**UJIAN AKHIR SEMESTER**

**PEMROGRAMAN BERORIENTASI OBYEK**

**DOKUMENTASIAN SNAKE GAME**



**Disusun Oleh :**

TANGGAL : 07 Desember 2018

NAMA : As’Adur Rofiq

NRP : 140411100121

NAMA : Renaldo Fachrozi

NRP : 140411100138

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**2018**

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

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 PBO
3. Mahasiswa dapat membuat kelas project PBO
4. Mahasiswa dapat membuat game snake secara berkelompok berbasis OOP

**BAB II**

**DASAR TEORI**

* 1. **Pengertian OOP**

OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Saya ambil contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkirim pesan kepada objek yang lain.

* + 1. **Konsep OOP**
* Abstrak Class
  1. Kelas merupakan deskripsi abstrak informasi dan tingkah laku dari sekumpulan data.
  2. Kelas dapat diilustrasikan sebagai suatu cetak biru(blueprint) atau prototipe yang digunakan untuk menciptakan objek.
  3. Kelas merupakan tipe data bagi objek yang mengenkapsulasi data dan operasi pada data dalam suatu unit tunggal.
  4. Kelas mendefinisikan suatu struktur yang terdiri atas data kelas (data field), prosedur atau fungsi (method), dan sifat kelas (property).
* Encapsulation

1. Istilah enkapsulasi sebenarnya adalah kombinasi data dan fungsionalitas dalam sebuah unit tunggal sebagai bentuk untuk menyembunyikan detail informasi.
2. Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.
3. Enkapsulasi menekankan pada antarmuka suatu kelas, atau dengan kata lain bagaimana menggunakan objek kelas tertentu.
4. Contoh: kelas mobil menyediakan antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.

* Inheritance
  1. Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada.
  2. Penurunan sifat ini bisa dilakukan secara bertingkattingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.
  3. Sub kelas memungkinkan kita untuk melakukan spesifikasi detail dan perilaku khusus dari kelas supernya.
  4. Dengan konsep pewarisan, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kodekode itu.
* Polymorphism
  + 1. Polimorfisme merupakan kemampuan objekobjek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama.
    2. Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan method mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.
    3. Method overriding.
    4. Method name overloading.
  1. **Pygame**

Pygame adalah seperangkat modul Python yang dirancang untuk membuat permainan. Pygame menambahkan fungsi di atas dengan sangat baik di SDL perpustakaan. Hal ini memungkinkan Anda untuk membuat sebuah game dengan fitur yang lengkap dan sebuah program multimedia dalam bahasa python. Pygame sangat portabel dan dapat berjalan pada hampir semua platform dan sistem operasi. Pygame sendiri telah didownload jutaan kali, dan telah memiliki jutaan kunjungan ke situsnya.

Pygame ini gratis. Dan dirilis di bawah Lisensi GPL, Anda dapat membuat sebuah aplikasi yang open source, gratis, freeware, shareware, dan game komersial dengan pygame ini.

* 1. **Instalasi pygame**

Pada tutorial ini, kita akan menggunakan modul PyGame dengan Python 3.

Jadi ada dua hal utama yang harus disiapkan:

1. Python 3
2. PyGame

PyGame dapat kita install menggunakan pip dengan perintah berikut:

sudo pip install pygame

Atau bisa juga melalui apt:

sudo apt-get install python3-pygame

Maka modul pygame akan terinstal ke dalam lingkungan OS (Sistem Operasi).

Buat yang menggunakan Virtual Environtment gunakan perintah ini:

# membuat virtual env

virtualenv -p python3 pygame

# pindah ke direktori virtualenv

cd pygmae

# aktifkan virtual env

. bin/activate

# install modul pygame di dalam virtualenv

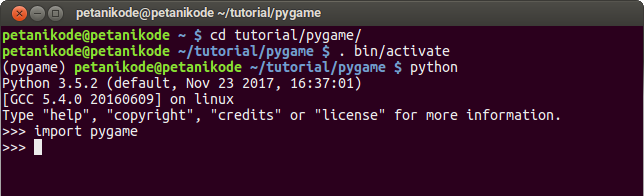
pip install pygame

Sekarang kita tes dulu, apakah modul pygame sudah terinstal atau tidak. Silahkan masuk ke *shell* python dengan mengetik perintah python3 atau python (di virtual env).

