**Pendokumentasian Drawing RGB Untuk Tugas Ujian Akhir Semester**

**“ PEMEROGAMAN BERORIENTASI OBYEK”**



**Disusun Oleh :**

**KELOMPOK**

TANGGAL : 08 Desember 2018

NAMA : Mohammad Awaliza

NRP : 140411100004

NAMA : Arief Wicaksono

NRP : 140411100008

\

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**2018**

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

Prinsip pemrograman berorientasi objek adalah membagi modul program yang besar menjadi kumpulan objek-objek yang saling berinteraksi, dengan prinsip seperti ini maka pemrograman berorientasi objek lebih menekankan pada proses yang terjadi antar objek. Sehingga dikemudian akan terdapat kemungkinan sebuah program merupakan kumpulan objek-objek yang saling bertukar informasi dan data dan memiliki fungsi dan tanggung jawab masing-masing.

Kelas dan objek merupakan satuan yang berbeda, Kelas Merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu dimana didalamnya terdapat kumpulan atribut dan method, contoh kelas : binatang, kendaraan, benda dsb.

1. **Tujuan**
2. Mahasiswa dapat memahami karakteristik OOP
3. Mahasiswa dapat membuat kelas project OOP
4. Mahasiswa dapat membuat karya software secara berkelompok berbasis OOP

**BAB II**

**DASAR TEORI**

* 1. **Teori**
  2. **Apa itu kivy ?**

Kini Python merupakan salah satu bahasa pemrograman yang populer digunakan oleh developer dalam mengembangkan aplikasi web, web, aplikasi perkantoran, simulasi, dan masih banyak lagi. Python sendiri dapat populer karena disebabkan oleh bahasa pemrogramannya yang mudah dipahami dan dinamis.

Selain itu, berdasarkan tren masa kini dimana developer dapat mengembangkan aplikasi multiplatform dalam sekali langkah, Python juga kini memiliki beberapa alat pengembangan maupun framework untuk mengembangkan aplikasi mobile secara multiplatform. Bagi pembaca yang tertarik untuk mengembangkan aplikasi mobile secara multiplatform menggunakan bahasa pemrograman Python, ada baiknya mencoba salah satu framework yang bernama Kivy.

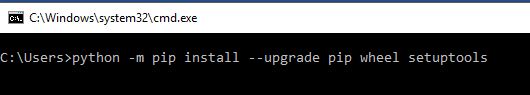
Kivy ini sendiri merupakan framework yang dibangun menggunakan library dari bahasa pemrograman Python yang bersifat Open Source. Tujuan dikembangkannya framework ini agar dapat membantu developer secara cepat dalam mengembangkan aplikasi yang memiliki tampilan antarmuka inovatif seperti aplikasi yang mendukung multitouch.

## **Cara menginstall Kivy**

Buka cmd dengan cara tekan tombol “windows + r” ketik “cmd” lalu tekan enter.

### 1. Pada tampilan cmd silahkan anda ketik perintah dibawah ini lalu tekan enter

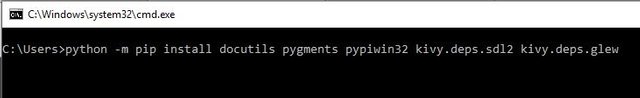
python -m pip install --upgrade pip wheel setuptools



Pastikan bahwa perangkat anda terhubung ke internet, lalu tunggu prosesnya hingga selesai, mungkin akan memakan waktu sekitar 5 menit tergantung pada kecepatan internet anda.

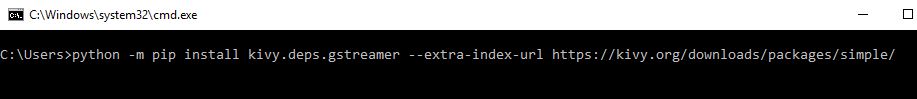
### 2. Jika langkah pertama sudah selesai ketik perintah berikut pada cmd pula, tunggu hingga proses selesai

python -m pip install docutils pygments pypiwin32 kivy.deps.sdl2 kivy.deps.glew



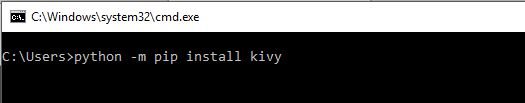
### 3. Sama seperti sebelumya

python -m pip install kivy.deps.gstreamer --extra-index-url https://kivy.org/downloads/packages/simple/

