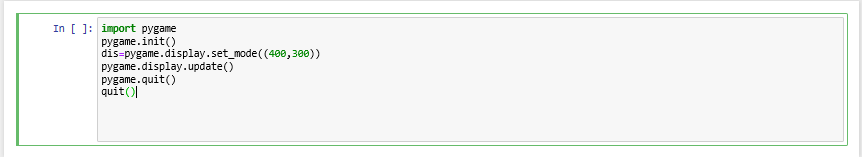
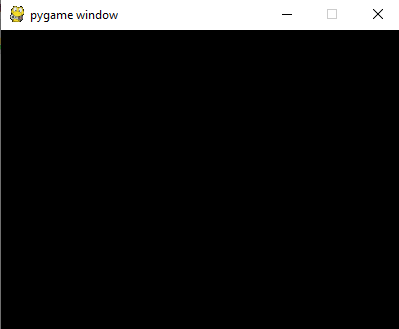
**Cara pembuatan game :**

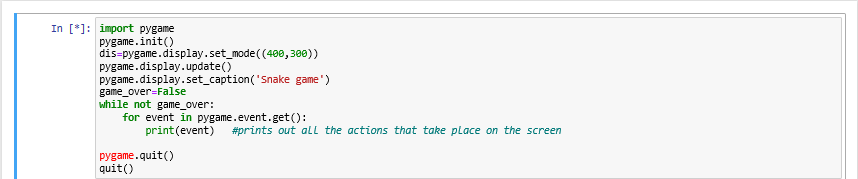
**Buat Layar:**

Untuk membuat layar menggunakan Pygame, menggunakan fungsi display.set\_mode(). Selain itu, juga menggunakan metode init() dan quit() untuk menginisialisasi dan menghapus semua inisialisasi di awal dan akhir kode. Metode update() digunakan untuk memperbarui setiap perubahan yang dibuat pada layar. Ada metode lain yaitu flip() yang bekerja mirip dengan fungsi update(). Perbedaannya adalah metode update() hanya memperbarui perubahan yang dibuat (namun, jika tidak ada parameter yang diteruskan, memperbarui layar lengkap) tetapi metode flip() mengulang layar lengkap lagi.

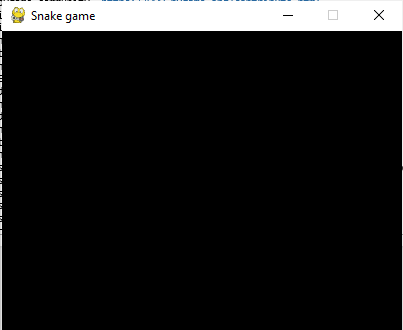




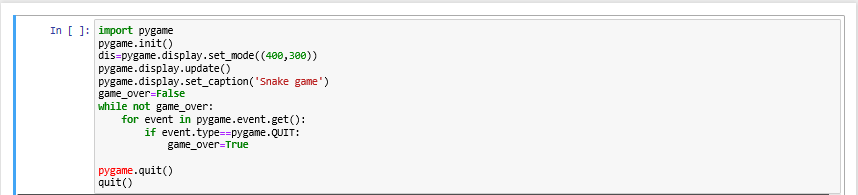
Tetapi ketika Anda menjalankan kode ini, layar akan muncul, tetapi akan segera menutup juga. Untuk memperbaikinya, Anda harus menggunakan game loop menggunakan while loop sebelum saya benar-benar keluar dari game sebagai berikut:



Saat menjalankan kode ini, kita akan melihat bahwa layar yang dilihat sebelumnya tidak berhenti dan juga mengembalikan semua tindakan yang terjadi di atasnya, dengan melakukannya menggunakan fungsi event.get(). Kami juga telah menamai layarnya sebagai “Snake Game” menggunakan fungsi display.set\_caption().



Sekarang, kita memiliki layar untuk memainkan Game Ular, tetapi ketika mencoba mengklik tombol tutup, layar tidak menutup. Ini karena belum menentukan bahwa layar harus keluar saat kita menekan tombol tutup itu. Untuk melakukan itu, Pygame menyediakan event yang disebut "QUIT" dan itu harus digunakan sebagai berikut:

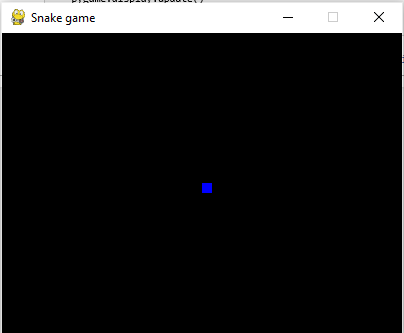


Jadi sekarang layar sudah siap dan langkah selanjutnya adalah menggambar ular kita di layar yang tercakup dalam topik dibawah ini.