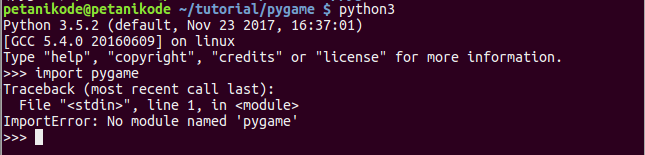
Ketik perintah berikut:

>>> import pygame

Kalau tidak terjadi error, maka modul pygame sudah terinstal dengan benar.



Tapi kalau terjadi error, artinya modul pygame belum terinstall.



Coba install lagi sampai berhasil.

**PENDOKUMENTASIAN OOP PYGAME**

**‘‘SNAKE‘‘**

|  |  |  |
| --- | --- | --- |
| **Snake** | | |
| **+** init  + posotion  + body  + direction  + changeDirectionTo | + changeDirto  - not direction left  - not direction right  - not direction down  - not direction up | + move  + posotion right  - posotion left  - posotion up  + posotion down |
| + checkCollision | + getHeadPos | + getBody |

|  |
| --- |
| **FoodSpawer** |
| **+** init  + position  + isFoodOnScreen |
| **+** spawnFood  + isFoodOnScreen  + position |
| **+** setFoodOnScreen |

|  |
| --- |
| + gameover  - quit  - exit |

**BAB III**

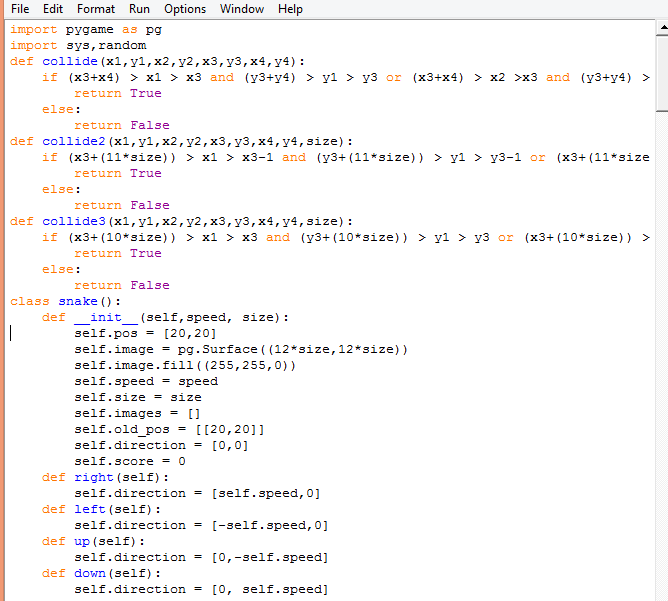
**TUGAS**

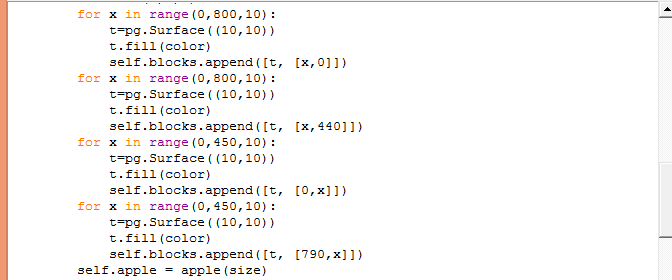
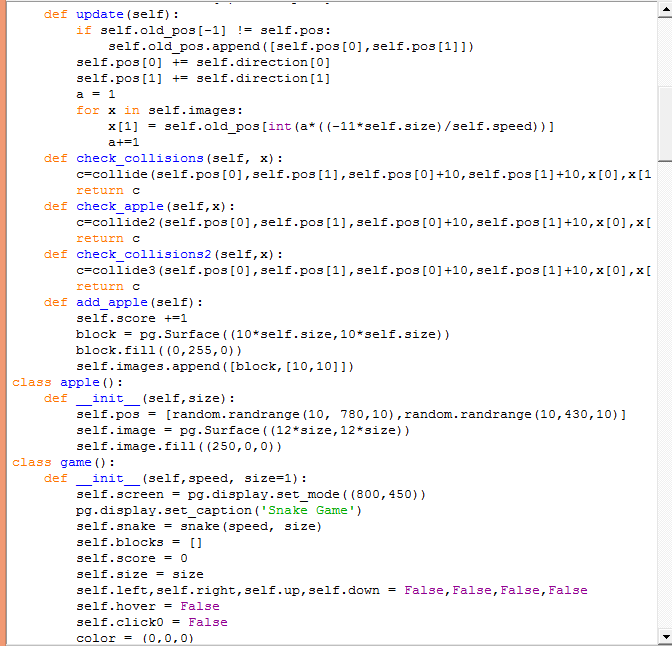
* 1. **Soal**