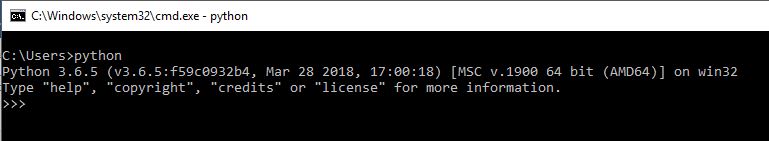


### 4. Langkah terakhir

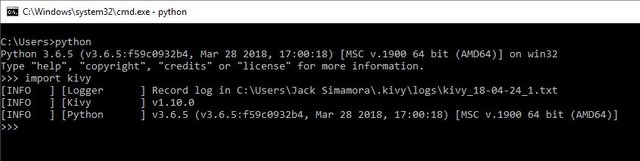
python -m pip install kivy



Untuk memastikan bahwa Kivy sudah terinstall dengan benar silahkan buka “cmd” lalu ketik python dan enter.



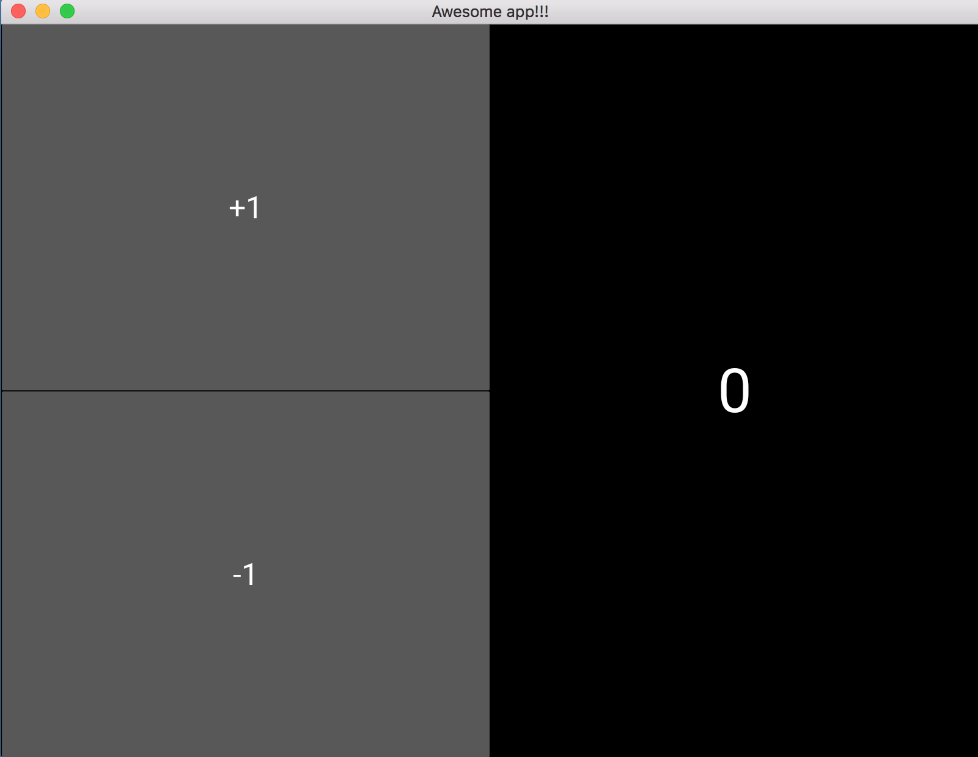
Ketik import kivy lalu enter



Jika kivy sudah terinstall dengan benar maka tampilannya akan sedikit seperti gambar yang diatas

