**LAPORAN PROJECT**

**PEMROGRAMAN BERORIENTASI OBJEK**

**“ ANGRY BIRDS DAN STOPWATCH ”**

****

**DISUSUN OLEH :**

**ANA MAGHFIROH 170411100087**

**FITRI SUWANDARI 170411100048**

**DOSEN PENGAMPU :**

**ARIK KURNIAWATI, S.Kom., M.T.**

**NIP. 19780309 200312 2 009**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**2018/2019**

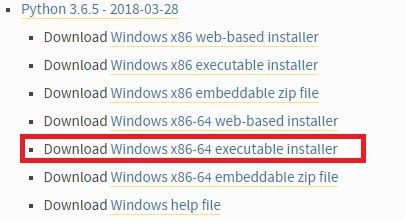
**DEFINISI PYGAME DAN KIVY**

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.

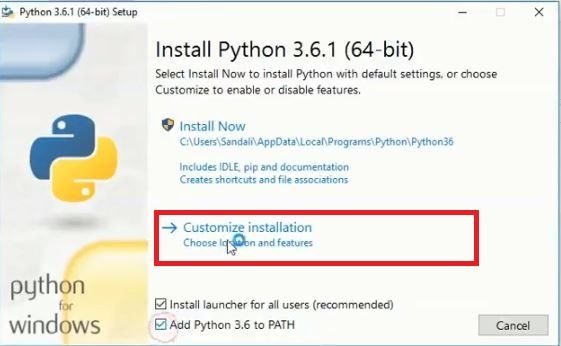
Framework ini sendiri dapat dijalankan pada sistem operasi Windows, Mac OS X, Linux, Android, hingga Raspberry Pi. Selain itu, Kivy mendukung untuk input beragam perangkat seperti WM\_Touch, WM\_Pen, Mac OS X Trackpad, Magi Mouse, Mtdev, Linux Kernel HID, dan TUIO.

## Cara menginstall Python

### 1. Download Python versi 3.6 [disini](https://www.python.org/downloads/windows/)



### 2. Buka file yang sudah diinstall tadi

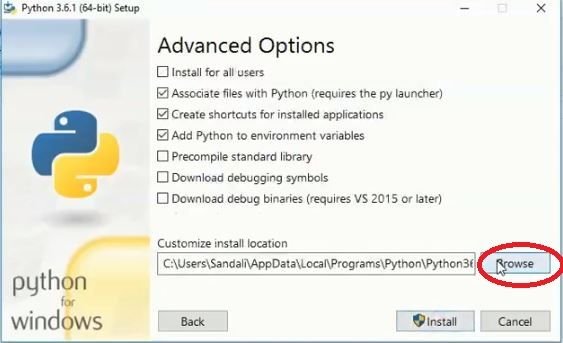


Seperti gambar yang diatas kemudian klik Customize installation

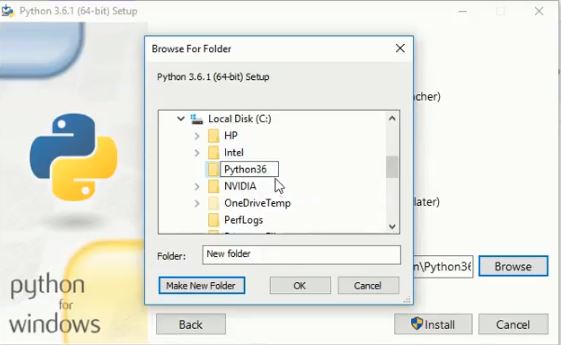
### 3. Pasitkan semua diceklis dan klik next.



### 4. Klik tombol browse



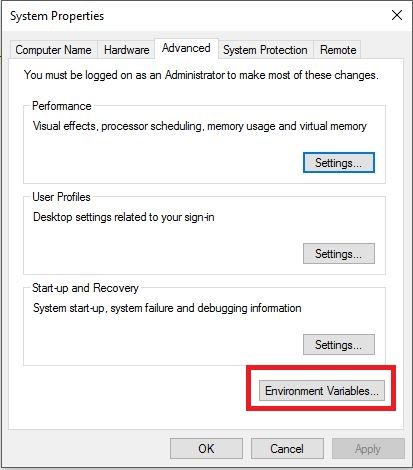
### 5. Lalu pilih Local Disk (C:), dan buat folder baru dengan nama “Python36”(tanpa tanda petik), klik ok.



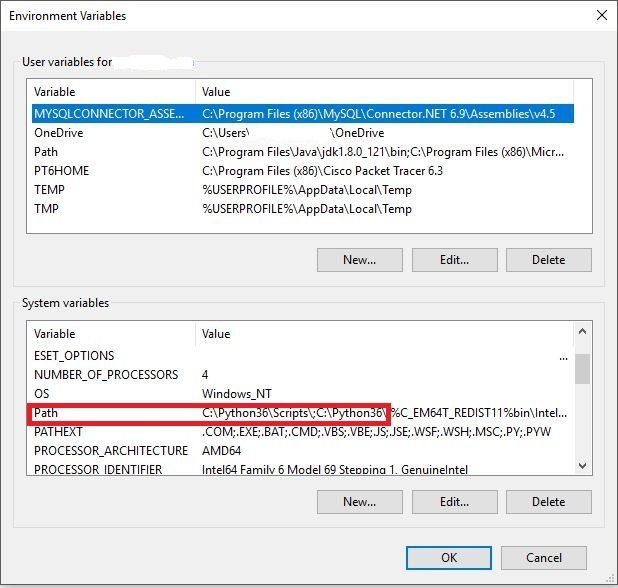
### 6. Pastikan gambarnya seperti yang dibawah, lalu klik Install



### 7. Buka contol panel, System and Security, System, Advanced system settings, lalu klik environment variables



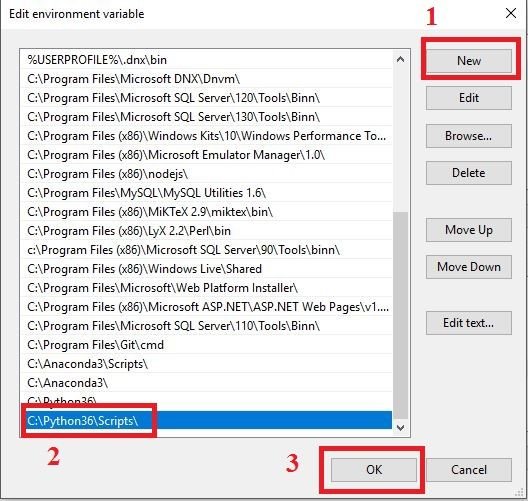
### 8. Pastikan file Python sudah ditambahkan di path seperti gambar yang dibawah



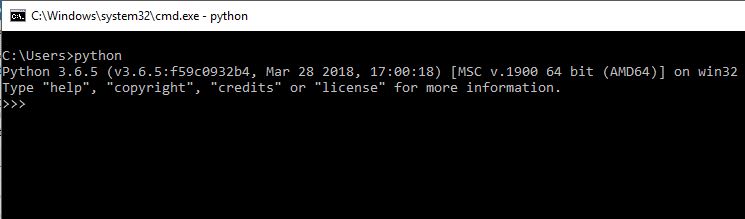
Jika belum silahkan tambahkan dengan cara klik Path, lalu klik tombol edit, klik new, dan isi seperti gambar dibawah ini



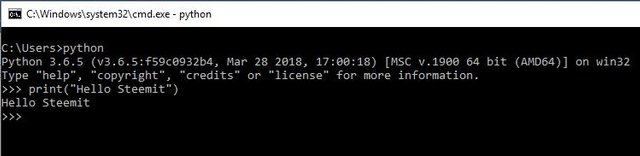
Klik new lagi dan isi seperti gambar dibawah ini



### 9. Buka cmd untuk memastikan bahwa Python sudah terinstall dengan benar, tekan tombol “windows + r” lalu ketik “cmd” dan tekan enter. Setelah cmd terbuka lalu ketik python, dan tekan enter, maka tampilannya akan seperti yang dibawah



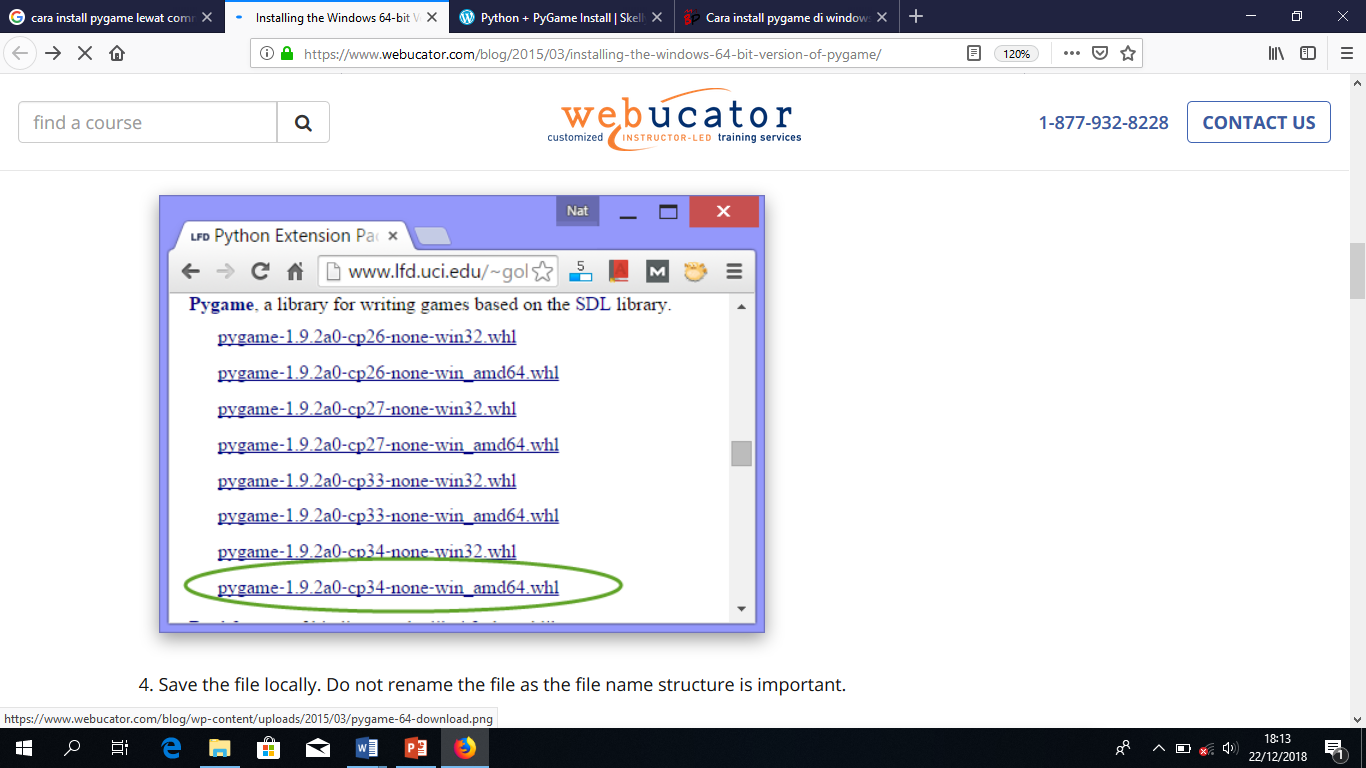
### 10. ketik print(“Hello Steemit”)



Maka outputnya adalah Hello Steemit.

