

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 : 8 Lembar



LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2022

LISTING

```
1 import sys
2 from PyQt5.QtWidgets import *
3
4 class ComplexGUI(QWidget):
5     def __init__(self):
6         super().__init__()
7         self.initUI()
8
9     def initUI(self):
10         self.setWindowTitle('Contoh GUI Kompleks')
11
12         # Membuat label
13         label_name = QLabel('Nama:', self)
14         label_gender = QLabel('Jenis Kelamin:', self)
15         label_class = QLabel('Kelas:', self)
16         label_student_id = QLabel('Nomor Induk:', self)
17         label_uts = QLabel('Nilai UTS:', self)
18         label_uas = QLabel('Nilai UAS:', self)
19
20         # Membuat input teks
21         self.name_input = QLineEdit(self)
22         self.student_id_input = QLineEdit(self)
23         self.uts_input = QLineEdit(self)
24         self.uas_input = QLineEdit(self)
25
26         # Membuat combo box untuk jenis kelamin
27         self.gender_combo = QComboBox(self)
28         self.gender_combo.addItem("Laki-laki")
29         self.gender_combo.addItem("Perempuan")
30
31         # Membuat combo box untuk kelas
32         self.class_combo = QComboBox(self)
33         self.class_combo.addItem("11A21")
34         self.class_combo.addItem("11A22")
35         self.class_combo.addItem("11A23")
36         self.class_combo.addItem("11A24")
37         self.class_combo.addItem("11A25")
38         self.class_combo.addItem("11A26")
39         self.class_combo.addItem("11A27")
40
41         # Membuat tombol
42         button = QPushButton('Cetak', self)
43         button.clicked.connect(self.on_button_clicked)
44
45         # Membuat area teks
46         self.text_area = QTextEdit(self)
47
48         # Membuat layout vertikal
49         layout = QVBoxLayout()
50         layout.addWidget(label_name)
51         layout.addWidget(self.name_input)
52         layout.addWidget(label_gender)
53         layout.addWidget(self.gender_combo)
54         layout.addWidget(label_class)
55         layout.addWidget(self.class_combo)
56         layout.addWidget(label_student_id)
57         layout.addWidget(self.student_id_input)
58         layout.addWidget(label_uts)
59         layout.addWidget(self.uts_input)
60         layout.addWidget(label_uas)
61         layout.addWidget(self.uas_input)
62         layout.addWidget(button)
63         layout.addWidget(self.text_area)
64
65         # Mengatur layout utama
66         self.setLayout(layout)
67
68         self.show()
69
70     def on_button_clicked(self):
71         name = self.name_input.text()
72         gender = self.gender_combo.currentText()
73         student_class = self.class_combo.currentText()
74         student_id = self.student_id_input.text()
75         uts = self.uts_input.text()
76         uas = self.uas_input.text()
77
78         if name and gender and student_class and student_id and uts and uas:
79             uts = float(uts)
80             uas = float(uas)
81             result = (uts + uas) / 2
82             grade = ''
83
84             if result >= 90:
85                 grade = 'A'
86             elif result >= 80:
87                 grade = 'B'
88             elif result >= 70:
89                 grade = 'C'
90             elif result >= 60:
91                 grade = 'D'
92             else:
93                 grade = 'Error'
94
95             info = f'Nama: {name}\nJenis Kelamin: {gender}\nKelas: {student_class}\nNomor Induk: {student_id}\nNilai Rata-rata: {result}\nGrade: {grade}'
96             self.text_area.setText(info)
97         else:
98             QMessageBox.warning(self, 'Peringatan', 'Silakan isi semua bidang.')
99
100
101 if __name__ == '__main__':
102     app = QApplication(sys.argv)
103     gui = ComplexGUI()
104     sys.exit(app.exec_())
```


LOGIKA

```
1 import sys
2 import requests
3 from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QMessageBox, QVBoxLayout, QPushButton, QLineEdit, QTextEdit
4
```

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


```
1 class komplek(QWidget):
2     def __init__(self):
3         super().__init__()
4         self.initUI()
```