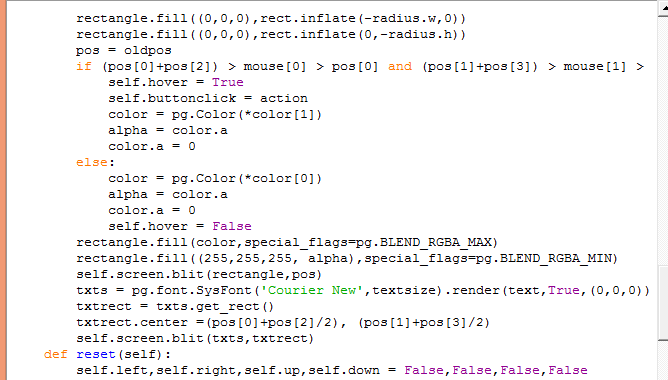
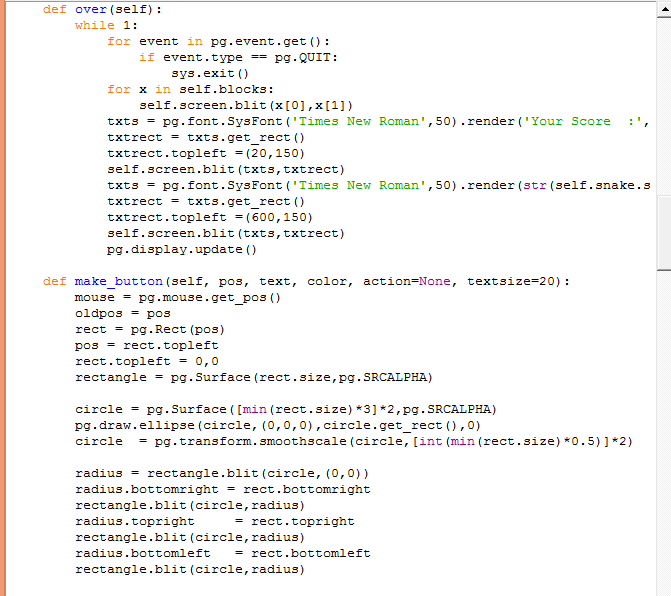
1. buatlah sebuah game yang ada pada jurnal atau materi yang telah di berikan oleh dosen dan pilihlah game mana yang akan kalian buat dan modif game tersebut sesuai kemampuan kalian berdasarkan konsep OOP dan presentasikan setiap minggunya
   1. Sebelum memulai ke pemerogaman saya ingin menjelaskan bahwa saya mengunakan sebuah software PyCharm dengan liberary sebuah pygame