**Kivy(Implementasi OOP dengan Python dan Kivy)**

* cara untuk membuat aplikasi Android dengan Python adalah ... apa yang dapat Anda lakukan dengannya? Apakah cukup berkinerja untuk game, dapatkah Anda memanggil API Android, apakah semua aplikasi tampak sama?
* aplikasi mobile secara multiplatform menggunakan bahasa pemrograman Python
  + official home page: https://kivy.org/#home
  + Download kivy: https://kivy.org/#download
  + Instalation:https://steemit.com/aceh/@alucard14/cara-membuat-aplikasi-android-python-and-kivy-1-or-install-program-python-and-kivy
  + Dokumentation:https://www.codementor.io/kiok46/beginner-kivy-tutorial-basic-crash-course-for-apps-in-kivy-y2ubiq0gz
  + Exampleproject : https://kivy.org/doc/stable/examples/gallery.html
* **Contoh Tutorial Game**   
  Konten Posting ini akan mencakup aplikasi yang sangat dasar pemula mutlak di Kivy) di mana Anda hanya menambahkan (+1) atau mengurangi (-1) menggunakan 2 tombol dan memperbarui nilai baru ke label. Pada akhir tutorial ini, Anda akan memiliki (saya harap) beberapa pengetahuan dan pemahaman yang baik tentang cara bekerja dengan Kivy. Berikut adalah beberapa hal yang akan kami bahas dalam posting ini:
* Memulai
* Metode App () & GridLayout secara detail
* Apa itu metode membangun dan apa fungsinya?
* Penataan aplikasi anda
* Apa itu on\_release dan id?
* Properti objek



### **Mari kita mulai**

Baiklah, mari kita mulai dengan jendela hitam yang tidak melakukan apa-apa. Buat file main.py.

**from** kivy.app **import** App

**from** kivy.uix.gridlayout **import** GridLayout

**class** **Container**(GridLayout):

**pass**

**class** **MainApp**(App):

**def** **build**(self):

self.title = 'Awesome app!!!'

**return** Container()

**if** \_\_name\_\_ == "\_\_main\_\_":

app = MainApp()

app.run()

**dan buat file main.kv**

<Container>:

rows: 1

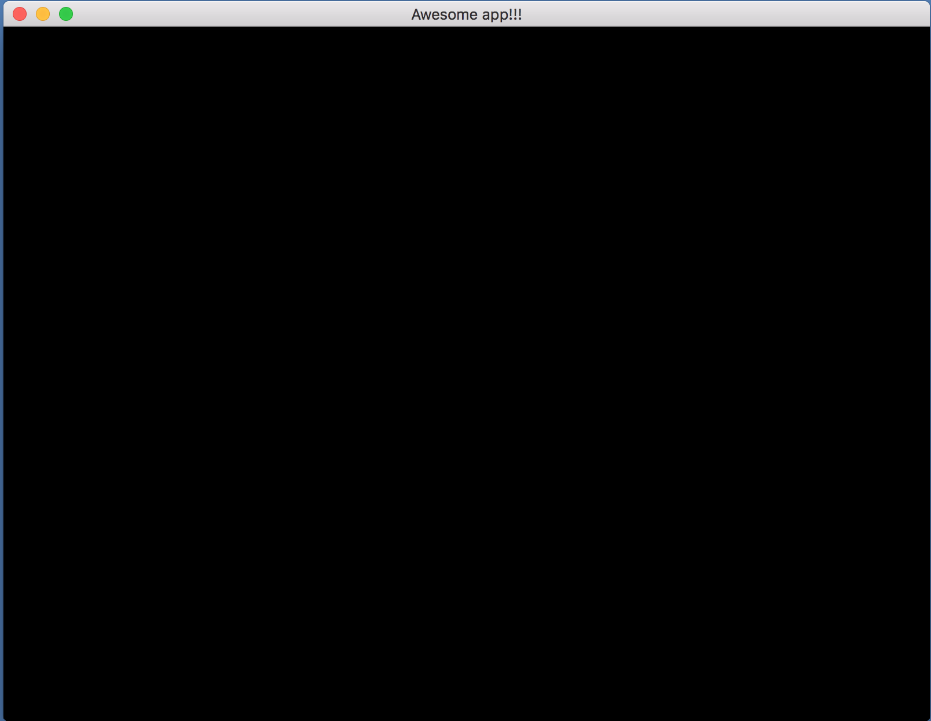
**Current file structure**:

Basic\_app/

- main.py

- main.kv

**Dan anda membuat.**



**BAB III**

**TUGAS**

* 1. **Soal**

1. Mencoba Kivy dan Memahami Kivy Menjalankan project kivy yang sudah ada ditambahkan dengan mendokumentasikan Meng-customize project kivy di sertai dengan lampirkan referensi sumber