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.



```
1  label_name = QLabel('Nama:',self)
2      label_jk = QLabel('Jenis Kelamin:',self)
3      label_class = QLabel('Kelas:',self)
4      label_student_id = QLabel('Nomor Induk:',self)
5      label_uts = QLabel('Nilai UTS:',self)
6      label_uas = QLabel('Nilai UAS:',self)
7
8
9      self.name_input = QLineEdit(self)
10     self.jk_input = QComboBox(self)
```

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



```
1  self.jk_input.addItem(' ')
2      self.jk_input.addItem('laki - laki')
3      self.jk_input.addItem('perempuan')
```

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



```
1
2 button = QPushButton('Cetak',self)
3 button2 = QPushButton('Print From Server',self)
4 button.clicked.connect(self.on_button_clicked)
5 button2.clicked.connect(self.ketika_diklik)
```

membuat tombol menggunakan kelas `QPushButton` dan menghubungkannya ke method `on_button_clicked` saat tombol tersebut diklik.membuat area teks menggunakan kelas `QTextEdit` untuk menampilkan output informasi.



```
1  self.class_input = QComboBox(self)
2      self.class_input.addItem(' ')
3      self.class_input.addItem('1IA21')
4      self.class_input.addItem('1IA22')
5      self.class_input.addItem('1IA23')
6      self.class_input.addItem('1IA24')
7      self.class_input.addItem('1IA25')
8      self.class_input.addItem('1IA26')
9      self.class_input.addItem('1IA27')
10
11     self.student_id_input = QLineEdit(self)
12     self.class_uts_input = QLineEdit(self)
13     self.student_uas_input = QLineEdit(self)
14
15
16     button = QPushButton('Cetak',self)
17     button2 = QPushButton('Print From Server',self)
18     button.clicked.connect(self.on_button_clicked)
19     button2.clicked.connect(self.ketika_diklik)
20
21     self.text_area = QTextEdit(self)
22
23     layout = QVBoxLayout()
24     layout.addWidget(label_name)
25     layout.addWidget(self.name_input)
26     layout.addWidget(label_jk);
27     layout.addWidget(self.jk_input)
28     layout.addWidget(label_class)
29     layout.addWidget(self.class_input)
30     layout.addWidget(label_student_id)
31     layout.addWidget(self.student_id_input)
32     layout.addWidget(label_uts)
33     layout.addWidget(self.class_uts_input)
34     layout.addWidget(label_uas)
35     layout.addWidget(self.student_uas_input)
36     layout.addWidget(button)
37     layout.addWidget(button2)
38     layout.addWidget(self.text_area)
39
40
41
42     self.setLayout(layout)
43
44     self.show()
```

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()`.

```
1 def on_button_clicked(self):
2     name = self.name_input.text()
3     jk = self.jk_input.currentText()
4     kelas = self.class_input.currentText()
5     id = self.student_id_input.text()
6     uts = int(self.class_uts_input.text())
7     uas = int(self.student_uas_input.text())
8
9     rata = (uts + uas) / 2
10
11     if rata >= 90:
12         nilai = "A"
13     elif rata >= 80:
14         nilai = "B"
15     elif rata >= 70:
16         nilai = "C"
17     elif rata < 70:
18         nilai = "D"
19     else:
20         nilai = "Error"
21
22
23
24     if name and kelas and id:
25         info = f>Nama: {name}\nJenis Kelamin: {jk}\nKelas: {kelas}\nNomor Induk: {id}\nNilai Rata Rata: {rata}\nGrade: {nilai}'
26         self.text_area.setText(info)
27     else:
28         QMessageBox.warning(self, 'Peringatan', 'Silahkan isi semua bidang')
29
30
```

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`.

-



```
1
2  if __name__ == '__main__':
3      app = QApplication(sys.argv)
4      gui = komplek()
5      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

Activity

Nama:

Jenis Kelamin:

Kelas:

Nomor Induk:

Nilai UTS:

Nilai UAS:

Cetak

Print From Server

Activity

Nama:

Faris

Jenis Kelamin:

laki - laki

Kelas:

11A23

Nomor Induk:

51422041

Nilai UTS:

90

Nilai UAS:

80

Cetak

Print From Server

Nama: Faris

Jenis Kelamin: laki - laki

Kelas: 11A23

Nomor Induk: 51422041

Nilai Rata Rata: 85.0

Grade: B