

REFLEKSI MATAKULIAH PEMROGRAMAN DEKSTOP

Disusun untuk Memenuhi Tugas Matakuliah Pemrograman Dekstop yang Dibimbing oleh Moh.
Kautsar Sophan, S.Kom., M.MT.



Disusun oleh:

Achmad Farid Alfa Waid

(190411100073)

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

2020

1. Buat tulisan yang menjelaskan hal-hal apa saja yang telah anda pelajari di kuliah ini!

Pada matakuliah pemrograman desktop yang sudah berlangsung selama 5 pertemuan ini, sudah banyak sekali yang saya pelajari. Pertama pada awal pertemuan saya berkenalan dengan PyQt5 yang ada di python dan bagaimana cara menjalankan pyqt untuk membuat sebuah tampilan GUI. Seiring berjalannya waktu saya belajar banyak syntax seperti syntax untuk membuat tampilan label, mengatur letak dan ukuran window, membuat button, dan mengatur button tersebut agar sesuai keinginan saya dengan syntax stylesheet, membuat checkbox, membuat radio button, membuat line edit, membuat combo box, membuat group box, membuat spin box, membuat date time edit, membuat slider, membuat scroll bar, membuat dial, membuat signal, dan membuat berbagai macam layout. Pada waktu lain saya juga mencari referensi di internet untuk menggunakan syntax lain, seperti menconnectkan sebuah button ke fungsi lain ketika button tersebut di klik.

2. Buat 1 produk aplikasi bebas!

Pada tugas kali ini saya membuat sebuah aplikasi BMI (Body Mass Index) Calculator, jadi pada aplikasi ini user dapat menginputkan tinggi badan dan berat bandannya masing – masing yang nantinya akan dihitung oleh aplikasi ini dan akan di perlihatkan hasilnya.

3. Tuliskan listing program + jelaskan!

```
4. import sys
5. from PyQt5.QtWidgets import *
6. from PyQt5.Qt import *
7.
8. #Membuat class yang didalamnya terdapat QWidget
9. class BMI(QWidget):
10.     #Membuat fungsi init untuk inisialisasi class BMI
11.     def __init__(self):
12.         #untuk mengembalikan semua atribut dan method yang ada
13.         super().__init__()
14.         #memanggil fungsi BMI_Calculate yang sudah dibuat agar ditampilkan
           hasilnya
15.         self.BMI_Calculate()
16.
17.     #membuat fungsi BMI calculate
18.     def BMI_Calculate(self):
19.         #membuat grid layout dengan nama variabel grid
20.         grid = QGridLayout()
21.         #mengatur batas pada isi konten layout grid yaitu batas kana,kiri,
           atas,bawah bernilai 10
22.         grid.setContentsMargins(10,10,10,10)
23.
24.         #membuat sebuah QGroupBox dengan nama variabel group1
25.         group1 = QGroupBox("Calculator untuk menghitung BMI (Body Mass Ind
           ex)",self)
```

```

26.         #mengatur semua isi konten yang ada di group1 menjadi tulisan bold
           berwarna hitam dengan ukuran 15px
27.         group1.setStyleSheet("font: bold 15px; color: black;")
28.         #memasukkan group1 kedalam layout grid
29.         grid.addWidget(group1, 1, 0)
30.         #membuat QVBoxLayout dengan nama variabel vbox
31.         vbox = QVBoxLayout()
32.         #membuat vbox menjadi layout utama tetapi berada di dalam group1
33.         group1.setLayout(vbox)
34.
35.         #membuat label1 dan akan dimasukkan ke dalam layout vbox
36.         label1 = QLabel(self)
37.         label1.setText("Masukkan Tinggi Badan Anda (m/meter):")
38.         vbox.addWidget(label1)
39.
40.         #membuat QLineEdit dengan nama variabel height_input
41.         self.height_input = QLineEdit(self)
42.         #mengatur height_input dengan border 0.5 px dan berwarna solid bla
ck
43.         self.height_input.setStyleSheet("border: 0.5px solid black;")
44.         #memasukkan height_input kedalam layout vbox
45.         vbox.addWidget(self.height_input)
46.         #menambahkan addStretch ke dalam vbox agar tertata rapi
47.         vbox.addStretch()
48.
49.         #membuat label1 dan akan dimasukkan ke dalam layout vbox
50.         label2 = QLabel(self)
51.         label2.setText("Masukkan Berat Badan Anda (kg/kilogram):")
52.         vbox.addWidget(label2)
53.
54.         #membuat QLineEdit dengan nama variabel weight_input
55.         self.weight_input = QLineEdit(self)
56.         #mengatur weight_input dengan border 0.5 px dan berwarna solid bla
ck
57.         self.weight_input.setStyleSheet("border: 0.5px solid black;")
58.         #memasukkan weight_input kedalam layout vbox
59.         vbox.addWidget(self.weight_input)
60.         #menambahkan addStretch ke dalam vbox agar tertata rapi
61.         vbox.addStretch(1)
62.
63.         #membuat sebuah button dengan nama variable cal_button
64.         self.cal_button = QPushButton("Calculate", self)
65.         #mengatur cal_button dengan background berwarna #33f8c7
66.         self.cal_button.setStyleSheet("background-color: #33f8c7;")
67.         #memasukkan cal_button kedalam layout vbox