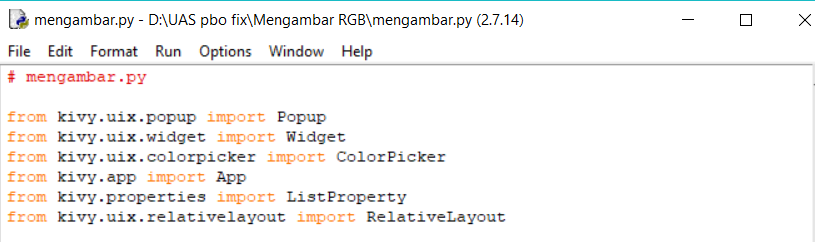
Keterangan:

Sebelum memulai ke pemerogaman saya ingin menjelaskan bahwa saya mengunakan sebuah software Python dengan liberary sebuah Kivy

Dengan tujuan agar mempermudah saya untuk membuat program dengan konsep OOP dengan terstruktur dan untuk mempermudah tentunya dan dalam proses pendokumentasian ini hanya berfokus kepada penjelasan konsep OOP nya saja, untuk code programnya saya sertakan dalam Gambar, Dalam project ini kami membuat sebuah aplikasi MengambarRGB menunakan konsep OOP

* 1. **Hasil Code Game Snake**

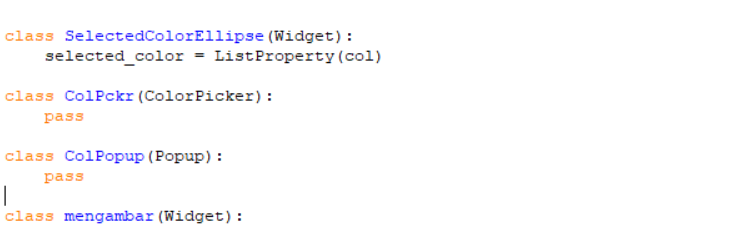
**1 Langkah pertama :**

****

**Gambar 1**

Importlah sebuah liberary Kivy.uix. popup liberary ini berguna untuk menampilkan sebuah popup selanjutnya,kemudian import juga widgetnya seperti pada gambar di atas liberary ini berguna untuk tampilan aksi, import juga color pickernya,app,propertiesmdan relativelayour. Jika sudah kita lanjur ke tahap selanjutnya.

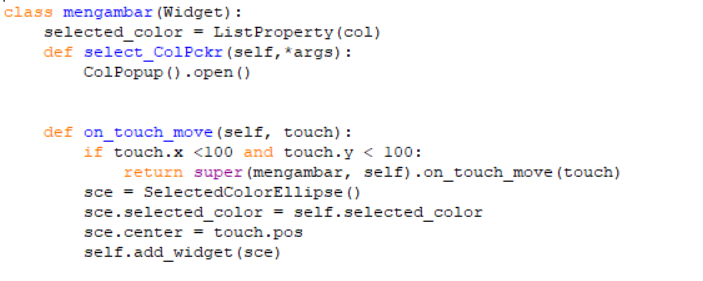
**2 Langkah kedua :**

****

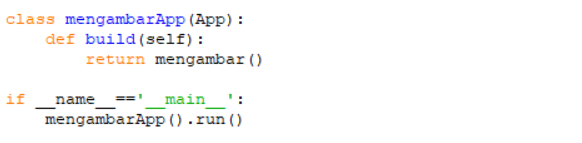
Selanjutnya kita harus membuat kelas terserah anda mau menamai apa namun dalam project ini,kami membuat selectedcolorellipse berfungsi untuk mengunakan fungsi widget dengan listproperti.

