**LAPORAN AKHIR PRAKTIKUM**

Mata Praktikum : Algoritma dan Pemrograman 2B

Kelas : 1IA23

Praktikum Ke- : 7

Tanggal : 19 May 2022

Materi : PyQt

NPM : 51422041

Nama : Muhammad Faris Rasyid Raharjo

Ketua Asisten :  Fernando Michael Hebert

Nama Asisten :

Jumlah Lembar : 7 Lembar

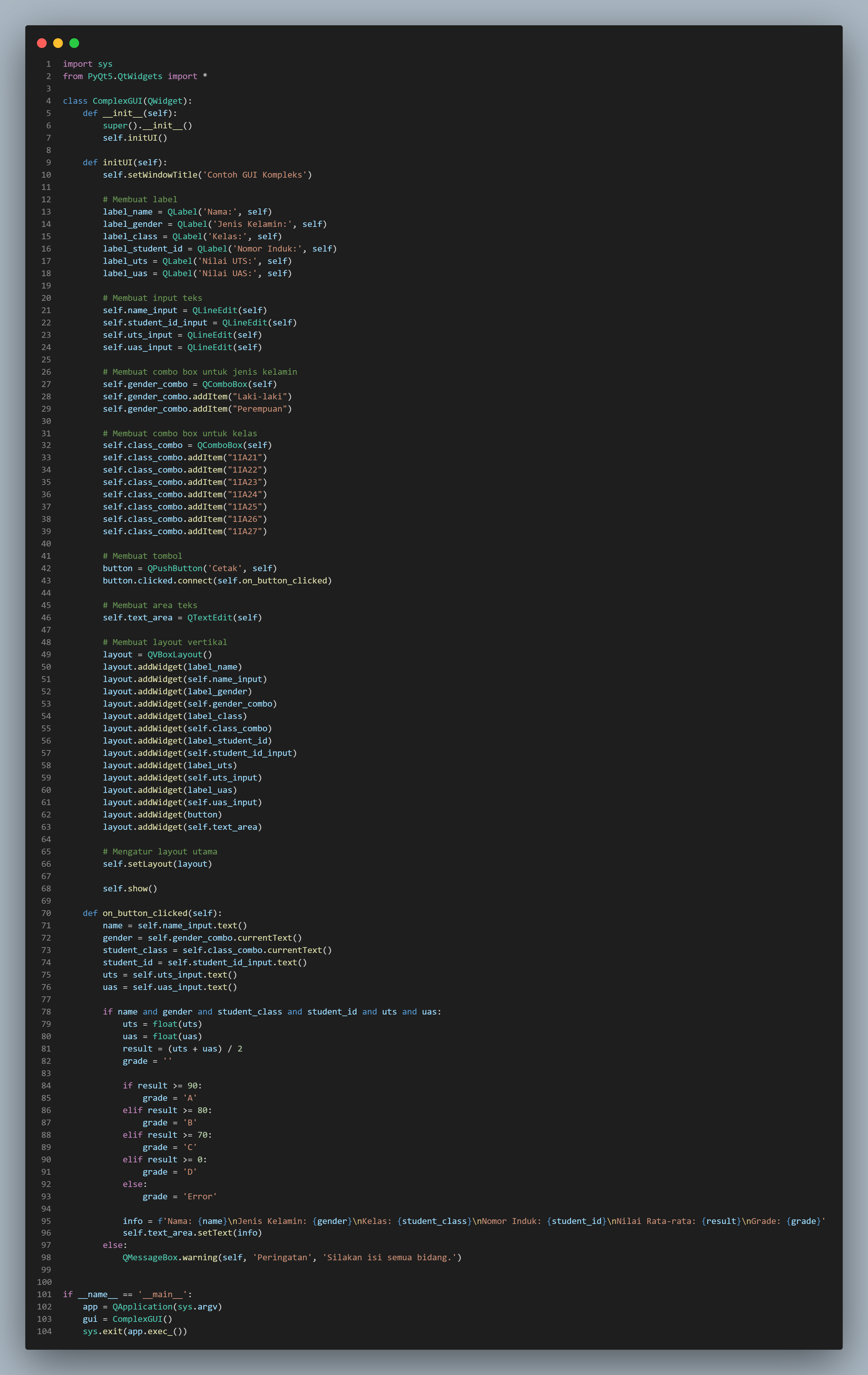


LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2022

**LISTING**

****

**LOGIKA**

Berikut adalah penjelasan program tersebut yang telah dibagi per segmen:

**Segment 1: Import Library**

python

import sys

from PyQt5.QtWidgets import \*

* Segmen ini bertanggung jawab untuk mengimpor modul sys dan kelas-kelas yang diperlukan dari modul QtWidgets di PyQt5.

**Segment 2: Deklarasi Kelas ComplexGUI**

python

class ComplexGUI(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.initUI()

* Segmen ini mendefinisikan kelas ComplexGUI yang merupakan turunan dari kelas QWidget dari modul QtWidgets.
* Konstruktor \_\_init\_\_ dipanggil saat objek dari kelas ComplexGUI dibuat.
* Method initUI akan dipanggil untuk menginisialisasi antarmuka pengguna.