**Cara install pygame**

1. download file pygame terlebih dahulu.



1. setelah itu ubah file menjadi .zip
2. Selanjutnya install pygame sesuai dengan file yang di download tadi
3. Ketika sudah selesei, cek di command prompt . tampilan pygame ketika berhasil di install

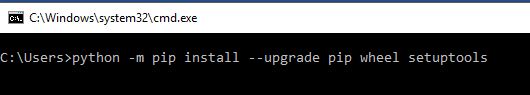


## Cara menginstall Kivy

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

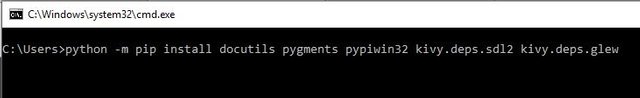
python -m pip install --upgrade pip wheel setuptools



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

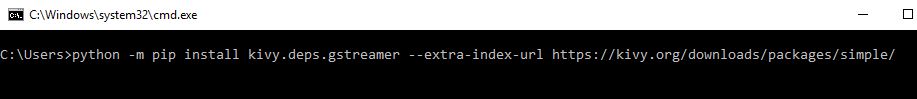
### 2. Jika langkah pertama sudah selesai ketik perintah berikut pada cmd pula, tunggu hingga proses selesai

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



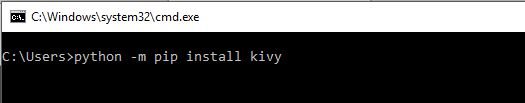
### 3. Sama seperti sebelumya

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

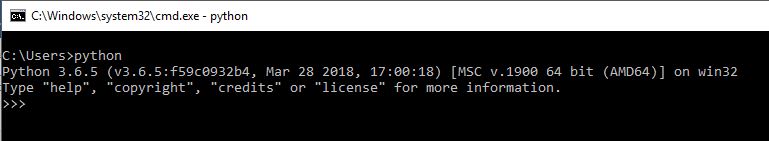


### 4. Langkah terakhir

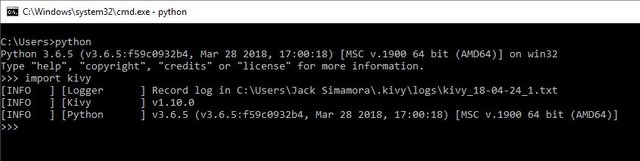
python -m pip install kivy



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

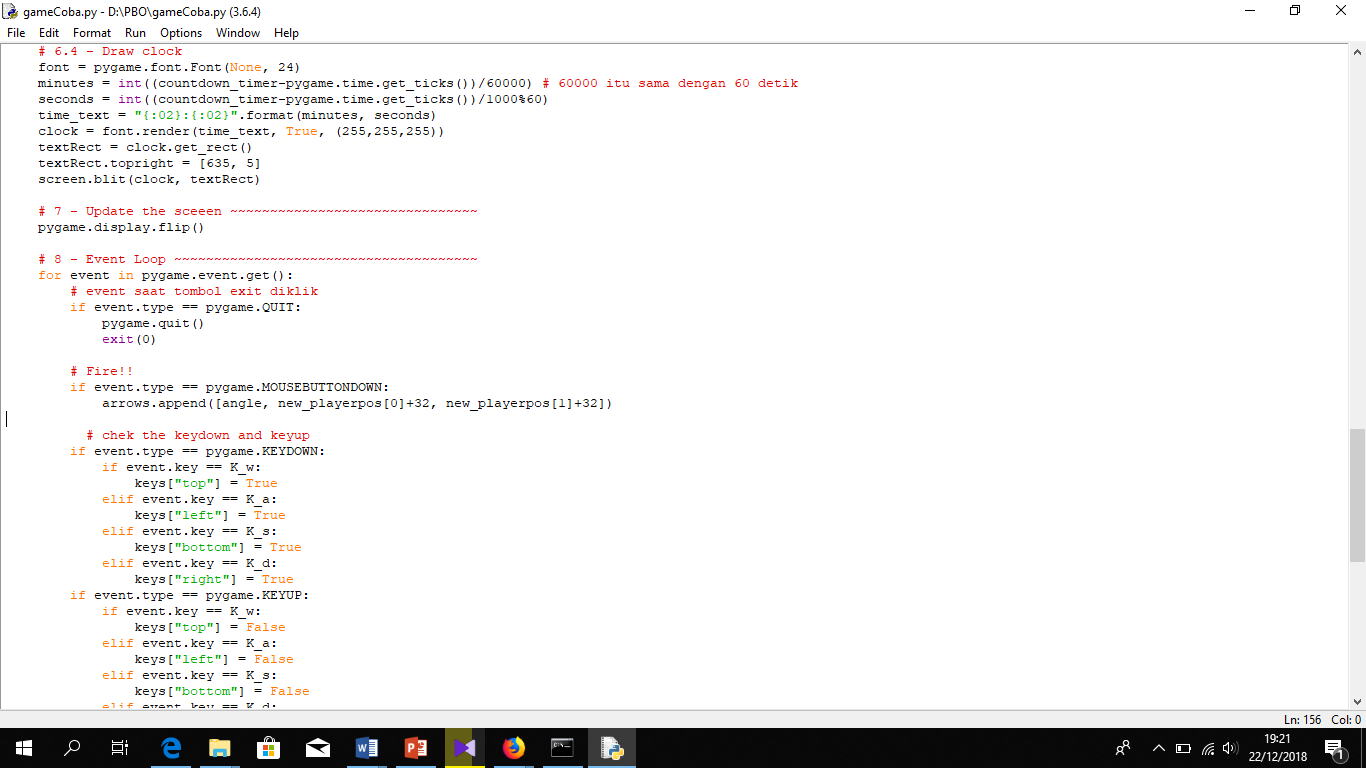
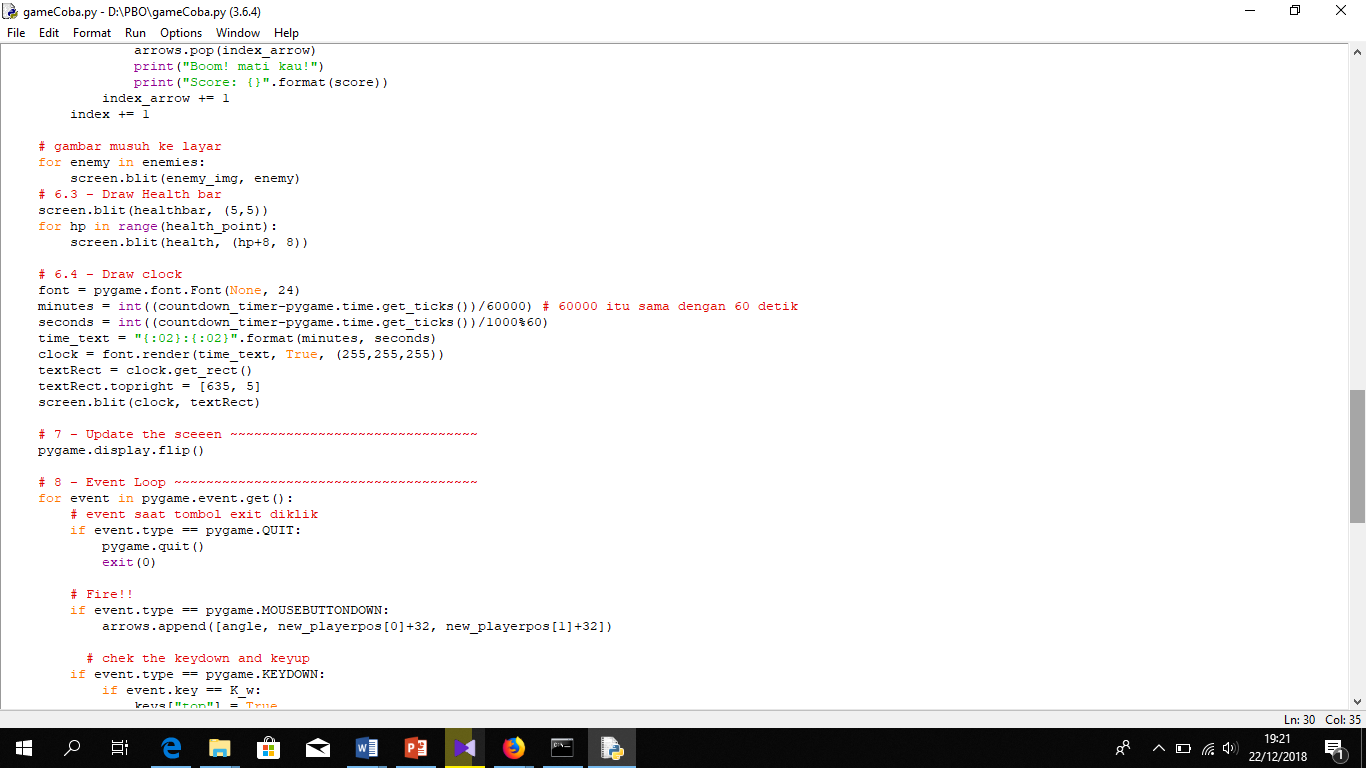
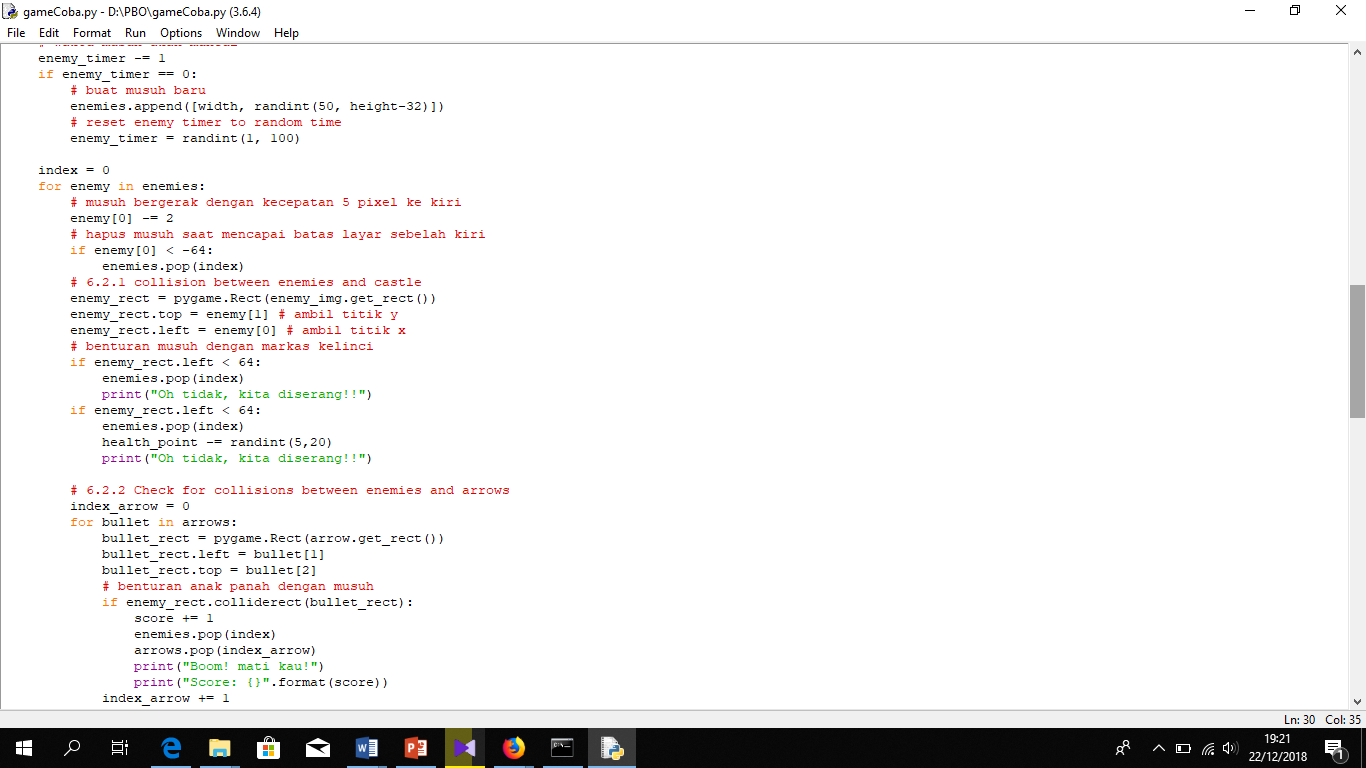
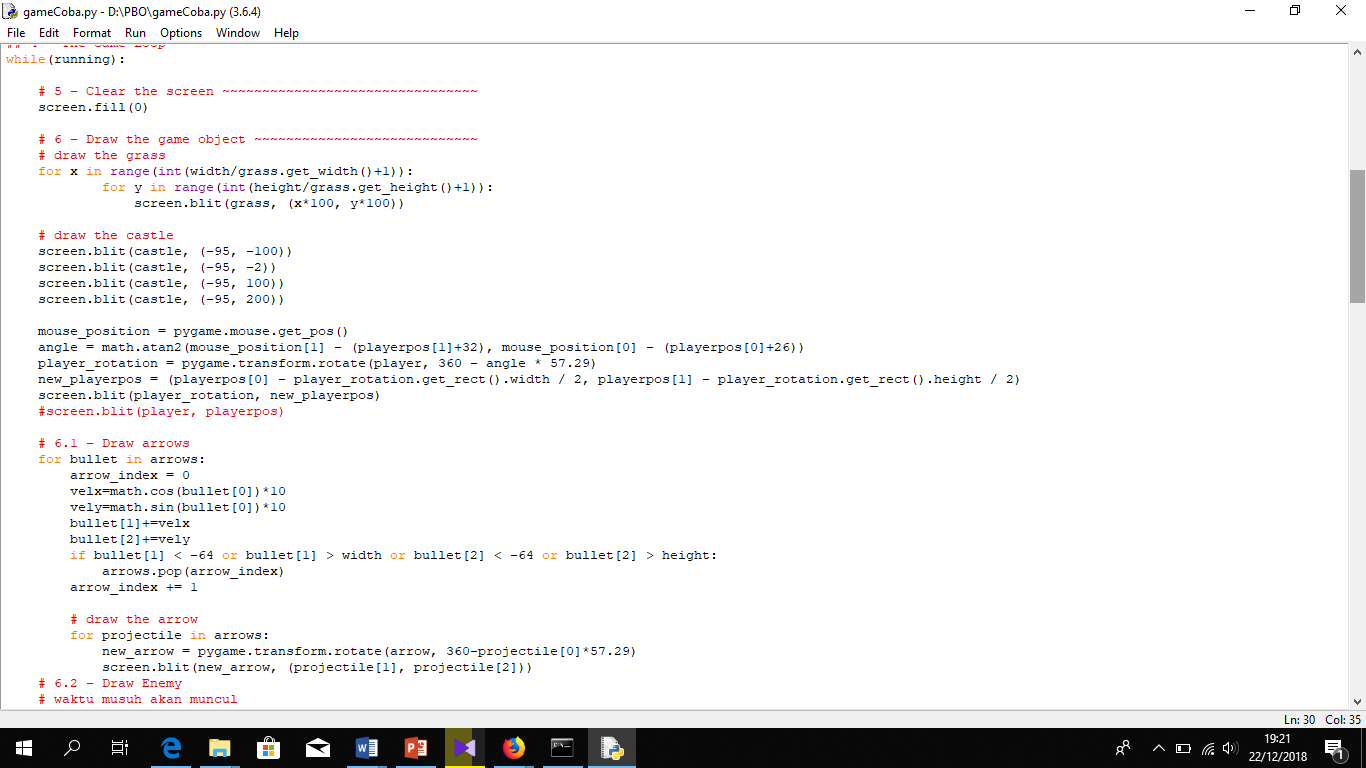
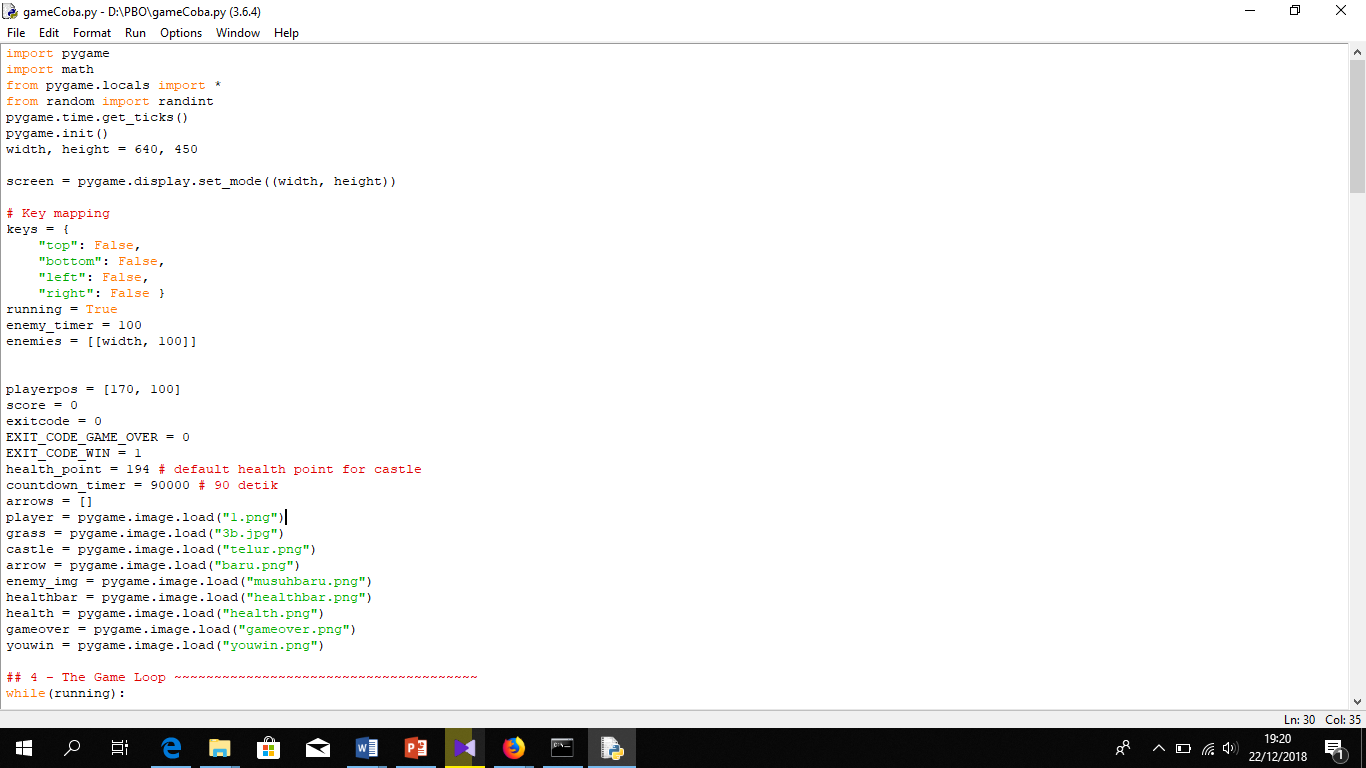