```

```

68.         vbox.addWidget(self.cal_button)
69.         #menambahkan addStretch ke dalam vbox agar tertata rapi
70.         vbox.addStretch(1)
71.
72.         #membuat sebuah button dengan nama variable hasil
73.         self.hasil = QPushButton("Hasil", self)
74.         ##mengatur hasil dengan font bold dan ukuran 20px dan berwarna hit
    am
75.         self.hasil.setStyleSheet("font: bold 20px; color: black")
76.         #mengatur button hasil dengan setDisabled agar tidak daat diklik
77.         self.hasil.setDisabled(True)
78.         #memasukkan button hasil ke dalam layout vbox
79.         vbox.addWidget(self.hasil)
80.
81.         #membuat QGroupBox dengan nama variabel group2
82.         group2 = QGroupBox("Classification BMI",self)
83.         #mengatur group2 dengan font bol dan ukuran 15px dan berwarna hita
    m
84.         group2.setStyleSheet("font: bold 15px; color: black;")
85.         #memasukkan group2 kedalam layout utama grid
86.         grid.addWidget(group2, 1, 1)
87.         #membuat QVBoxLayout yang kedua untuk group2 dengan nama variabel
    vbox2
88.         vbox2 = QVBoxLayout()
89.         #mengatur vbox2 sebagai layout yang berada di dalam group2
90.         group2.setLayout(vbox2)
91.
92.         #membuat label 3
93.         label3 = QLabel(self)
94.         label3.setText("Berat Badan Kurang: (BMI < 18.5)")
95.         #mengatur label3 dengan background berwarna orange dan font bold d
    engan ukuran 20px dan warna tulisan hitam
96.         label3.setStyleSheet("background-
    color: orange; font: bold 20px; color: black")
97.         #memasukkan label3 kedalam layout vbox2
98.         vbox2.addWidget(label3)
99.
100.        #membuat label 4
101.        label4 = QLabel(self)
102.        label4.setText("Berat Badan Normal: (18.5 <= BMI <= 22.9)")
103.        #mengatur label3 dengan background berwarna hijau dan font b
    old dengan ukuran 20px dan warna tulisan hitam
104.        label4.setStyleSheet("background-
    color: green; font: bold 20px; color: black")
105.        #memasukkan label4 kedalam layout vbox2

```

```

106.         vbox2.addWidget(label4)
107.
108.         #membuat label 5
109.         label5 = QLabel(self)
110.         label5.setText("Berat Badan Cenderung Obesitas: (23 <= BMI <
    = 29.9)")
111.         #mengatur label3 dengan background berwarna kuning dan font
    bold dengan ukuran 20px dan warna tulisan hitam
112.         label5.setStyleSheet("background-
    color: yellow; font: bold 20px; color: black")
113.         #memasukkan label5 kedalam layout vbox2
114.         vbox2.addWidget(label5)
115.
116.         #membuat label 6
117.         label6 = QLabel(self)
118.         label6.setText("Obesitas: (BMI >= 30)")
119.         #mengatur label3 dengan background berwarna merah dan font b
    old dengan ukuran 20px dan warna tulisan hitam
120.         label6.setStyleSheet("background-
    color: red; font: bold 20px; color: black")
121.         #memasukkan label6 kedalam layout vbox2
122.         vbox2.addWidget(label6)
123.
124.         #connect button to fungsi on_click
125.         self.cal_button.clicked.connect(self.on_click)
126.         #membuat grid menjadi layout utama di window
127.         self.setLayout(grid)
128.
129.         #membuat fungsi on_click yang berisi sistem untuk menghitung nil
    ai BMI
130.         def on_click(self):
131.             #membuat QMessageBox dengan nama variabel notif
132.             notif = QMessageBox()
133.             try:
134.                 #membuat variabel weight dan height yang berisi inputan
    dari weight_input dan height_input yang akan dirubah menjadi sebuah float
135.                 weight = float(self.weight_input.text())
136.                 height = float(self.height_input.text())
137.                 #membuat variabel BMI untuk menghitung nilai BMI
138.                 BMI = weight/(height*height)
139.                 #menampilkan nilai BMI pada button hasil
140.                 self.hasil.setText("Hasil Perhitungan BMI Anda: " + str(
    "%.2f" %BMI))
141.                 ##### SISTEM BMI #####
142.                 #jika nilai BMI < 18.5

