

TRANSFORMASI DIGITAL
CREATE A SIMPLE GAME WITH PYTHON



Mukhtada Billah Nasution

F1E122037

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS JAMBI

2024

1. Pendahuluan

Dalam era perkembangan teknologi, kemampuan pemrograman sangat penting, terutama dalam menciptakan aplikasi interaktif seperti permainan. Salah satu bahasa pemrograman yang populer dan mudah dipelajari untuk pemula adalah Python. *Python* tidak hanya memiliki sintaks yang sederhana, tetapi juga menyediakan berbagai pustaka dan alat yang mendukung pengembangan aplikasi, termasuk permainan sederhana.

Laporan ini berfokus pada pembuatan permainan sederhana menggunakan Python IDLE. Permainan ini dirancang untuk menggunakan metode bisection atau pembagian rentang secara berulang untuk menemukan angka yang dipilih oleh pemain dalam rentang 0 hingga 1024. Metode bisection merupakan salah satu algoritma pencarian efisien yang sering digunakan dalam pemrograman karena prinsip dasarnya yang sederhana dan logis.

2. Hasil dan Pembahasan

2.1. Tujuan

Pada lab ini penulis akan membuat game sederhana dengan Python IDLE.

2.2. Sumber daya yang dibutuhkan

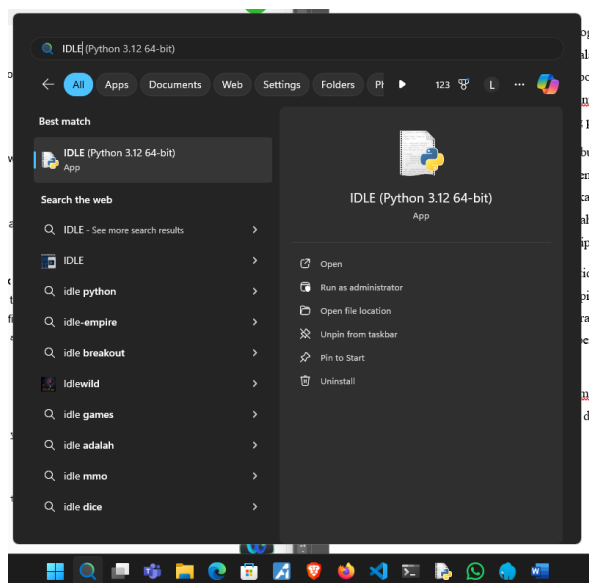
Komputer pribadi yang memadai dengan RAM yang cukup dan Akses Internet.

2.3. Petunjuk

Tujuan dari game adalah untuk mencari nomor yang dipilih oleh pengguna di antara 0 dan 1024 menggunakan metode bisection.

2.4. Tahapan Dalam Membuat Game Sederhana di Python

- a. Membuka **IDLE** dari Python, dengan cara mencari IDLE di pencarian lalu membukanya.



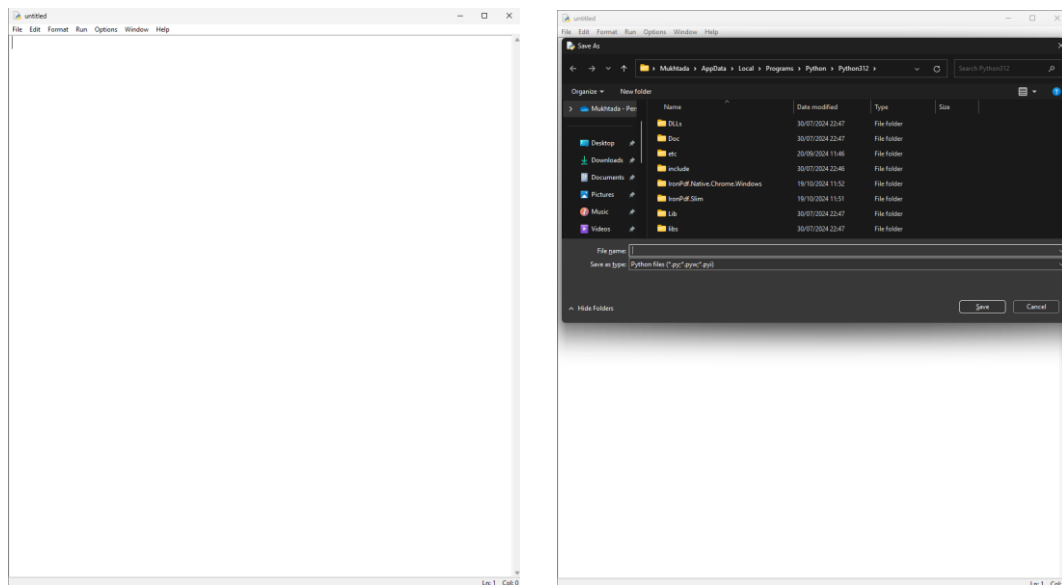
Tampilan pencarian

- b. Pada tampilan di bawah, kemudian tekan CTRL N untuk membuka editor baru.



