

Pemrograman Desktop 5

Yonathan F. Hendrawan

Widget untuk Color: layout_colorwidget.py

```
from PyQt6.QtGui import QColor, QPalette
from PyQt6.QtWidgets import QWidget

class Color(QWidget):
    def __init__(self, color):
        super().__init__()
        self.setAutoFillBackground(True)

        palette = self.palette()
        palette.setColor(QPalette.ColorRole.Window, QColor(color))
        self.setPalette(palette)
```

Contoh sederhana penggunaan class Color

```
import sys
from PyQt6.QtCore import Qt
from PyQt6.QtWidgets import QApplication, QMainWindow

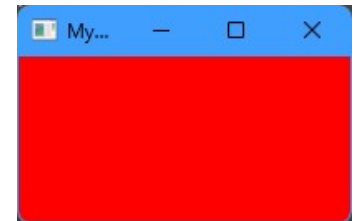
from layout_colorwidget import Color

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()

        self.setWindowTitle("My App")

        widget = Color("red")
        self.setCentralWidget(widget)

app = QApplication(sys.argv)
window = MainWindow()
window.show()
app.exec()
```



QVBoxLayout: layout vertikal

```
import sys
from PyQt6.QtWidgets import ( QApplication, QMainWindow, QVBoxLayout, QWidget)

from layout_colorwidget import Color

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("My App")
        layout = QVBoxLayout()
        layout.addWidget(Color("red"))
        layout.addWidget(Color("green"))
        layout.addWidget(Color("blue"))

        widget = QWidget()
        widget.setLayout(layout)
        self.setCentralWidget(widget)

app = QApplication(sys.argv)
window = MainWindow()
window.show()
app.exec()
```



QHBoxLayout: layout horizontal

```
import sys
from PyQt6.QtWidgets import QApplication, QHBoxLayout, QLabel, QMainWindow, QWidget

from layout_colorwidget import Color

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("My App")
        layout = QHBoxLayout()
        layout.addWidget(Color("red"))
        layout.addWidget(Color("green"))
        layout.addWidget(Color("blue"))

        widget = QWidget()
        widget.setLayout(layout)
        self.setCentralWidget(widget)

app = QApplication(sys.argv)
window = MainWindow()
window.show()
app.exec()
```



Gabungan layout

```
layout1 = QHBoxLayout()  
layout2 = QVBoxLayout()  
layout3 = QVBoxLayout()  
  
layout2.addWidget(Color("red"))  
layout2.addWidget(Color("yellow"))  
layout2.addWidget(Color("purple"))  
  
layout1.addLayout(layout2)  
  
layout1.addWidget(Color("green"))  
  
layout3.addWidget(Color("red"))  
layout3.addWidget(Color("purple"))  
  
layout1.addLayout(layout3)
```



Grid layout

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

			0,3
	1,1		
		2,2	
3,0			

QGridLayout with unfilled slots.

Grid layout

```
layout = QGridLayout()
```

```
layout.addWidget(Color("red"), 0, 0)
```

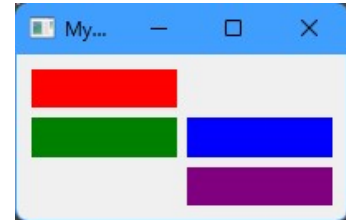
```
layout.addWidget(Color("green"), 1, 0)
```

```
layout.addWidget(Color("blue"), 1, 1)
```

```
layout.addWidget(Color("purple"), 2, 1)
```

```
widget = QWidget()
```

```
widget.setLayout(layout)
```



Nested (QStackedLayout) layout

- Layout bertumpuk
- Layout terbatas yang tampil
- Digunakan dalam mekanisme tab
- Tidak dibahas, langsung ke tab widget

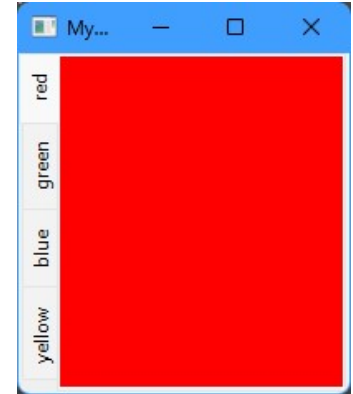


Tab Widget

```
tabs = QTabWidget()  
tabs.setTabPosition(QTabWidget.TabPosition.West)  
tabs.setMovable(True)
```

```
for n, color in enumerate(["red", "green", "blue", "yellow"]):  
    tabs.addTab(Color(color), color)
```

```
self.setCentralWidget(tabs)
```



UTS