```

```

143.         if BMI < 18.5:
144.             #button hasil akan diubah menjadi background berwarna
a orange dan font bold berukuran wopx dan berwarna hitam
145.             self.hasil.setStyleSheet("background-
color: orange; font: bold 20px; color: black")
146.             #menampilkan sebuah message box dengan information
147.             notif.information(self, "hasil", "Hasil Perhitungan
BMI Anda: " + str("%.2f" %BMI) + "\nCategory: Berat Badan Anda Kurang", no
tif.Ok)
148.
149.             #jika nilai 18.5<=BMI<=22.9
150.             elif 18.5<=BMI<=22.9:
151.                 #button hasil akan diubah menjadi background berwarna
a hijau dan font bold berukuran wopx dan berwarna hitam
152.                 self.hasil.setStyleSheet("background-
color: green; font: bold 20px; color: black")
153.                 #menampilkan sebuah message box dengan information
154.                 notif.information(self, "hasil", "Hasil Perhitungan
BMI Anda: " + str("%.2f" %BMI) + "\nCategory: Berat Badan Anda Normal", no
tif.Ok)
155.
156.                 #jika nilai 23<=BMI<=29.9
157.                 elif 23<=BMI<=29.9:
158.                     #button hasil akan diubah menjadi background berwarna
a kuning dan font bold berukuran wopx dan berwarna hitam
159.                     self.hasil.setStyleSheet("background-
color: yellow; font: bold 20px; color: black")
160.                     #menampilkan sebuah message box dengan information
161.                     notif.information(self, "hasil", "Hasil Perhitungan
BMI Anda: " + str("%.2f" %BMI) + "\nCategory: Berat Badan Anda Cenderung O
besitas", notif.Ok)
162.
163.                     #jika nilai 23<=BMI<=29.9
164.                     elif BMI>= 30:
165.                         #button hasil akan diubah menjadi background berwarna
a merah dan font bold berukuran wopx dan berwarna hitam
166.                         self.hasil.setStyleSheet("background-
color: red; font: bold 20px; color: black")
167.                         #menampilkan sebuah message box dengan information
168.                         notif.information(self, "hasil", "Hasil Perhitungan
BMI Anda: " + str("%.2f" %BMI) + "\nCategory: Berat Badan Anda Obesitas",
notif.Ok)
169.
170.             except:
171.                 #jika pada weight_input dan height_input diisi kosong

```

```

172.         if str(self.weight_input.text()) == "" and str(self.height_input.text()) == "":
173.             #maka message box akan menampilkan sebuah warning
174.             notif.warning(self, "Maaf Inputan yang Anda Masukkan Salah", "Tolong Masukkan dengan sebuah angka! ", notif.Ok)
175.
176.             #jika pada weight_input diisi kosong
177.             elif str(self.weight_input.text()) == "" and str(self.height_input.text()) != "":
178.                 #maka message box akan menampilkan sebuah warning
179.                 notif.warning(self, "Masukkan Berat Badan Anda Dengan Benar!", "Tolong masukkan berat badan anda dengan sebuah angka! ", notif.Ok)
180.
181.                 #jika pada height_input diisi kosong
182.                 elif str(self.height_input.text()) == "" and str(self.weight_input.text()) != "":
183.                     #maka message box akan menampilkan sebuah warning
184.                     notif.warning(self, "Masukkan Tinggi Badan Anda Dengan Benar!", "Tolong masukkan tinggi badan anda dengan sebuah angka! ", notif.Ok)
185.                     #jika yang diisikan selain angka dan kosongana
186.                     else:
187.                         #maka message box akan menampilkan sebuah warning
188.                         notif.warning(self, "Maaf Inputan yang Anda Masukkan Salah", "Tolong Masukkan dengan sebuah angka! ", notif.Ok)
189.
190.
191.
192.     if __name__ == '__main__':
193.         #Inisialisasi PyQt
194.         app = QApplication(sys.argv)
195.         #mengatur style di window menjadi style fusion
196.         app.setStyle("fusion")
197.         #menambahkan icon pada window
198.         app.setWindowIcon(QIcon('bmi_logo.ico'))
199.         #membuat variabel ex yang berisi class BMI
200.         ex = BMI()
201.
202.         #Menentukan ukuran window dan title untuk menampilkan
203.         ex.setGeometry(100,100,500,250)
204.         #membuat judul window
205.         ex.setWindowTitle("BMI (Body Mass Index) Calculator")