Ketik import kivy lalu enter

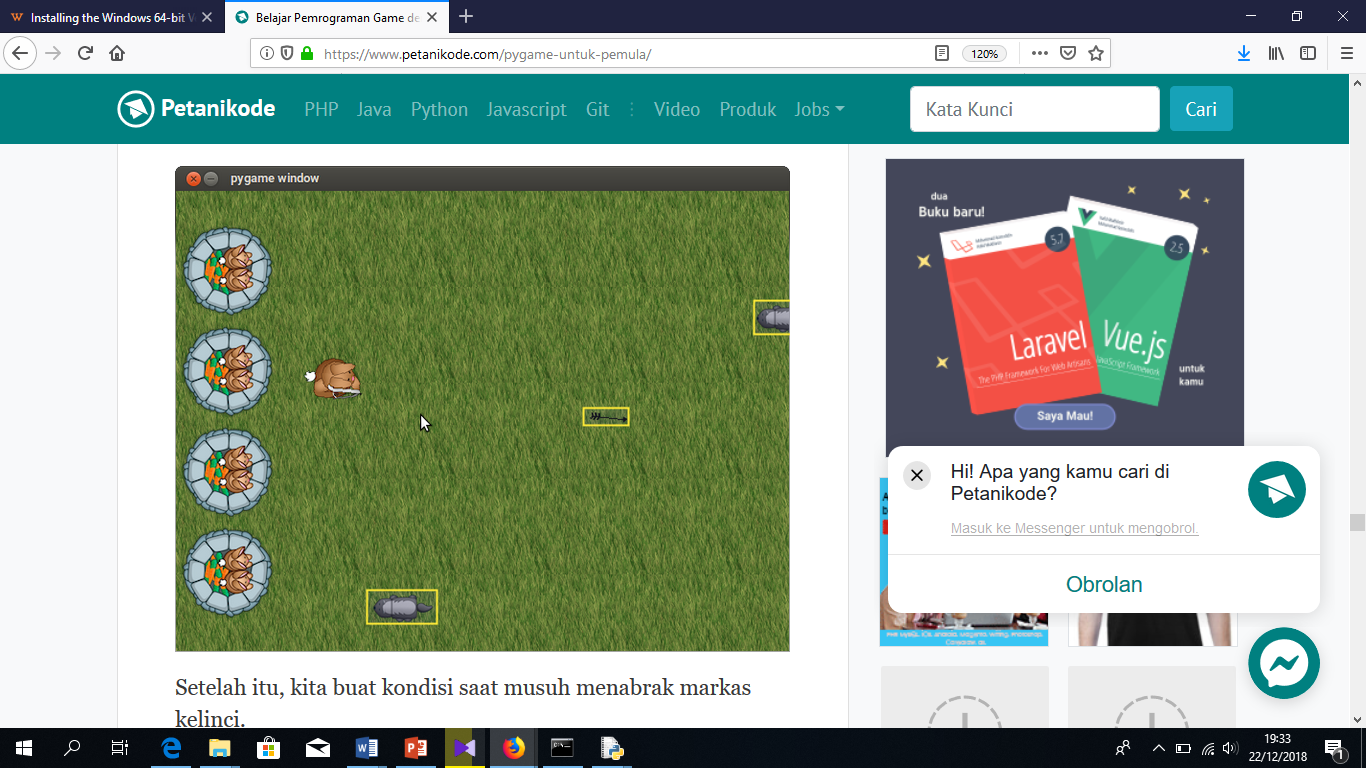
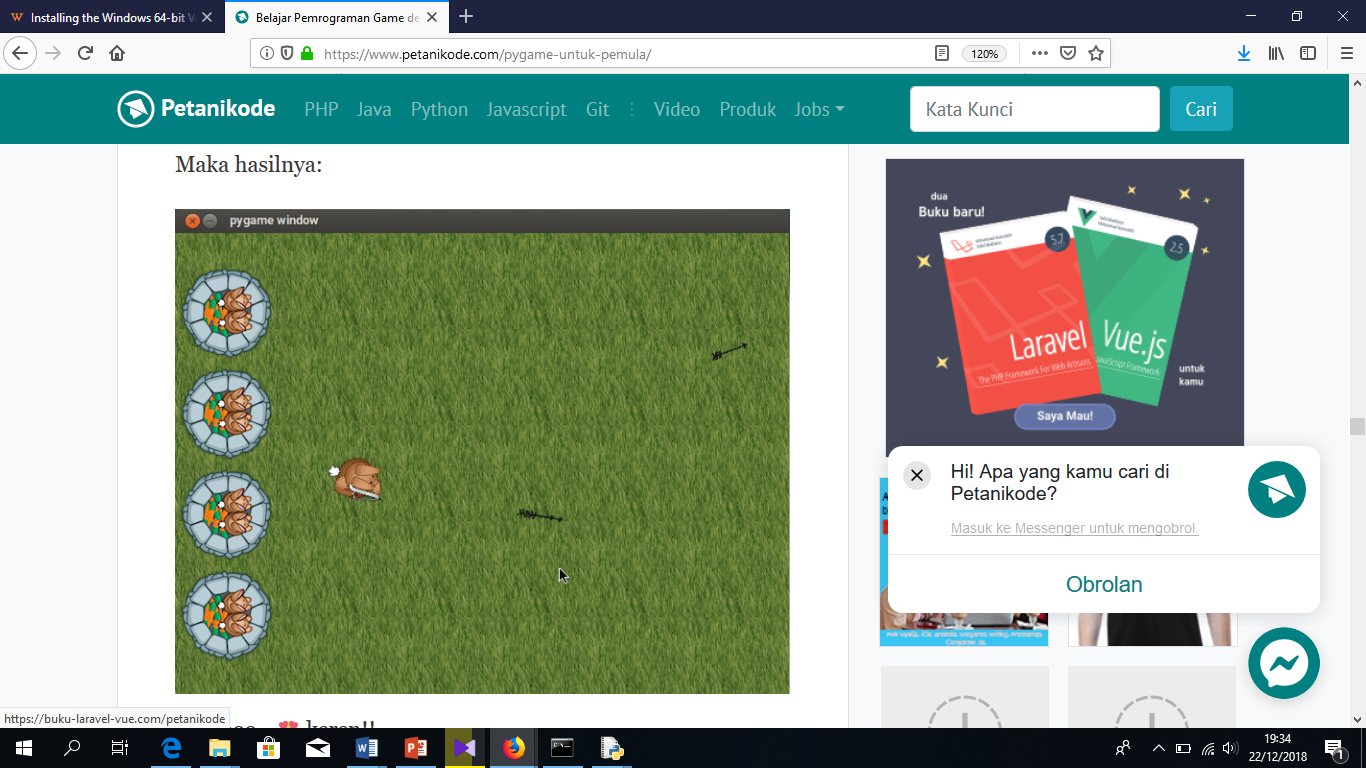