**Buat Ular:**

Untuk membuat ular, pertama-tama dengan menginisialisasi beberapa variabel warna untuk mewarnai ular, makanan, layar, dll. Skema warna yang digunakan di Pygame adalah RGB yaitu "Merah Hijau Biru". Jika mengatur semua ini ke 0, warnanya akan menjadi hitam dan 255 akan menjadi putih. Jadi bentuk ular dibuat menjadi persegi panjang. Untuk menggambar persegi panjang di Pygame, dapat menggunakan fungsi yang disebut draw.rect() yang akan membantu menggambar persegi panjang dengan warna dan ukuran yang diinginkan.





Langkah selanjutnya adalah membuat ular bergerak.

Dalam pembuatan game ini kami juga memasuki sumber suara dalam permainan yang dimasukkan Ketika pada saat memulai permainan, pada saat ular melewati makannya dan Ketika permainan kalah dengan memasukkan perintah seperti dibawah ini:

crash\_sound = pygame.mixer.Sound("crash.mp3")

pygame.mixer.music.load("jazz.mp3")

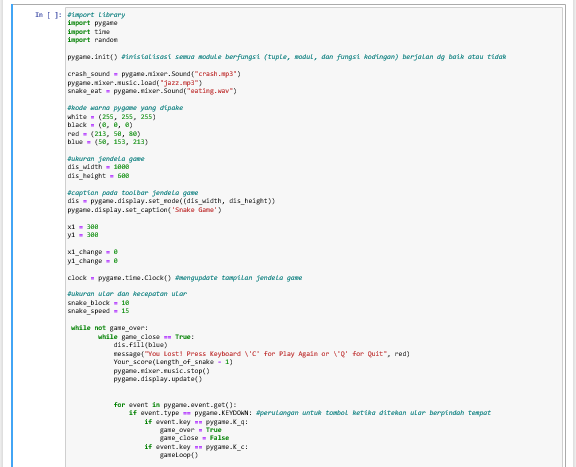
snake\_eat = pygame.mixer.Sound("eating.wav")

**Perpindahan gerak ular:**

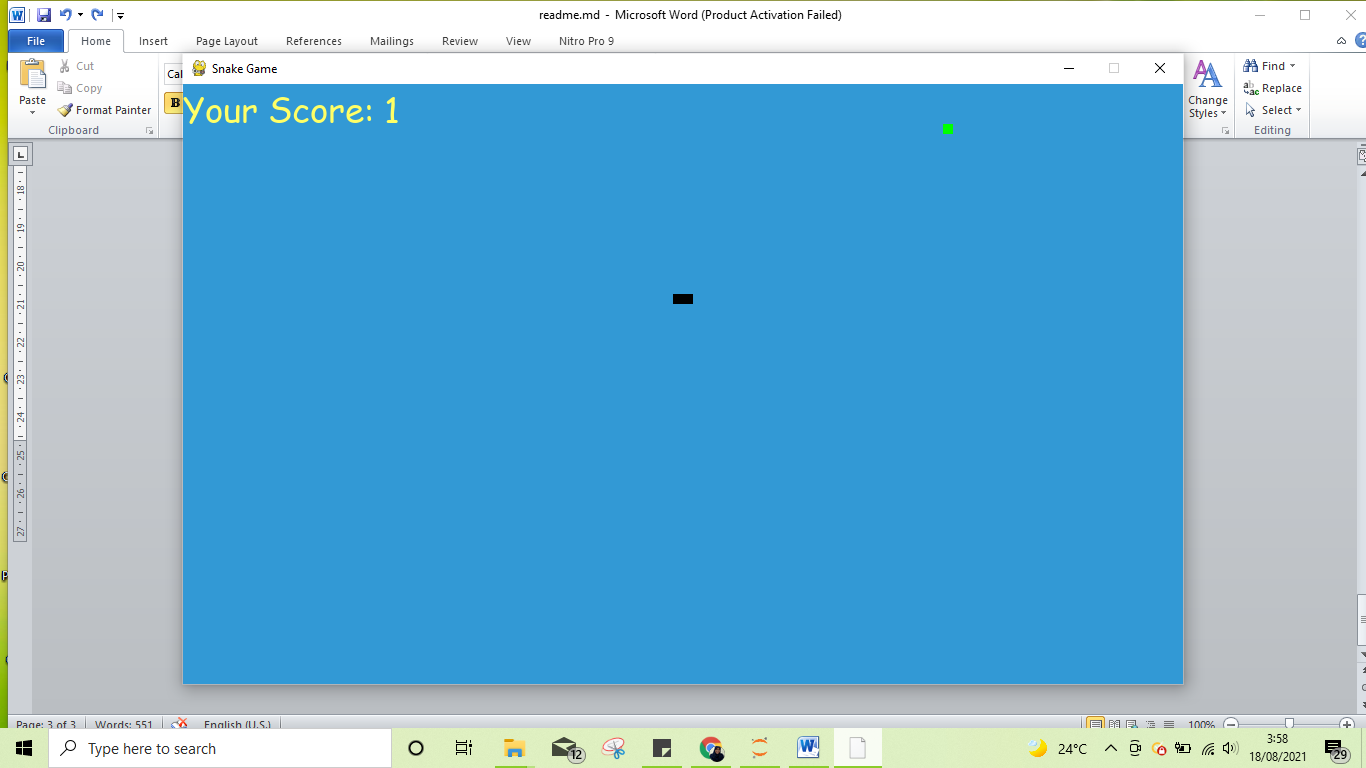
Permindahan gerakan ularnya mengikuti pergerakan dijendela permainan dengan menggunakan perintah yang ada di kelas KEYDOWN di Pygame. Event yang digunakan di sini adalah, K\_UP, K\_DOWN, K\_LEFT, dan K\_RIGHT untuk membuat ular bergerak ke atas, bawah, kiri dan kanan. Kemudian juga kita masukkan perintah fungsi suara pada saat permainan ini dimulai dengan perintah