**Segment 3: Method initUI**

python

def initUI(self):

self.setWindowTitle('Contoh GUI Kompleks')

# Membuat label

label\_name = QLabel('Nama:', self)

label\_gender = QLabel('Jenis Kelamin:', self)

label\_class = QLabel('Kelas:', self)

label\_student\_id = QLabel('Nomor Induk:', self)

label\_uts = QLabel('Nilai UTS:', self)

label\_uas = QLabel('Nilai UAS:', self)

# Membuat input teks

self.name\_input = QLineEdit(self)

self.student\_id\_input = QLineEdit(self)

self.uts\_input = QLineEdit(self)

self.uas\_input = QLineEdit(self)

* Segmen ini menginisialisasi antarmuka pengguna dengan menambahkan label dan input teks menggunakan kelas QLabel dan QLineEdit dari modul QtWidgets.

**Segment 4: Combo Box Jenis Kelamin**

python

# Membuat combo box untuk jenis kelamin

self.gender\_combo = QComboBox(self)

self.gender\_combo.addItem("Laki-laki")

self.gender\_combo.addItem("Perempuan")

* Segmen ini membuat combo box menggunakan kelas QComboBox dan menambahkan dua item "Laki-laki" dan "Perempuan" ke dalam combo box tersebut.

**Segment 5: Tombol Cetak**

python

# Membuat tombol

button = QPushButton('Cetak', self)

button.clicked.connect(self.on\_button\_clicked)

* Segmen ini membuat tombol menggunakan kelas QPushButton dan menghubungkannya ke method on\_button\_clicked saat tombol tersebut diklik.

**Segment 6: Area Teks**

python

# Membuat area teks

self.text\_area = QTextEdit(self)

* Segmen ini membuat area teks menggunakan kelas QTextEdit untuk menampilkan output informasi.

**Segment 7: Layout dan Penempatan Widget**

python

# Membuat layout vertikal

layout = QVBoxLayout()

layout.addWidget(label\_name)

layout.addWidget(self.name\_input)

layout.addWidget(label\_gender)

layout.addWidget(self.gender\_combo)

layout.addWidget(label\_class)

layout.addWidget(self.class\_combo)

layout.addWidget(label\_student\_id)

layout.addWidget(self.student\_id\_input)

layout.addWidget(label\_uts)

layout.addWidget(self.uts\_input)

layout.addWidget(label\_uas)

layout.addWidget(self.uas\_input)

layout.addWidget(button)

layout.addWidget(self.text\_area)

# Mengatur layout utama

self.setLayout(layout)

self.show()

* Segmen ini membuat layout vertikal menggunakan kelas QVBoxLayout untuk menempatkan widget-widget dalam tata letak yang diinginkan.
* Widget-widget seperti label, input teks, combo box, tombol, dan area teks ditambahkan ke dalam layout.
* Layout utama ditetapkan menggunakan setLayout().
* Terakhir, antarmuka pengguna ditampilkan dengan memanggil show().

**Segment 8: Method on\_button\_clicked**

python

def on\_button\_clicked(self):

name = self.name\_input.text()

gender = self.gender\_combo.currentText()

student\_class = self.class\_combo.currentText()

student\_id = self.student\_id\_input.text()

uts = self.uts\_input.text()

uas = self.uas\_input.text()

if name and gender and student\_class and student\_id and uts and uas:

uts = float(uts)

uas = float(uas)

result = (uts + uas) / 2

grade = ''

if result >= 90:

grade = 'A'

elif result >= 80:

grade = 'B'

elif result >= 70:

grade = 'C'

elif result >= 0:

grade = 'D'

else:

grade = 'Error'

info = f'Nama: {name}\nJenis Kelamin: {gender}\nKelas: {student\_class}\nNomor Induk: {student\_id}\nNilai Rata-rata: {result}\nGrade: {grade}'

self.text\_area.setText(info)

else:

QMessageBox.warning(self, 'Peringatan', 'Silakan isi semua bidang.')

* Segmen ini adalah method yang akan dipanggil saat tombol "Cetak" diklik.
* Method ini mengambil nilai dari input teks dan combo box yang diisi oleh pengguna.
* Jika semua bidang telah diisi, maka program akan menghitung nilai rata-rata dan menentukan grade berdasarkan hasilnya.
* Informasi tentang nama, jenis kelamin, kelas, nomor induk, nilai rata-rata, dan grade akan ditampilkan di area teks.
* Jika ada bidang yang kosong, maka akan muncul peringatan menggunakan QMessageBox.

**Segment 9: Main Program**

python

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

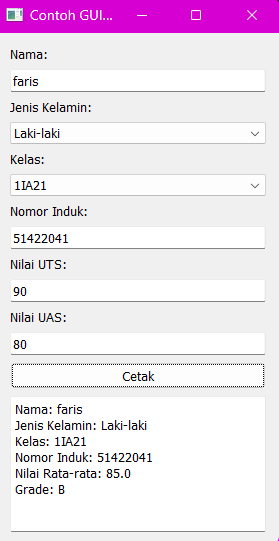
app = QApplication(sys.argv)

gui = ComplexGUI()

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

* Segmen ini adalah bagian utama dari program.
* Objek QApplication dibuat untuk menginisialisasi aplikasi Qt.
* Objek ComplexGUI dibuat untuk memulai antarmuka pengguna.
* Pemanggilan sys.exit(app.exec\_()) digunakan untuk menjalankan aplikasi Qt hingga keluar.

**OUTPUT**

****