Buat kelas colpckr,colpopup untuk kelas ini kita gunakan default saja maka kita tulis pass, kemudian kita buat kelas mengambar dan mengeksten widget/ kelas selectedcolorellipse agar program kita berbasi OOP. Langkah berikutnya

**3 Langkah ketiga :**

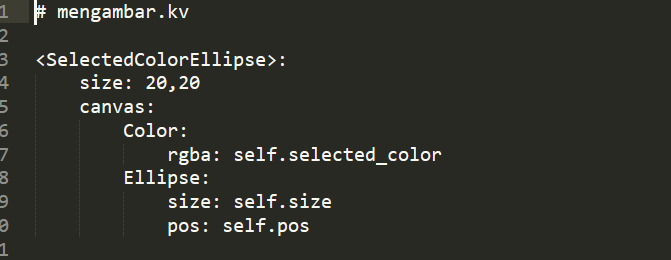


Untuk langkah ini kita hanya fokus pada kelas mengambar saja kita inisialisasikan selected\_color sama dengan list\_property, kemudian fungsi def select\_colorpckr tadi dengan colpopup untuk mengarahkan atau membuka layout pada disain colorpacker pada file .kv serta masuk ke liberary warna RGB, kemudian kita buat on\_touch\_move berfungsi sebagai aksi sentuh pada saat mengambar.

**4 Langkah keempat:**

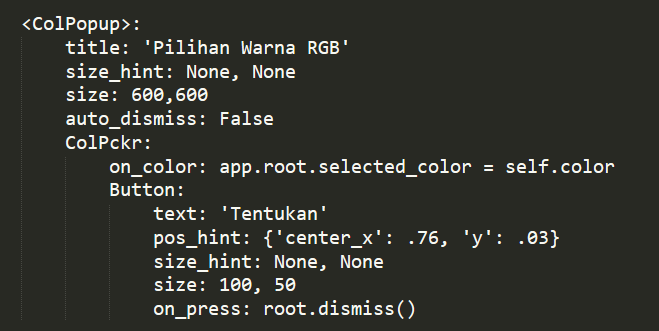
Untuk langkah yang ke empat ini kita membuat fungsi berfungsi sebagai fungsi main progam atau runing program seperti kebanyakan kita buat class mengambarapp dengan nilai kembalian atau menampung kelas mengambar tadi kemudian kita paggil di fungsi main run.

**5 Langkah kelima:**



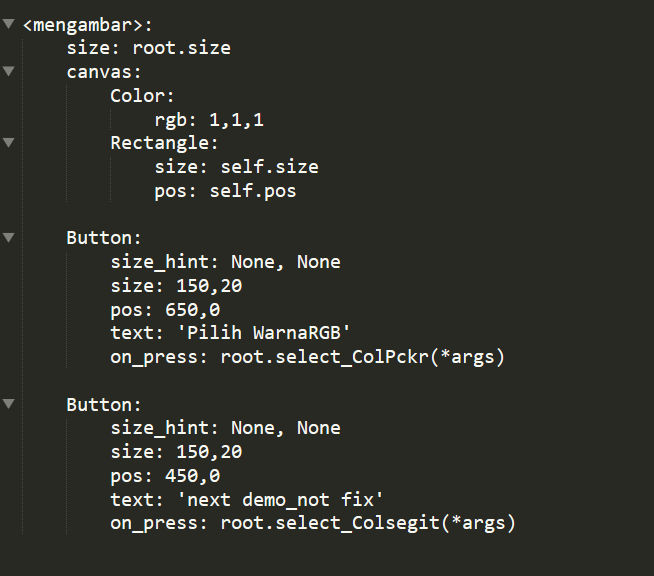
Langkah selanjutnya ini adalah sebuah tampilan disign yang ber ekstensi. Kv,code di atas membuat tampilan lebih spesifik ke gambar atau ukuran pena yang dapat di ubah di size, dan penerapan warna dari pena sesuai yg di pilih.