Tampilan IDLE

- c. Kemudian pada tampilan di bawah, ada baiknya disimpan terlebih dahulu ke suatu folder agar progres tidak hilang. Bisa disimpan pada directory yang diinginkan, seperti di C:/Project/Simple Game



Menyimpan file python

- d. Pada editor menyalin dan menempel kode berikut:

```
y=input('Enter an integer between 0 and 1024 -- ')
x=int(y)
a=0
b=1024
test=True
if x == 0:
    print('Your number is 0 , thank you for playing.')
    test=False
```

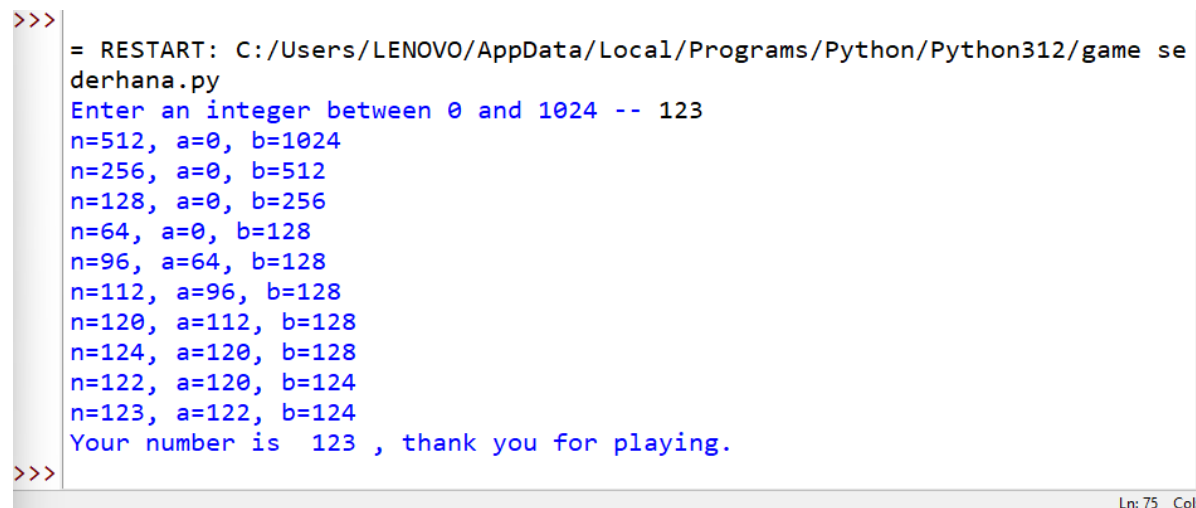
```

else:
    if x == 1024:
        print('Your number is 1024, thank you for playing.')
        test=False

    while test == True:
        n=int((a+b)/2)
        if n == x:
            print('Your number is ', n, ', thank you for playing.')
            break
        else:
            if n < x:
                a=n
            else:
                b=n

```

- e. Kemudian simpan lagi dengan menekan CTRL S untuk menyimpan progres.
- f. Pada hasil eksekusi dapat dilihat bahwa program akan menampilkan kembali nomor yang diberikan.



```

>>> = RESTART: C:/Users/LENOVO/AppData/Local/Programs/Python/Python312/game se
      derhana.py
      Enter an integer between 0 and 1024 -- 123
      n=512, a=0, b=1024
      n=256, a=0, b=512
      n=128, a=0, b=256
      n=64, a=0, b=128
      n=96, a=64, b=128
      n=112, a=96, b=128
      n=120, a=112, b=128
      n=124, a=120, b=128
      n=122, a=120, b=124
      n=123, a=122, b=124
      Your number is 123 , thank you for playing.
>>>

```

Hasil eksekusi kode program

2.5. Refleksi

- 1) Bagaimana kamu menangkap pemain yang memasukkan angka antara 0 dan 1024?
 Logika dari program ini adalah, di dalam program terdapat variabel a dan b yang bernilai 0 dan 1024. Kemudian variabel x akan menerima input yang dimasukkan user.
 Pada program terdapat percabangan if-else, di mana jika nilai yang diinput 0 dan 1024, program akan langsung menampilkan nilainya, jika tidak maka akan masuk ke perulangan while.
 Di dalam perulangan while, terdapat operasi penjumlahan antara a dan b, yang hasilnya di bagi dua, terdapat juga percabangan if-else, di mana jika hasil operasi kecil dari nilai input maka nilai variabel a akan menjadi nilai hasil operasi, sebaliknya jika lebih besar maka nilai variabel b akan menjadi nilai hasil operasi.

- 2) Bagaimana kamu menangkap pemain yang memasukkan angka berkoma? Program akan mengalami error. Karena program hanya mengkonversi masukan ke bilangan bulat, bukan berkoma.

3. Penutup

Pembuatan permainan sederhana menggunakan Python IDLE ini telah memberikan gambaran bagaimana algoritma bisection dapat diimplementasikan dalam sebuah program interaktif. Dengan menggunakan metode bisection, permainan mampu menemukan angka yang dipilih oleh pemain secara efisien melalui pembagian rentang secara berulang.

Melalui praktik ini, peserta dapat memahami dasar-dasar pemrograman Python, termasuk penggunaan input, percabangan, perulangan, serta operasi matematika sederhana. Selain itu, implementasi algoritma bisection juga memberikan wawasan tentang pentingnya logika dalam pemrograman untuk menyelesaikan masalah dengan cara yang sistematis dan terstruktur.

Keseluruhan kegiatan ini menunjukkan bahwa meskipun permainan ini sederhana, konsep di baliknya memiliki relevansi luas, terutama dalam bidang komputasi yang membutuhkan efisiensi dalam pencarian data. Pengalaman ini diharapkan dapat menjadi landasan bagi peserta untuk mengembangkan kemampuan pemrograman lebih lanjut di masa depan.