**PYGAME**

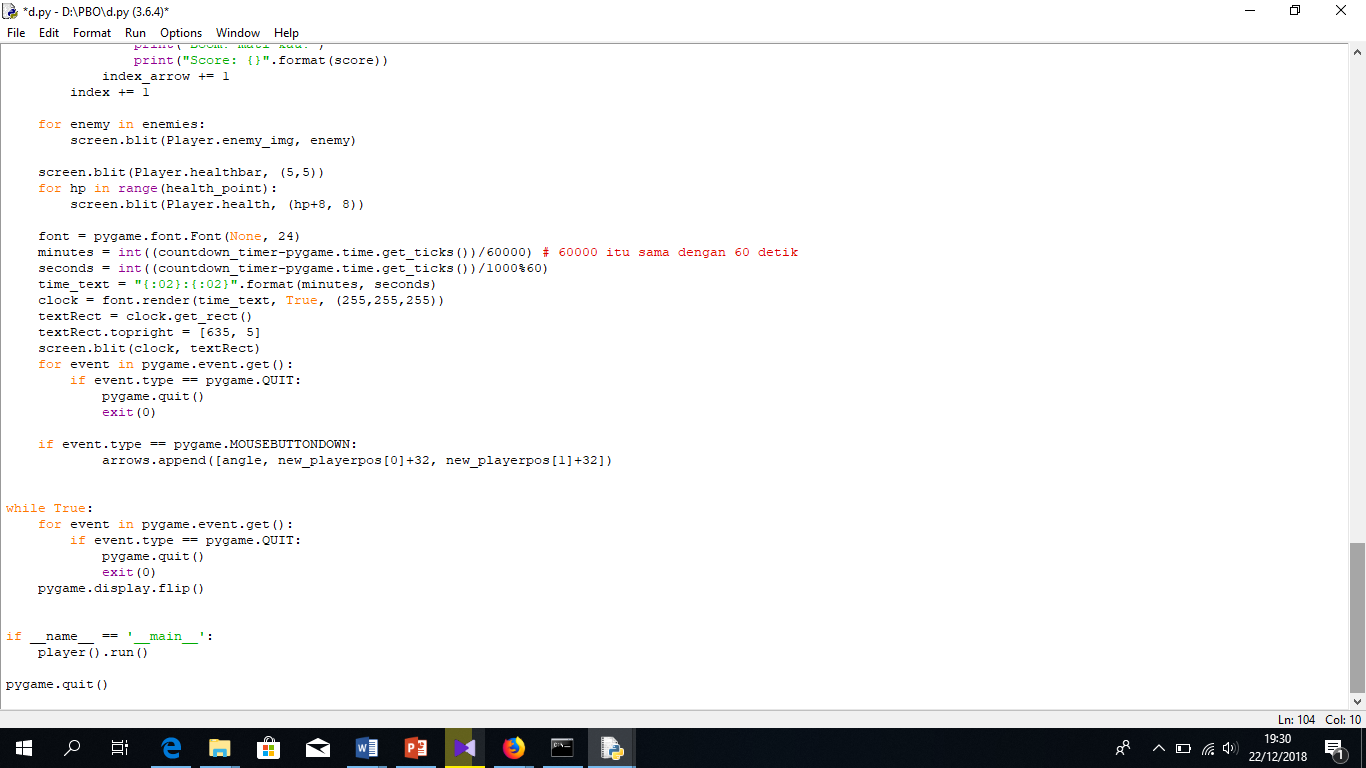
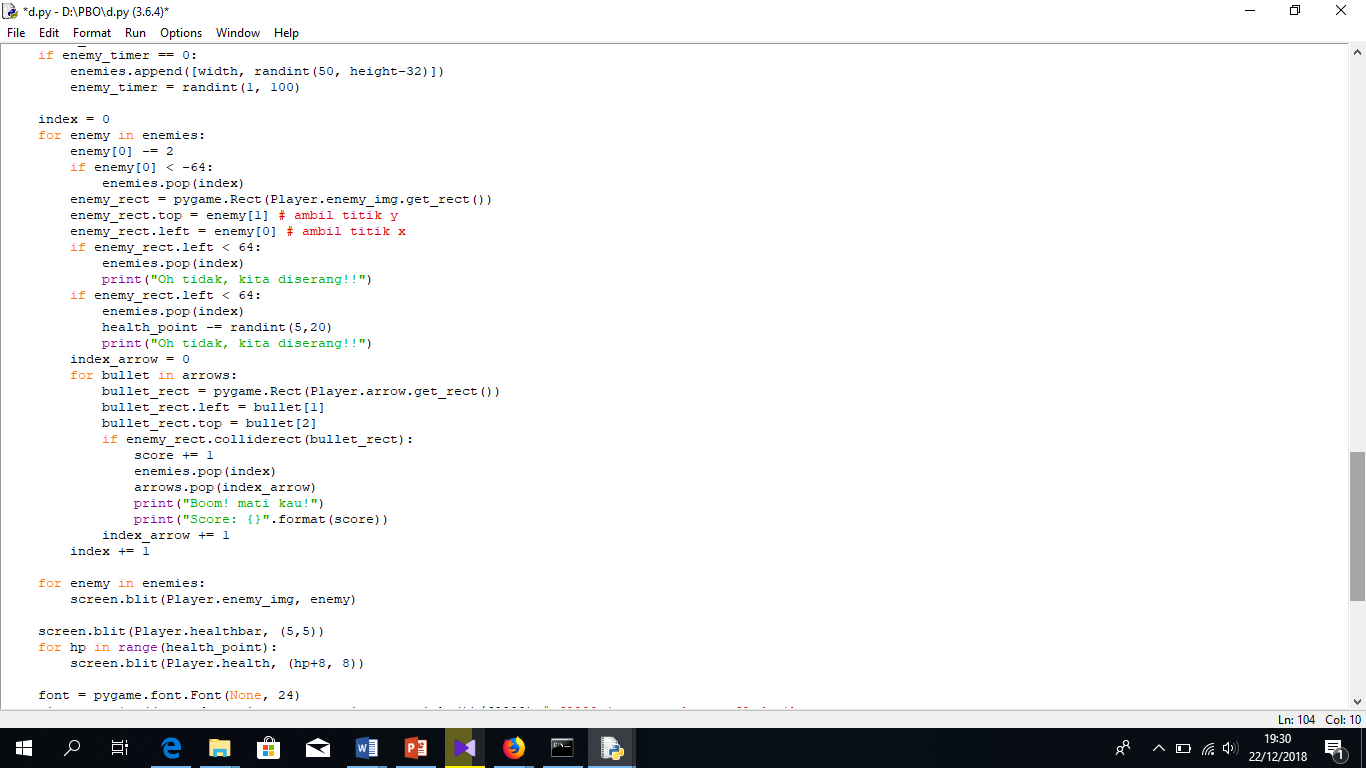
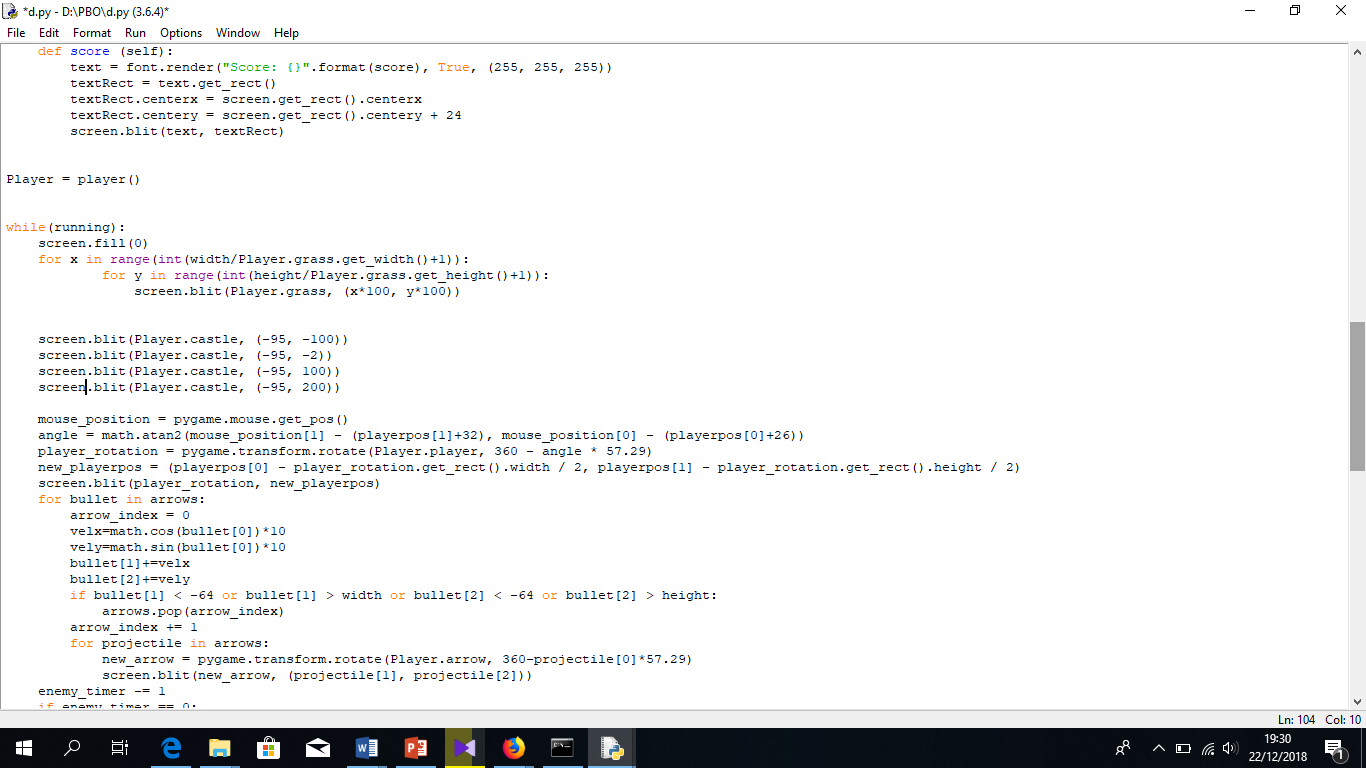
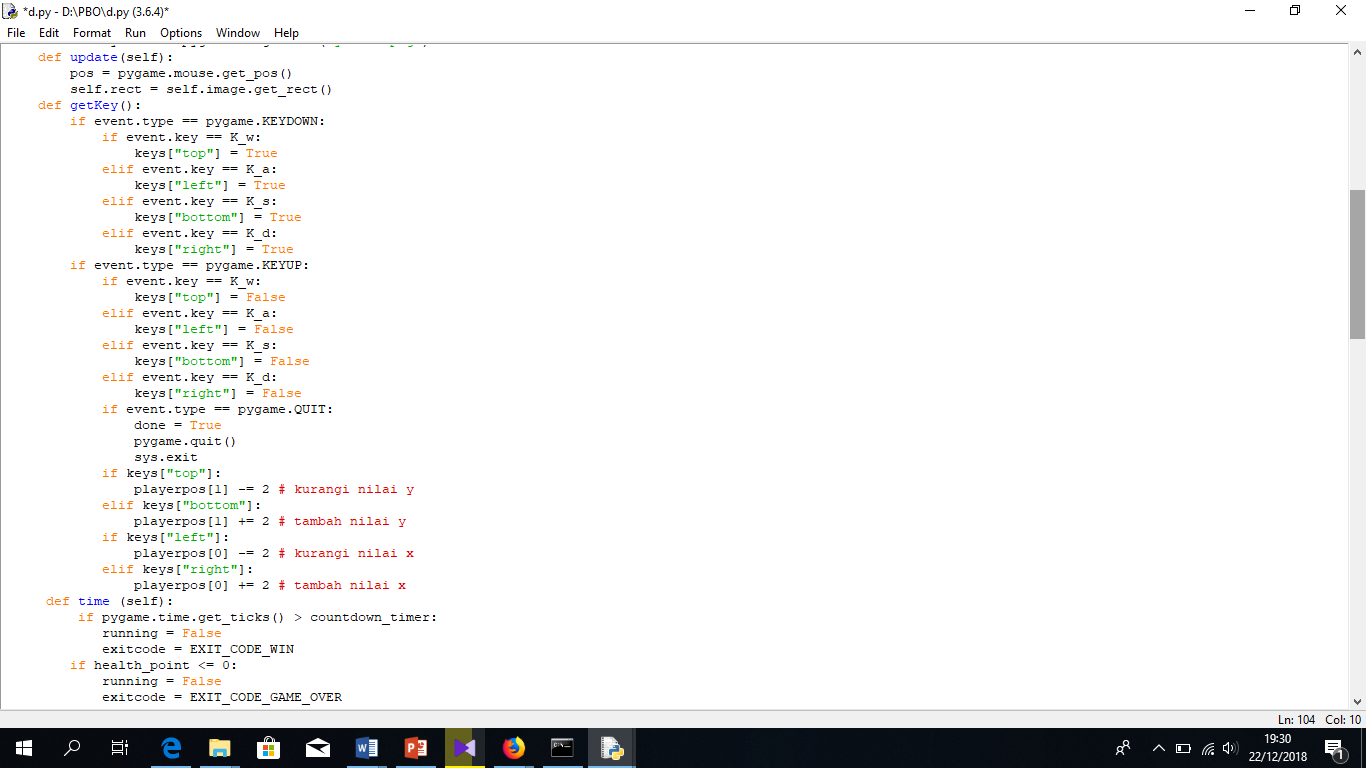
Code Awal