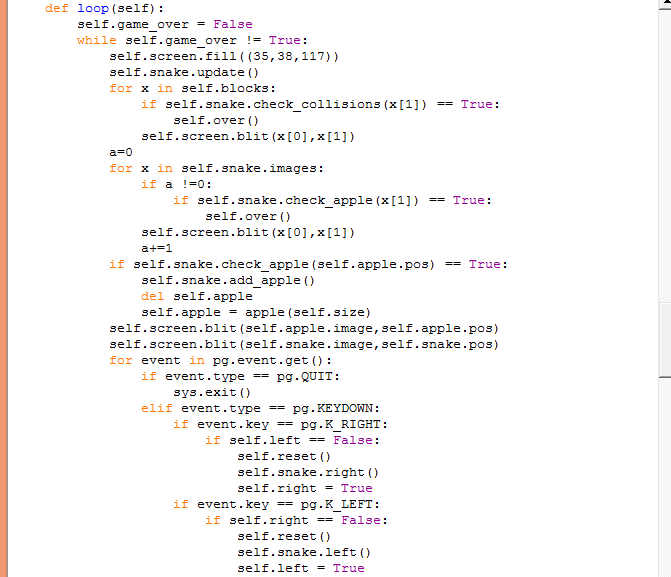
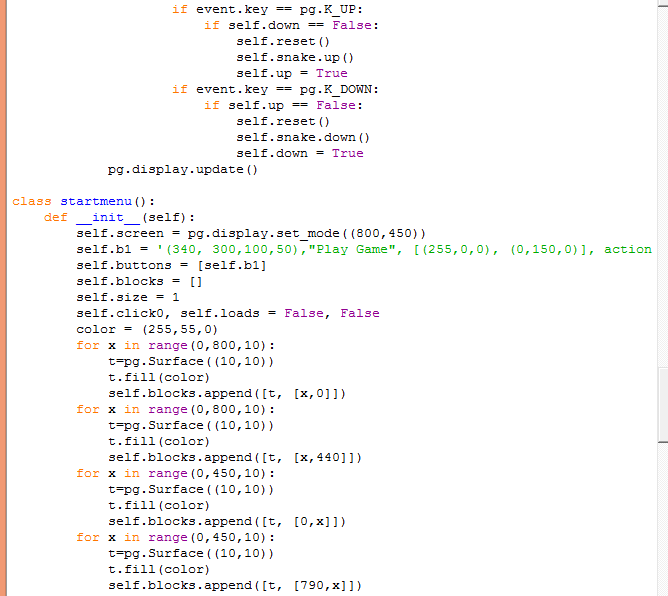
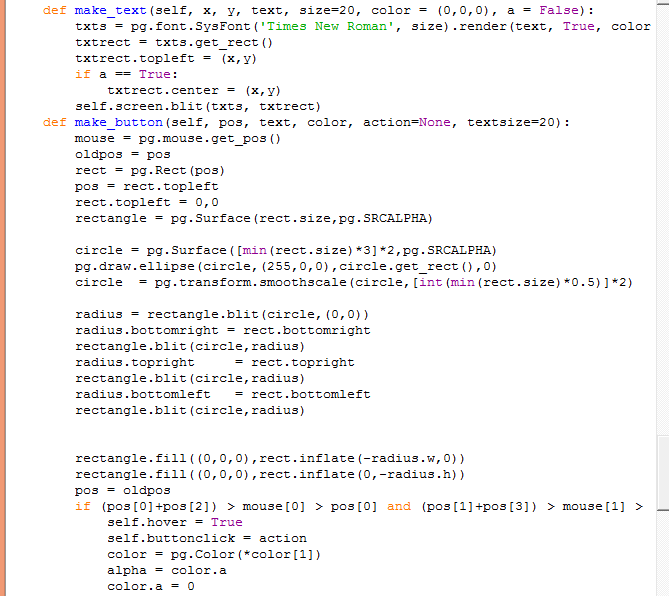
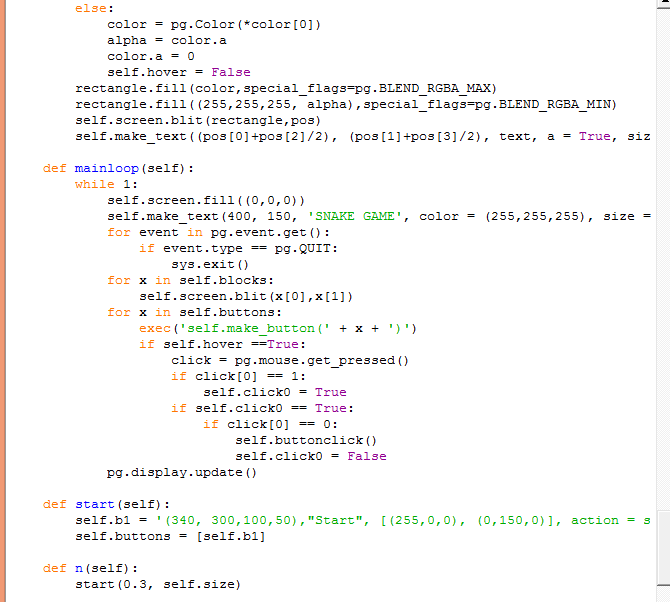
Dengan tujuan agar mempermudah saya untuk membuat program dengan konsep OOP dengan terstruktur dan untuk mempermudah tentunya dan dalam proses pendokumentasian ini hanya berfokus kepada penjelasan konsep OOP nya saja, untuk code programnya saya sertakan dalam Gambar.