pygame.mixer.music.load("jazz.mp3")

pygame.mixer.music.play(-1)





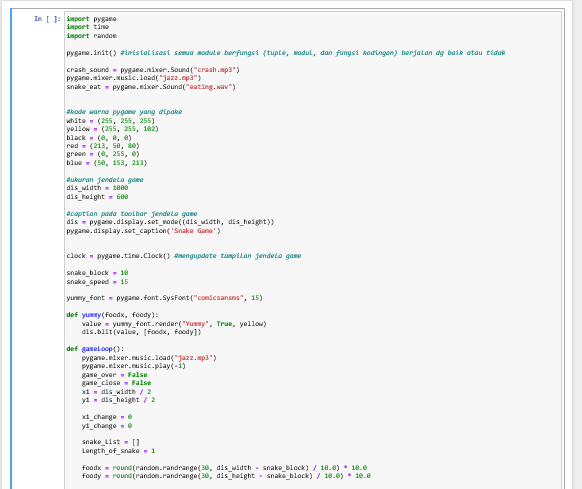


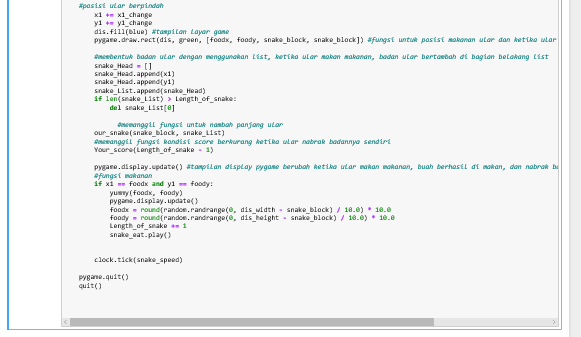
**Menambahkan Makanan:**

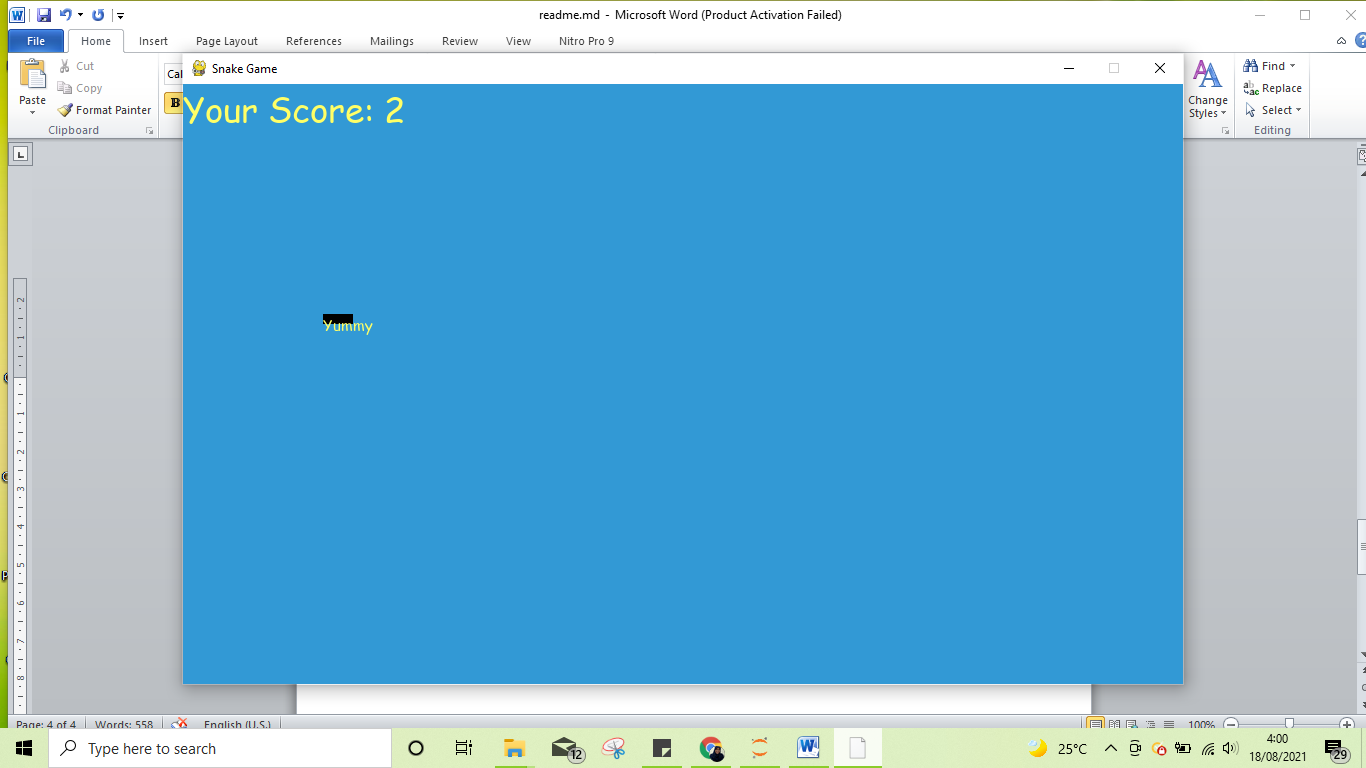
Di sini, kita akan menambahkan beberapa makanan untuk ular dan ketika ular melewati makanan itu, akan mendapat pesan yang mengatakan "yummy!!" Panjang ular pada dasarnya terkandung dalam daftar dan ukuran awal yang ditentukan dalam kode berikut adalah satu blok. Kemudian setelah ular berhasil melewati makanannya pertambahan Panjang ular akan bertambah 1 dan juga akan menampilkan dibagian skornya.

Kemudian juga kita masukkan perintah fungsi suara pada saat ularnya ini melewati makanannya dengan perintah

snake\_eat.play()