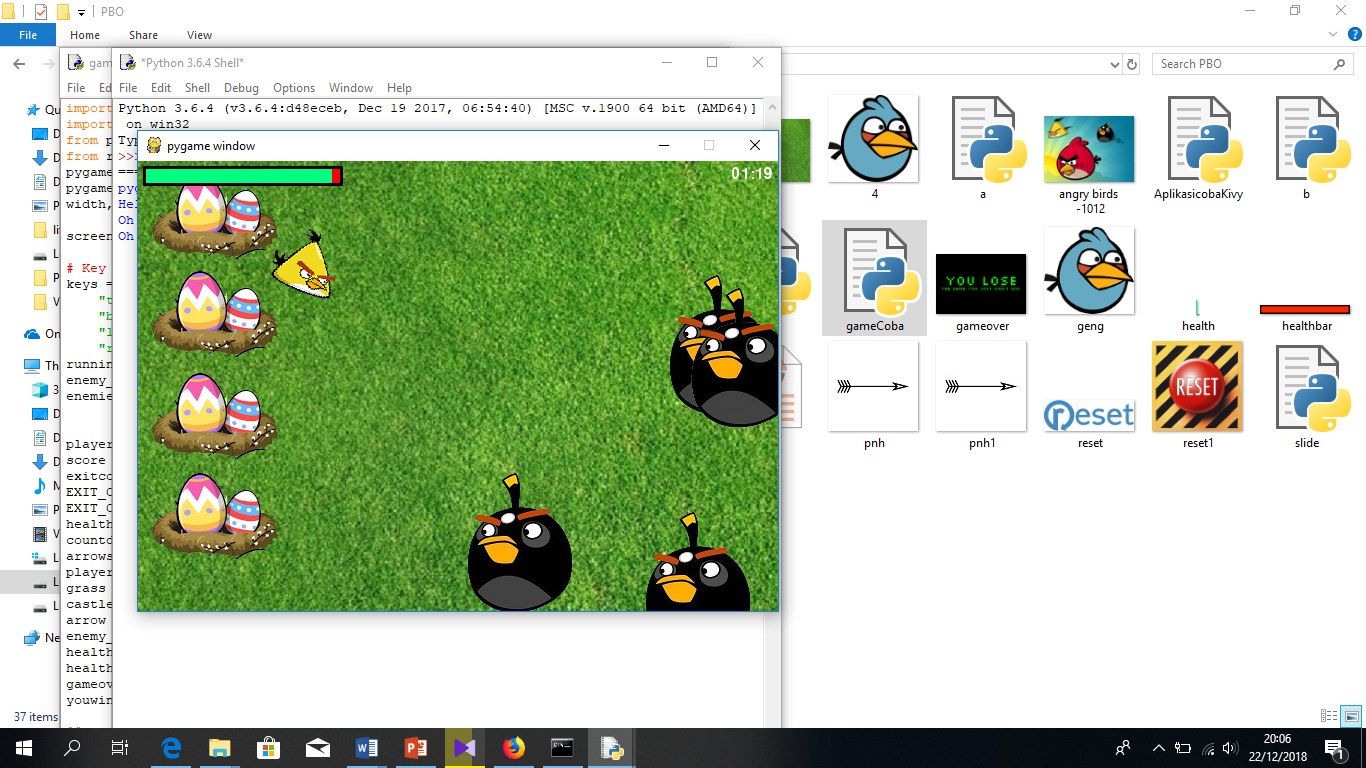
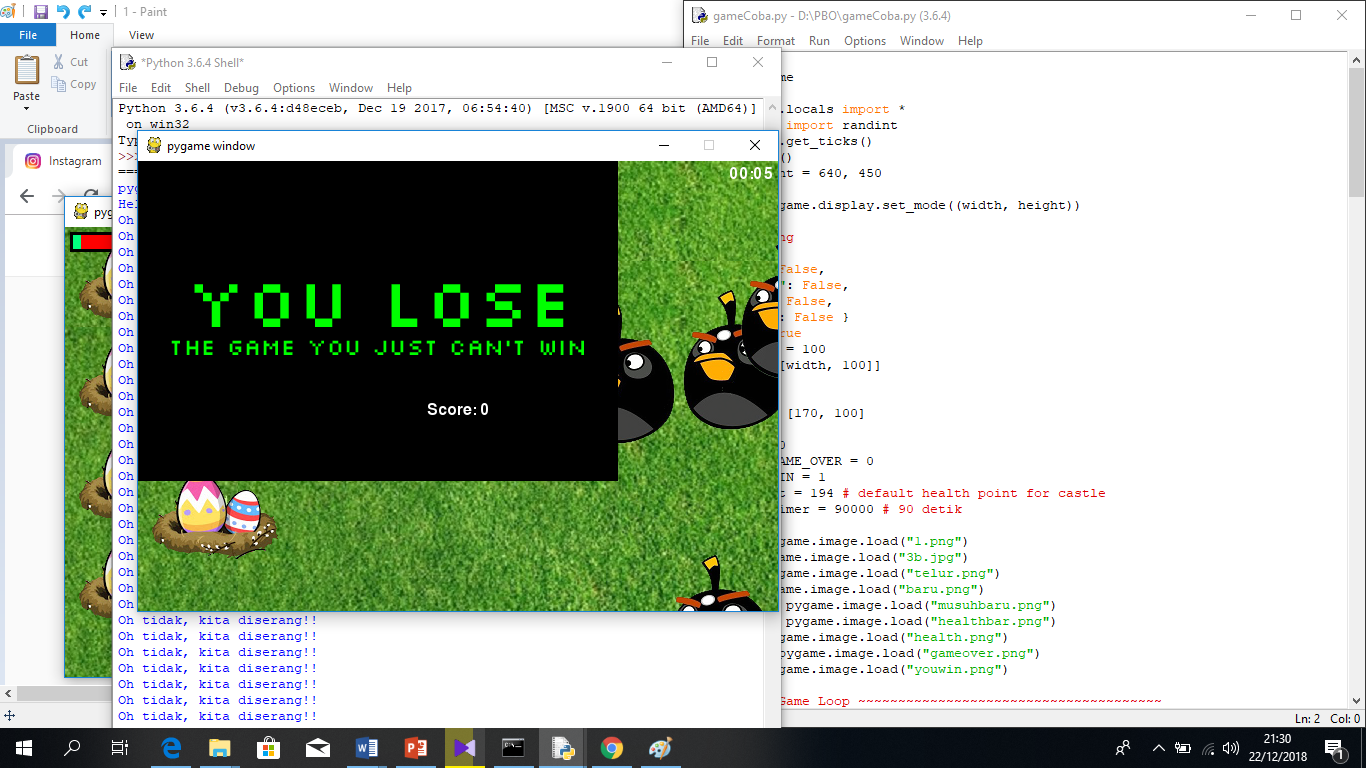
**Hasil Running :**



**CODE MODIFIKASI**



**Hasil Running :**

 ****

Penjelasan Program

|  |
| --- |
| import pygame  import math  from pygame.locals import \*  from random import randint  pygame.time.get\_ticks()  pygame.init()  width, height = 640, 450  screen = pygame.display.set\_mode((width, height))  # Key mapping  keys = {  "top": False,  "bottom": False,  "left": False,  "right": False }  running = True  enemy\_timer = 100  enemies = [[width, 100]]  class player() :  def \_\_init\_\_(self):  super().\_\_init\_\_()  self.player = pygame.image.load("1.png")  self.grass = pygame.image.load("3b.jpg")  self.castle = pygame.image.load("telur.png")  self.arrow = pygame.image.load("baru.png")  self.enemy\_img = pygame.image.load("geng.png")  self.healthbar = pygame.image.load("healthbar.png")  self.health = pygame.image.load("health.png")  self.gameover = pygame.image.load("gameover.png")  self.youwin = pygame.image.load("youwin.png")  def update(self):  pos = pygame.mouse.get\_pos()  self.rect = self.image.get\_rect()  Player = player()    #pygame.init()  playerpos = [170, 100]  exitcode = 0  EXIT\_CODE\_GAME\_OVER = 0  EXIT\_CODE\_WIN = 1  score = 0  health\_point = 194 # default health point for castle  countdown\_timer = 90000 # 90 detik  arrows = []  #The Game Loop ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  while(running):    # - Clear the screen ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  screen.fill(0)    # - Draw the game object ~~~~~~~~~~~~~~~~~~~~~~~~~~~~  # draw the grass  for x in range(int(width/Player.grass.get\_width()+1)):  for y in range(int(height/Player.grass.get\_height()+1)):  screen.blit(Player.grass, (x\*100, y\*100))    # draw the castle  screen.blit(Player.castle, (-95, -100))  screen.blit(Player.castle, (-95, -2))  screen.blit(Player.castle, (-95, 100))  screen.blit(Player.castle, (-95, 200))  mouse\_position = pygame.mouse.get\_pos()  angle = math.atan2(mouse\_position[1] - (playerpos[1]+32), mouse\_position[0] - (playerpos[0]+26))  player\_rotation = pygame.transform.rotate(Player.player, 360 - angle \* 57.29)  new\_playerpos = (playerpos[0] - player\_rotation.get\_rect().width / 2, playerpos[1] - player\_rotation.get\_rect().height / 2)  screen.blit(player\_rotation, new\_playerpos)  #screen.blit(player, playerpos)  # - Draw arrows  for bullet in arrows:  arrow\_index = 0  velx=math.cos(bullet[0])\*10  vely=math.sin(bullet[0])\*10  bullet[1]+=velx  bullet[2]+=vely  if bullet[1] < -64 or bullet[1] > width or bullet[2] < -64 or bullet[2] > height:  arrows.pop(arrow\_index)  arrow\_index += 1    # draw the arrow  for projectile in arrows:  new\_arrow = pygame.transform.rotate(Player.arrow, 360-projectile[0]\*57.29)  screen.blit(new\_arrow, (projectile[1], projectile[2]))  # - Draw Enemy  # waktu musuh akan muncul  enemy\_timer -= 1  if enemy\_timer == 0:  # buat musuh baru  enemies.append([width, randint(50, height-32)])  # reset enemy timer to random time  enemy\_timer = randint(1, 100)  index = 0  for enemy in enemies:  # musuh bergerak dengan kecepatan 5 pixel ke kiri  enemy[0] -= 2  # hapus musuh saat mencapai batas layar sebelah kiri  if enemy[0] < -64:  enemies.pop(index)  # collision between enemies and castle  enemy\_rect = pygame.Rect(Player.enemy\_img.get\_rect())  enemy\_rect.top = enemy[1] # ambil titik y  enemy\_rect.left = enemy[0] # ambil titik x  # benturan musuh dengan markas kelinci  if enemy\_rect.left < 64:  enemies.pop(index)  print("Oh tidak, kita diserang!!")  if enemy\_rect.left < 64:  enemies.pop(index)  health\_point -= randint(5,20)  print("Oh tidak, kita diserang!!")    # Check for collisions between enemies and arrows  index\_arrow = 0  for bullet in arrows:  bullet\_rect = pygame.Rect(Player.arrow.get\_rect())  bullet\_rect.left = bullet[1]  bullet\_rect.top = bullet[2]  # benturan anak panah dengan musuh  if enemy\_rect.colliderect(bullet\_rect):  score += 1  enemies.pop(index)  arrows.pop(index\_arrow)  print("Boom! mati kau!")  print("Score: {}".format(score))  index\_arrow += 1  index += 1  # gambar musuh ke layar  for enemy in enemies:  screen.blit(Player.enemy\_img, enemy)  # 6.3 - Draw Health bar  screen.blit(Player.healthbar, (5,5))  for hp in range(health\_point):  screen.blit(Player.health, (hp+8, 8))  # 6.4 - Draw clock  font = pygame.font.Font(None, 24)  minutes = int((countdown\_timer-pygame.time.get\_ticks())/60000) # 60000 itu sama dengan 60 detik  seconds = int((countdown\_timer-pygame.time.get\_ticks())/1000%60)  time\_text = "{:02}:{:02}".format(minutes, seconds)  clock = font.render(time\_text, True, (255,255,255))  textRect = clock.get\_rect()  textRect.topright = [635, 5]  screen.blit(clock, textRect)    # 7 - Update the sceeen ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~    # 8 - Event Loop ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  for event in pygame.event.get():  # event saat tombol exit diklik  if event.type == pygame.QUIT:  pygame.quit()  exit(0)  # Fire!!  if event.type == pygame.MOUSEBUTTONDOWN:  arrows.append([angle, new\_playerpos[0]+32, new\_playerpos[1]+32])    # chek the keydown and keyup  def getKey():  if event.type == pygame.KEYDOWN:  if event.key == K\_w:  keys["top"] = True  elif event.key == K\_a:  keys["left"] = True  elif event.key == K\_s:  keys["bottom"] = True  elif event.key == K\_d:  keys["right"] = True  if event.type == pygame.KEYUP:  if event.key == K\_w:  keys["top"] = False  elif event.key == K\_a:  keys["left"] = False  elif event.key == K\_s:  keys["bottom"] = False  elif event.key == K\_d:  keys["right"] = False  if event.type == pygame.QUIT:  done = True  pygame.quit()  sys.exit  # - End of event loop ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  # 9. Move the player ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  if keys["top"]:  playerpos[1] -= 2 # kurangi nilai y  elif keys["bottom"]:  playerpos[1] += 2 # tambah nilai y  if keys["left"]:  playerpos[0] -= 2 # kurangi nilai x  elif keys["right"]:  playerpos[0] += 2 # tambah nilai x  # 10 - Win/Lose check ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  if pygame.time.get\_ticks() > countdown\_timer:  running = False  exitcode = EXIT\_CODE\_WIN  if health\_point <= 0:  running = False  exitcode = EXIT\_CODE\_GAME\_OVER  # 11 - Win/lose display ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  if exitcode == EXIT\_CODE\_GAME\_OVER:  screen.blit(Player.gameover, (0, 0))  else:  screen.blit(Player.youwin, (0, 0))  # Tampilkan score  text = font.render("Score: {}".format(score), True, (255, 255, 255))  textRect = text.get\_rect()  textRect.centerx = screen.get\_rect().centerx  textRect.centery = screen.get\_rect().centery + 24  screen.blit(text, textRect)  while True:  for event in pygame.event.get():  if event.type == pygame.QUIT:  pygame.quit()  exit(0)  pygame.display.flip()  pygame.quit() |