* 1. **Source Code Game Snake**







**HASIL RUN:**



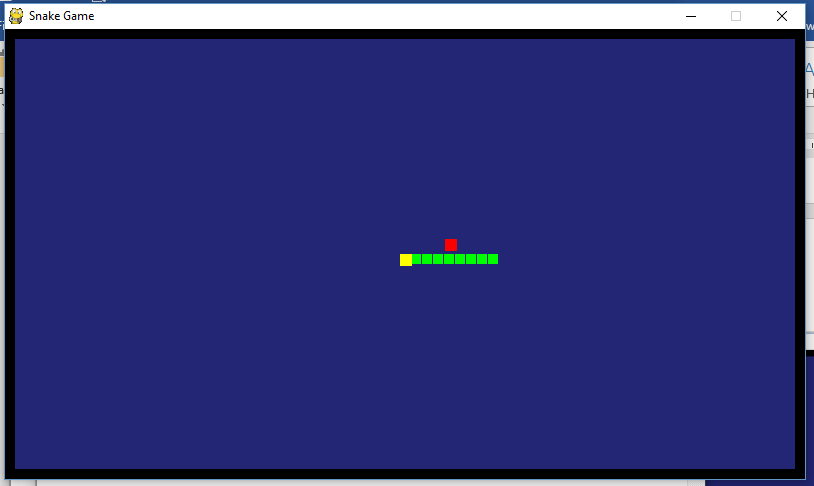
**Gambar 1 awal game ular**



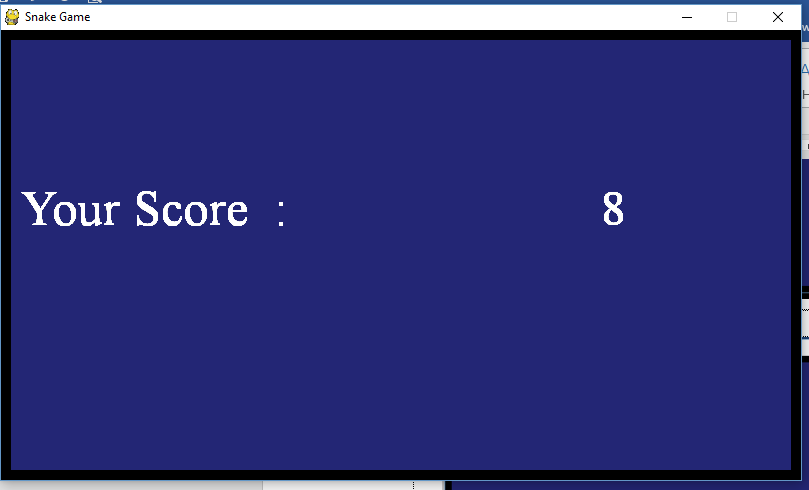
**Gambar 2 Start game ular**



**Gambar 3. Awal mulai game ular**



**Gambar 4. Saat ular di mainkan dan sedang memakan buah nya**



**Gambar 5. Hasil score dari coin yang dimakan**

**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, contoh kelas : binatang, kendaraan, benda dsb. Sedangkan objek merupakan bentuk representasi dari sebuah kelas, membungkus data dan fungsi bersama menjadi suatu unit atau entitas dalam sebuah program komputer.

**Referensi :**

https://code-projects.org/snake-game-in-python-with-source-code/

https://github.com/memoiry/Snaky

https://www.pygame.org/tags/snake

https://www.youtube.com/results?search\_query=+pygame+snake+source+code+levels

https://codereview.stackexchange.com/questions/42602/snake-game-in-c

https://github.com/martinjos/PySnake

http://www.learningpython.com/2006/03/19/creating-a-game-in-python-using-pygame-part-two-creating-a-level/

https://github.com/kharazi/angry-snake

https://python-forum.io/Thread-PyGame-Basic-Snake-game-using-OOP