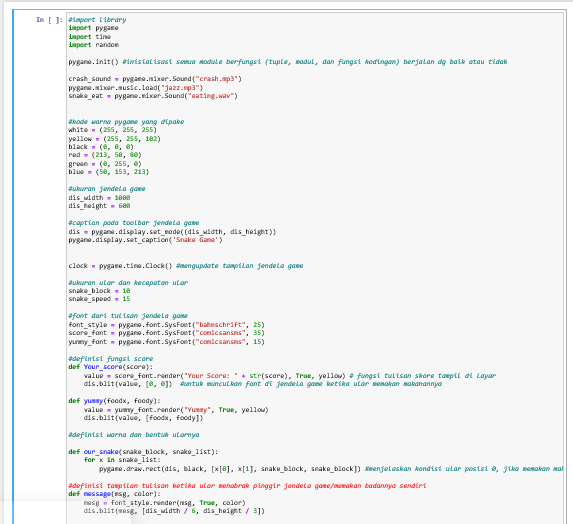


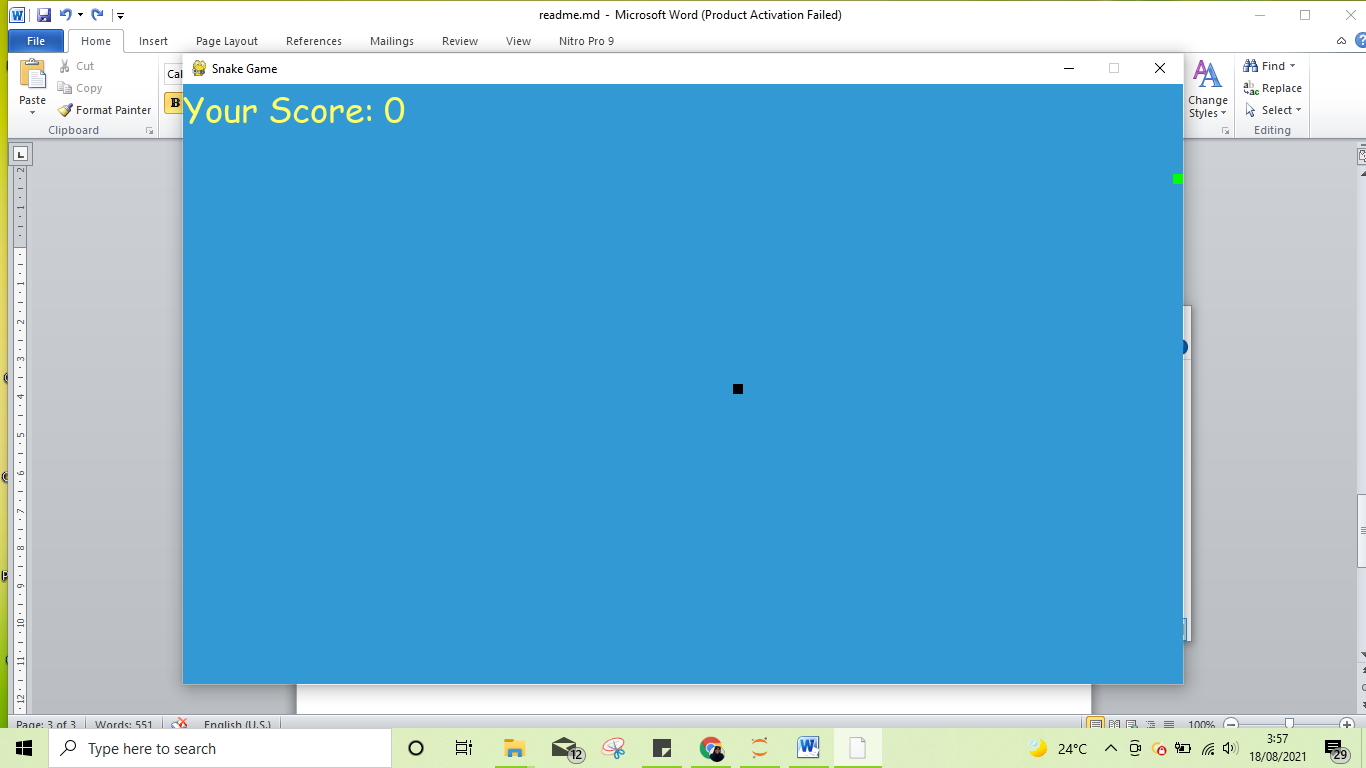




**Menampilkan Skor:**

Dalam menampilkan skor pemain, terlebih dahulu dengan membuat fungsi baru sebagai "Your\_score". Fungsi ini akan menampilkan panjang ular dikurangi 1 karena itulah ukuran awal ular.





**Permainan kalah :**

Dalam permainan ular ini, jika ular mencapai batas layar, maka permainan akan kalah. Untuk menentukan ini, menggunakan pernyataan 'jika' yang mendefinisikan batas koordinat x dan y ular menjadi kurang dari atau sama dengan layer.

Ketika permainan sudah kalah dan ingin melanjutkan permainannya kembali, dengan menggunakan perintah tekan tombol keyboard C untuk memulai kembali dan jika ingin keluar dari permainan ini dengan menekan tombol keyboard Q untuk keluar

Kemudian juga kita masukkan perintah fungsi suara pada saat permainan ini kalah dengan perintah:

pygame.mixer.Sound.play(crash\_sound)

dan juga perintah ketika permainan ini keluar atau terhenti, suara dalam permainan ini juga akan berhenti, sesuai dengan perintah fungsi dibawah ini:

pygame.mixer.music.stop()

pygame.display.update()