```

```

206.         #menampilkan isi dari variabel ex
207.         ex.show()
208.         #membuat system exit
209.         sys.exit(app.exec_())

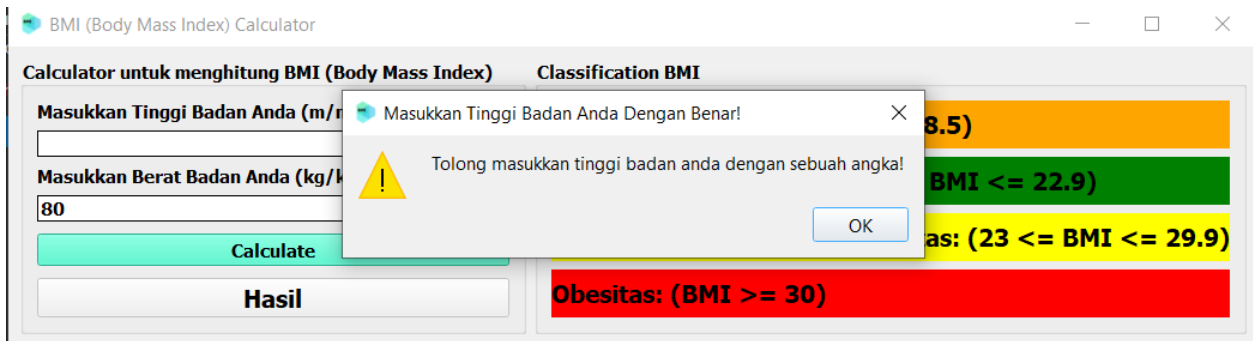
```

4. Capture hasil running!

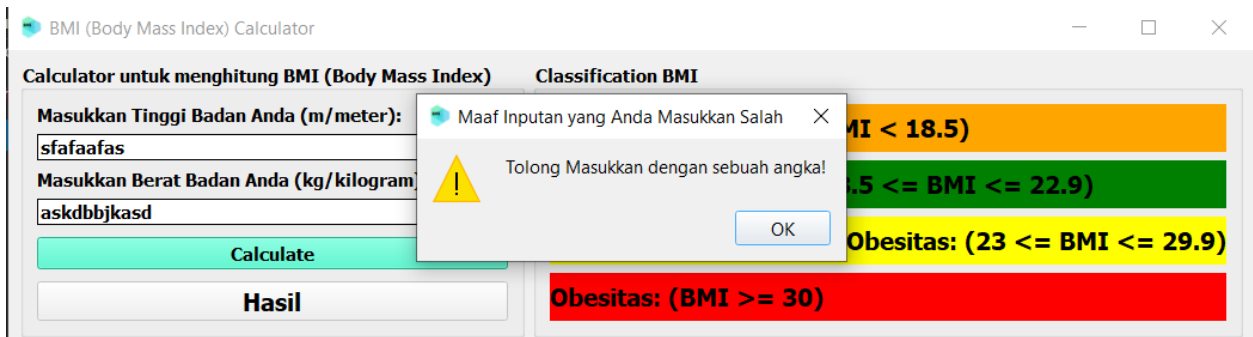
Pada saat tinggi badan dan berat badan bersisi kosongan.

Pada saat tinggi badan diisi dengan angka tetapi berat badan kosongan.

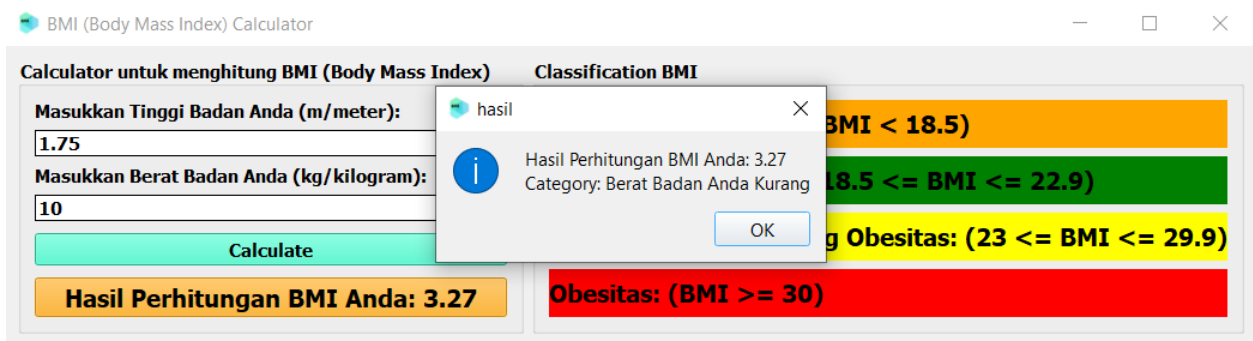
Pada saat berat badan diisi dengan angka dan tinggi badan kosong.



