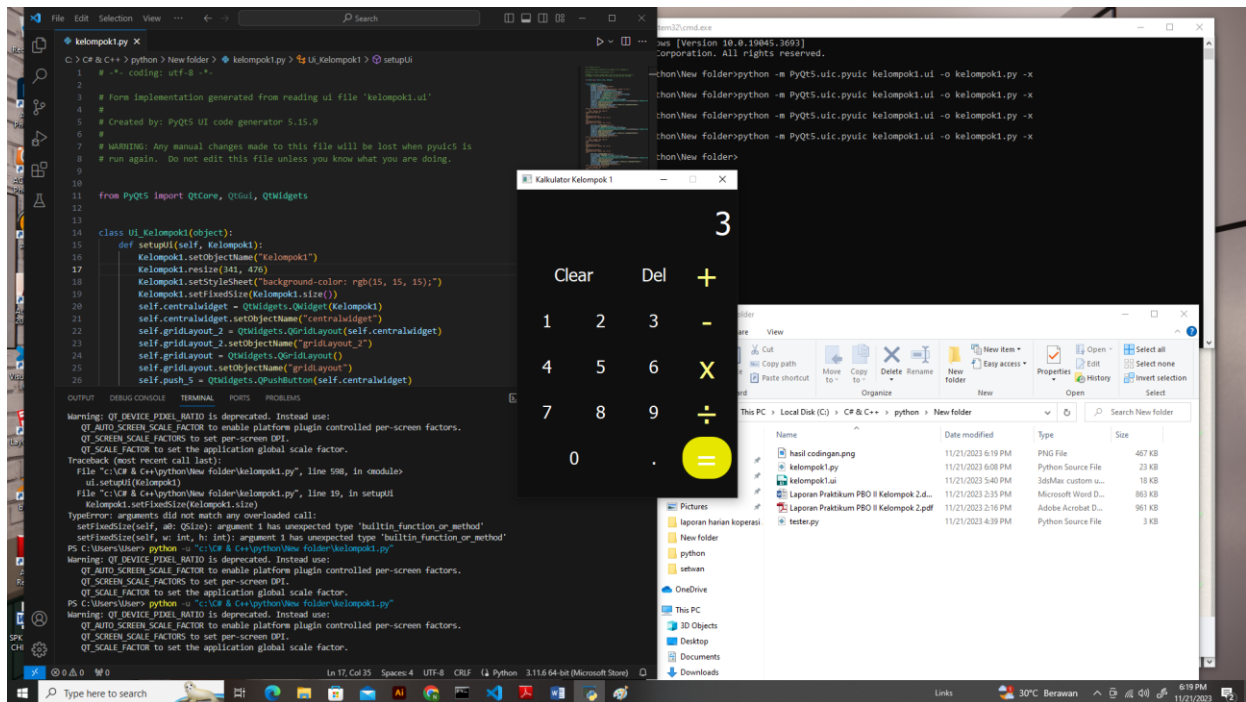
if __name__ == "__main__":
 import sys
 app = QtWidgets.QApplication(sys.argv)
 Kelompok1 = QtWidgets.QMainWindow()
 ui = Ui_Kelompok1()
 ui.setupUi(Kelompok1)
 Kelompok1.show()
 sys.exit(app.exec_())

```

## E. Langkah 5:

Jalankan skrip Python Anda. Ini akan membuka jendela aplikasikalkulator dengan antarmuka yang telah Anda desain.







## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Dalam laporan praktikum ini, kami berhasil mengimplementasikan alat perhitungan kalkulator dengan menggunakan metode PyQt. Hasil perhitungan ditampilkan dengan jelas dan akurat. Implementasi ini memberikan kontribusi praktis dalam berbagai konteks, mulai dari digunakan untuk mengembangkan konsep, *Adding It Up: Helping Children Learn Mathematics* (NRC, 2001), Kemudahan Penggunaan Antarmuka, Pada aplikasi yang telah dibangun, terlihat jelas bahwa antarmuka pengguna yang dirancang menggunakan PyQt memberikan kemudahan penggunaan. Pengguna dapat dengan mudah memahami dan menggunakan fungsi-fungsi perhitungan tanpa kesulitan, dan Kalkulator Meningkatkan Pemecahan Soal. Aplikasi ini juga memberikan wawasan tentang penggunaan PyQt dan Qt Designer dalam pengembangan antarmuka web yang interaktif dan menarik. Selain manfaat praktisnya, penelitian ini memberikan pemahaman tentang bagaimana PyQt dan Qt Designer dapat digunakan dalam pengembangan perangkat lunak. PyQt adalah serangkaian pustaka Python yang menyediakan pengikat (bindings) untuk Qt, sebuah framework pengembangan perangkat lunak lintas platform. Dengan menggunakan PyQt, pengembang Python dapat membuat aplikasi dengan antarmuka grafis menggunakan fungsi dan fitur yang disediakan oleh Qt. Qt Designer adalah alat pengembangan grafis yang disediakan oleh Qt untuk merancang antarmuka pengguna (GUI) aplikasi. Dengan Qt Designer, pengembang dapat membuat antarmuka pengguna secara visual tanpa menulis kode secara manual. Antarmuka yang dirancang dengan Qt Designer dapat disimpan dalam format XML dan kemudian diintegrasikan ke dalam aplikasi Qt menggunakan file tersebut. Implementasi ini juga menunjukkan pentingnya validasi input pengguna dalam memastikan data yang dimasukkan sesuai. Penelitian ini memiliki potensi untuk memberikan landasan bagi pengembangan lebih lanjut dalam penggunaan PyQt dalam proyek-proyek pengembangan aplikasi web.

#### **3.2 Saran**

Dalam pengembangan pembuatan aplikasi kalkulator menggunakan PyQt, beberapa saran dapat diberikan untuk memperbaiki dan meningkatkan kualitas aplikasi serta pengalaman mengevaluasi dan mengoptimalkan antarmuka pengguna agar lebih intuitif dan mudah digunakan. Pemahaman Konsep Layout, konsep layout pada Qt Designer terlihat baik dalam menyusun elemen-elemen antarmuka pengguna. Layout yang dinamis memberikan keleluasaan bagi pengguna untuk menyesuaikan ukuran jendela tanpa mengorbankan tata letak elemen. Kemudian Kejelasan dan Konsistensi Desain, Desain antarmuka yang konsisten dan jelas membantu pengguna untuk dengan cepat memahami cara menggunakan aplikasi. Element-elemen seperti warna, ukuran font, dan penataan memberikan kesan visual yang baik. Kami sangat mengharapkan, dengan diselesaikannya laporan praktikum ini, baik pembaca maupun penyusun, bisa saling memperbaiki dan memberikan saran yang membangun.