**Class player :**

**Method :**

* def \_\_init\_\_:

isinya gambar dari karakter dan background dan juga \_\_init\_\_() ini juga merupakan contructornya.

* def update:

*pos* untuk mengupdate posisi mouse dan *self.rect* untuk karakternya

* def getKey:

didalam keydown dan keyup dilakukan pengecekan tombol yang ditekan dengsn konstanta yg sudah disediakan pygame. Dan juga ketika tombol *w* ditekan maka diaakan berpindah sejauh 2y, ketika tombol *s* di tekan maka object akan berpindah sejauh 2y ke arah bawah, ketika tombol *a* ditekan maka object akan berpindah sejauh 2x kearah kiri, dan ketika tombol *d* ditekan maka object akan berpindah sejauh 2x kearah kanan.

* def time

time akan menampilkan waktu sebanayk 90 detik dengan format *01:30*

* def score

score akan ditampilakan di comment line python ketika object musuh tertembak maka akan bertambah nilai 1 dan seterusnya ketika object musuh melewati kita maka nilainya tidak akan ditambah.

* looping :

didalam loopingn ini terdapat kondisi contohnya *for bullet in arrows* (untuk perulangan panah agar terus muncul ketika mouse ditekan) ketika kita diserang maka di comment line akan muncul  *oh tidak, kita diserang!,* dan ketika kita bisa menmbak musuh maka kita akan mendapatkan score +1 dan akan muncul di comment line

**Class musuh :**

**Method**

* move(self): Moving the birds involves 3 steps : save the current head position, since it will be used to add a block to the tail, move the head one cell in the current direction, and add the new tail block to the tail
* remove(self): With our current birds, removing the whole thing sums up to remove its head and tail, so we just have to call the corresponding methods. How they deal with it is their problem, not the Snake's. It just passes down the command.
* set\_position(self, position): We consider the birds position as the position occupied by the head.
* get\_full\_position: But sometimes we'll want to know the whole set of cells occupied by the bird.

**class playered**

**Method**

* remove(self):

# reset the size if some fruits were eaten

# remove every block of the tail from the canvas

# this is why we don't need a is\_on\_board() here :

# if a block is not on board, it's not on the list

# thus we can't try to delete an object not already

* add\_block: 3 things happen here : the new block position passed as argument is appended to the object's list. The list's number of elements is adapted if need be by poping the oldest block. The blocks are drawn on the canvas, and the same process as before, happens so that our list of block objects keeps a constant size.

**INHERITANCE :**

class musuh(Widget):

# children widgets containers

head = ObjectProperty(None)

tail = ObjectProperty(None)

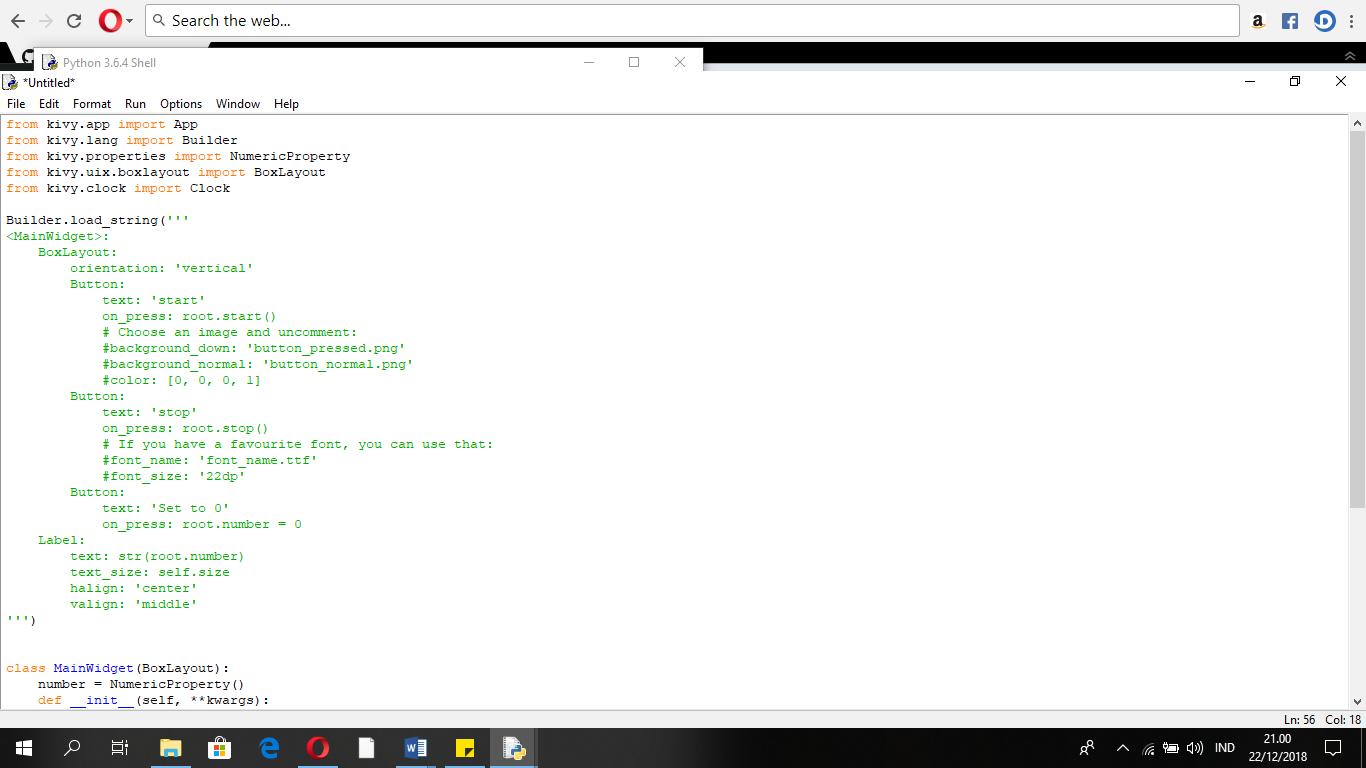
**POLYMORPHISM:** Terdapat nama method sama dalam dua class yang berbeda. Missal method remove pada class player dengan method remove pada class playered