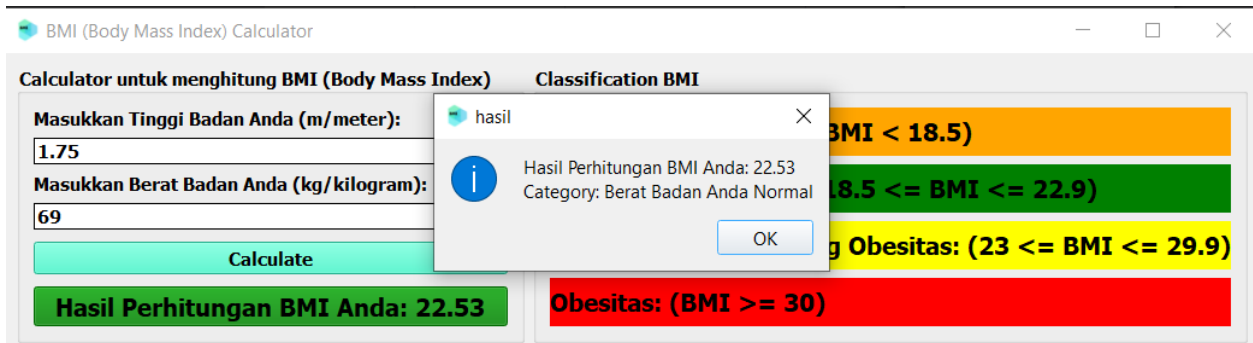
Pada saat tinggi badan dan berat badan diisi dengan selain angka.



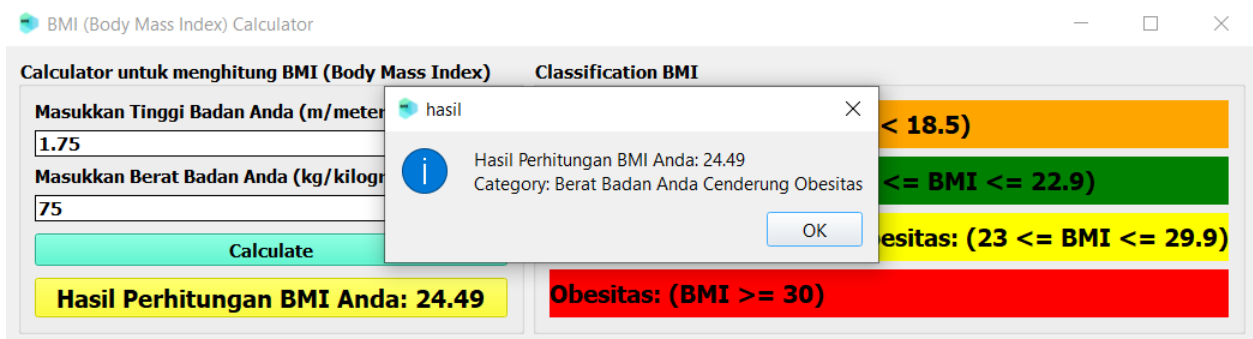
Pada saat nilai BMI < 18.5, button hasil berubah menjadi warna orange dan akan memunculkan nilai BMI user, dan akan ada notif seperti berikut.



Pada saat nilai $18.5 \leq \text{BMI} \leq 22.9$, button hasil berubah menjadi warna hijau dan akan memunculkan nilai BMI user, dan akan ada notif seperti berikut.



Pada saat nilai $23 \leq \text{BMI} \leq 29.9$, button hasil berubah menjadi warna kuning dan akan memunculkan nilai BMI user, dan akan ada notif seperti berikut.



Pada saat nilai $\text{BMI} \geq 30$, button hasil berubah menjadi warna merah dan akan memunculkan nilai BMI user, dan akan ada notif seperti berikut.