**6 Langkah keenam :**

****

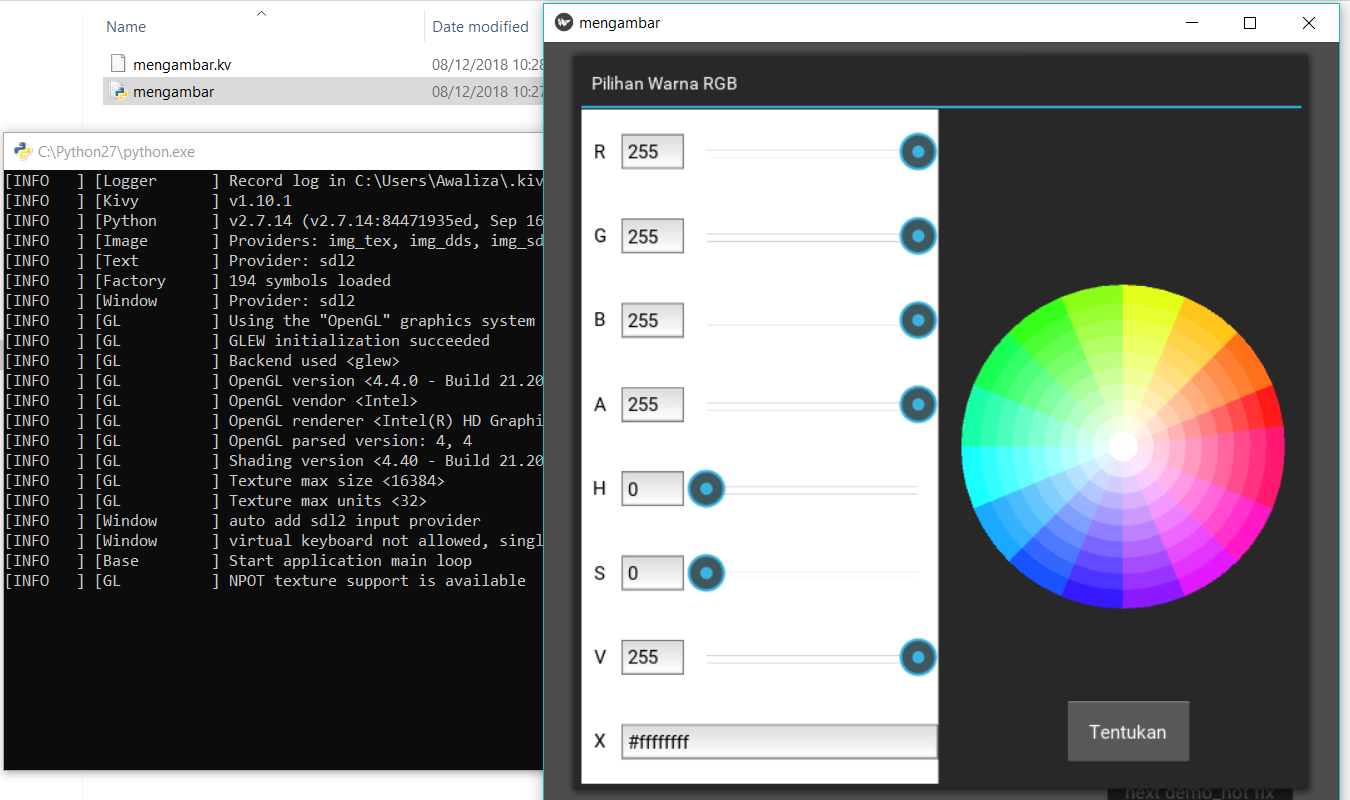
Dilangkah ini menerapkan tentang saat pilihan warna di klik akan keluar pilihan warna RGB, nah ini adalah codenya yang memanggil liberary pickercolor juga,setalah itu di tentukan berapa ukuranya dimana tempat atau posisinya dll.

**7 Langkah ketujuh :**

****

Selanjutnya langkah ini mendisign tentang warna dari kanvas letak tombol tombol, serta mengeksten atau singkron dengan kelas mengambar.

**HASIL RUN: **

****

**Pendokumentasian OOP Kivy**

**“Mengambar RGB”**

|  |
| --- |
| **Mengambar.py** |

|  |
| --- |
| **SelectedColorEllipse** |
| **+** selected\_color |
| **+** ListProperty() |
| **+** widget |

|  |
| --- |
| **ColPckr** |
| + Pass(default) |

|  |
| --- |
| **ColPopup** |
| + pass(default) |

|  |
| --- |
| **Mengambar** |
| + select\_ColPckr() |
| + on\_touch\_move() |

|  |
| --- |
| **mengambarApp** |
| **+**  build(self) |

**Penjelasan Class Dan Metod**

* Class SelectedColorEllipse:

#sebagai penetapan ListProperty

class SelectedColorEllipse(Widget):

selected\_color = ListProperty(col)

* Class ColorPicker

#sebagai inisialisasi color picker

class ColPckr(ColorPicker):

pass

* Class ColPopup

#sebagai inisialisasi Popup

class ColPopup(Popup):

pass

* Class mengambar

#sebagai kelas utama yang didalamnya memanggil class SelectedColorEllipse, ColPckr, ColPckr

class mengambar(Widget):

selected\_color = ListProperty(col)

def select\_ColPckr(self,\*args):