**OVERLOADING:-**

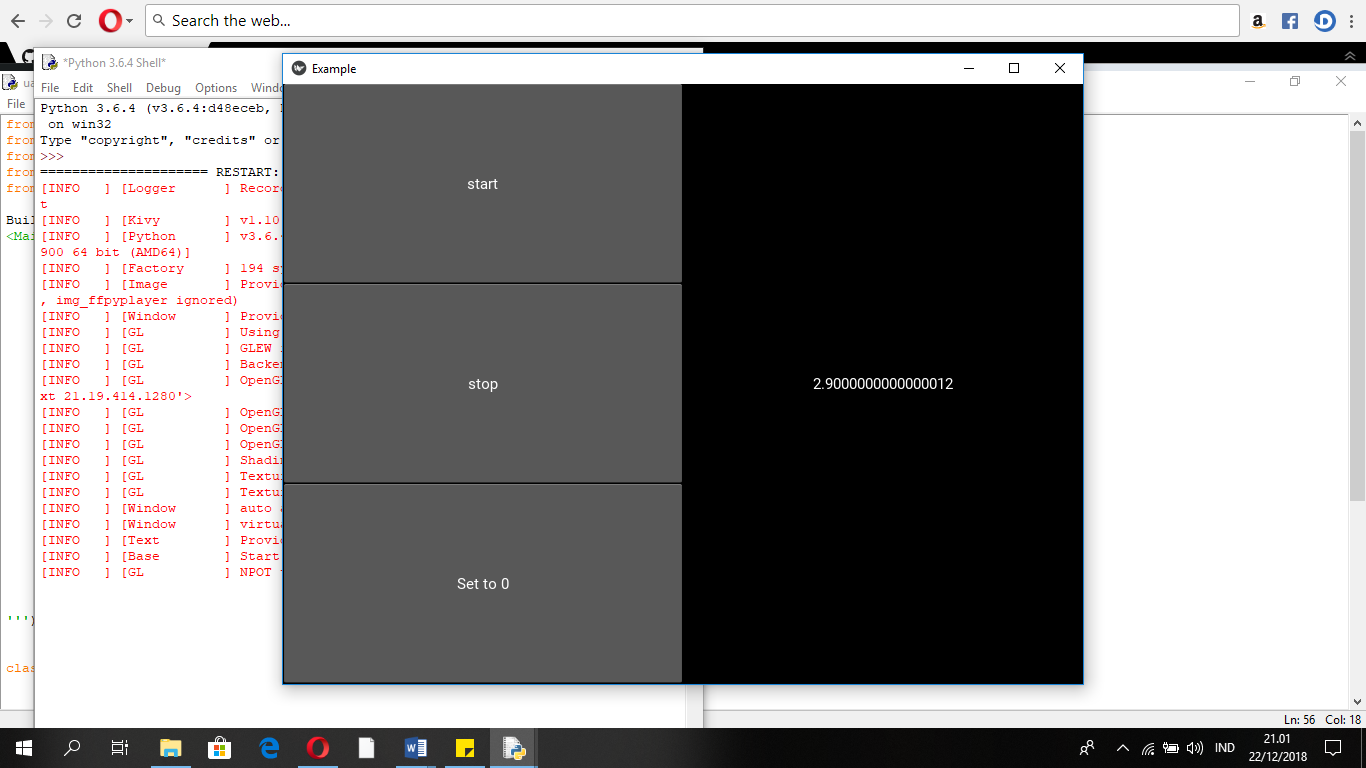
**ENCAPSULATION: -**

**KIVY**

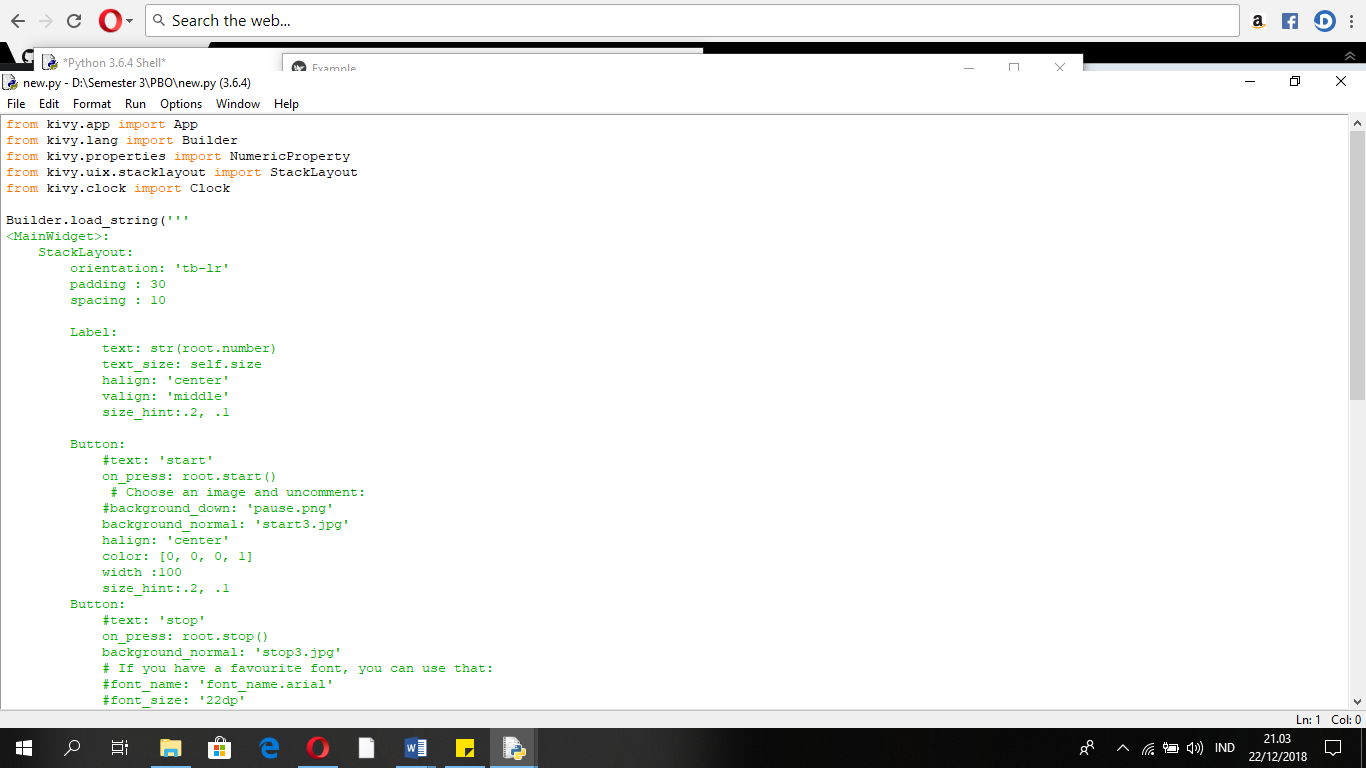
**Code Awal**



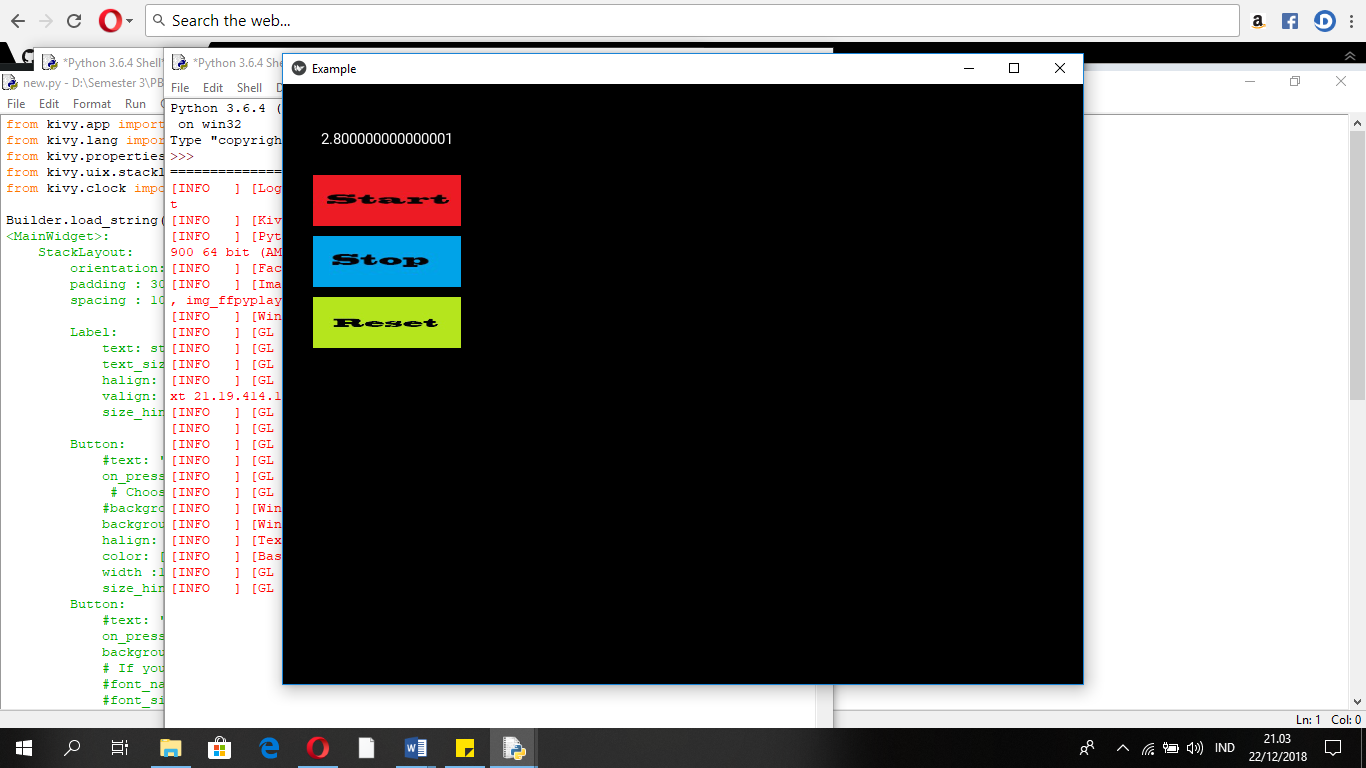
**Hasil Running :**



**Code Modifikasi**



**Hasil Running :**



**Penjelasan Program**

Class MainWidget(StackLayout)

* Increment\_time
* Start
* Stop
* Update\_sw

Class ExampleApp(App)

* build

Kelas MainWidget : kelas ini berfungsi untuk mengatur widget di grid. Anda harus menentukan setidaknya satu dimensi dari grid sehingga kivy dapat menghitung ukuran elemen dan bagaimana untuk mengatur mereka.

def increment\_time : fungsi yang akan menambahkan operasi yang diberikan.

def start : Berfungsi untuk memulai pada stopwatch.

def stop : Berfungsi untuk memberhetikan stopwatch yang masih melakukan operasi.

def update\_sw : Berfungsi untuk mengupdate atau me reset stopwatch.

Kelas ExampleApp : kelas ini berfungsi untuk menjalankan aplikasi. Di dalam kelas ini hanya terdapat 1 method, yaitu method build untuk menjalankan aplikasi ExampleApp denganmemanggil class MainWidget**.**

DAFTAR PUSTAKA

https://gist.github.com/Laspimon/740185c161c386ddaf2a

https://www.petanikode.com/pygame-untuk-pemula/