ColPopup().open()

def on\_touch\_move(self, touch):

if touch.x <100 and touch.y < 100:

return super(mengambar, self).on\_touch\_move(touch)

sce = SelectedColorEllipse()

sce.selected\_color = self.selected\_color

sce.center = touch.pos

self.add\_widget(sce)

* Class mengambarApp

#sebagai main program (fungsi run)

class mengambarApp(App):

def build(self):

return mengambar()

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

mengambarApp().run()

**Contoh reverensi Sumber**

|  |
| --- |
| **# ex41.py**  **from kivy.uix.screenmanager import ScreenManager, Screen**  **from kivy.app import App**  **from kivy.properties import ListProperty**  **class ScreenCircle(Screen):**  **pass**  **class ScreenSquare(Screen):**  **pass**  **class ScreenColor(Screen):**  **pass**  **class Ex41(ScreenManager):**  **selected\_color = ListProperty([1,0,0,1])**  **class Ex41App(App):**  **def build(self):**  **return Ex41()**  **if \_\_name\_\_=='\_\_main\_\_':**  **Ex41App().run()** |

|  |
| --- |
| **# ex41.kv**  **#:import RiseInTransition kivy.uix.screenmanager.RiseInTransition**  **<ScreenColor>:**  **name: 'screen\_color'**  **ColorPicker:**  **color: root.manager.selected\_color**  **on\_color: root.manager.selected\_color=self.color**  **Button:**  **text: 'Go to circle'**  **size\_hint: None,None**  **size: 100,100**  **pos: 500,0**  **on\_press: root.manager.current = 'screen\_circle'**  **Button:**  **text: 'Go to square'**  **size\_hint: None,None**  **size: 100,100**  **pos: 600,0**  **on\_press: root.manager.current = 'screen\_square'**  **<ScreenCircle>:**  **name: 'screen\_circle'**  **canvas:**  **Color:**  **rgba: root.manager.selected\_color**  **Ellipse:**  **size: 250,250**  **pos: 150,150**  **Button:**  **text: 'Go to square'**  **size\_hint: None, None**  **size: 100,100**  **pos: 0,0**  **on\_press: root.manager.current = 'screen\_square'**  **Button:**  **text: 'Select Color'**  **size\_hint: None,None**  **size: 100,100**  **pos: 300,0**  **on\_press: root.manager.current = 'screen\_color'**  **<ScreenSquare>:**  **name: 'screen\_square'**  **canvas:**  **Color:**  **rgba: root.manager.selected\_color**  **Rectangle:**  **size: 250,250**  **pos: 150,150**  **Button:**  **text: 'Go to circle'**  **size\_hint: None,None**  **size: 100,100**  **pos: 0,0**  **on\_press: root.manager.current = 'screen\_circle'**  **Button:**  **text: 'Select Color'**  **size\_hint: None,None**  **size: 100,100**  **pos: 300,0**  **on\_press: root.manager.current = 'screen\_color'**  **<Ex41>:**  **canvas:**  **Color:**  **rgb: 1,1,1**  **Rectangle:**  **size: self.size**  **pos: self.pos**  **transition: RiseInTransition()**  **ScreenCircle:**  **ScreenSquare:**  **ScreenColor:** |

**Reverensi sumber:**

**https://www.youtube.com/watch?v=QS1X-NiyGgg**

**BAB IV**

**PENUTUP**

**Kesimpulan**

Pemrograman berorientasi objek (*object-oriented programming* disingkat OOP) merupakan [paradigma pemrograman](https://id.wikipedia.org/wiki/Paradigma_pemrograman) yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika [pemrograman terstruktur](https://id.wikipedia.org/wiki/Pemrograman_terstruktur). Setiap objek dapat menerima [pesan](https://id.wikipedia.org/wiki/Pesan), memproses data, dan mengirim pesan ke objek lainnya,

Kelas dan objek merupakan satuan yang berbeda, Kelas Merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu dimana didalamnya terdapat kumpulan atribut dan method,liberary pada sebuah bahasa pemerogaman berperan sebagai pembantu untuk mempermudah dan mempersingkat sebuah program.