

**LAPORAN ANALISA HASIL PROGRAM Secant**

**SEMESTER GANJIL (3)**

**2020 / 2021**



**NAMA : AKBAR UMAR ALFAROQ**

**NPM : 19081010043**

**FAKULTAS : ILMU KOMPUTER**

**PROGRAM STUDI : TEKNIK INFORMATIKA**

**UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"**

**JAWA TIMUR**

Assalamu'alaikum Warahmatullahi Wabarakatuh

Pada kesempatan kali ini saya akan menjelaskan step by step hasil analisa program *Secant* yang telah saya buat menggunakan bahasa pemrograman **Python** menggunakan Jupyter Notebook. Jupyter Notebook adalah platform Notebook yang didalamnya bisa menjalankan kode program **Python**, format markdown, visualisasi data dll. Sehingga memudahkan user dalam membacanya (seperti membaca *e-book*).

Langsung saja ke laporan analisa

Ketika program pertama dijalankan, program akan meminta user untuk menginputkan dua tebakan awal yaitu  $x_0$  dan  $x_1$



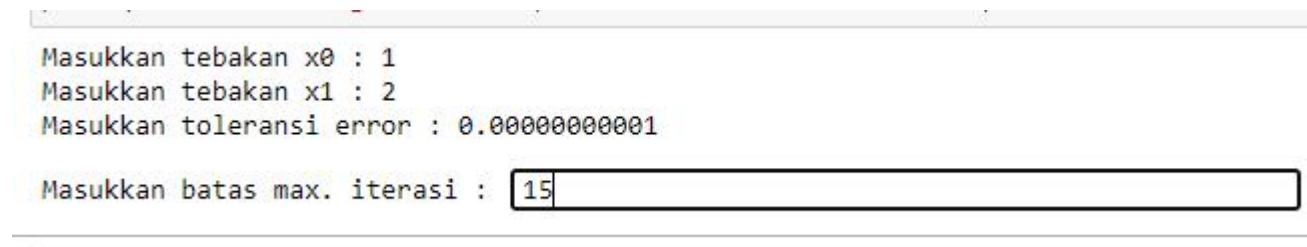
Masukkan tebakan  $x_0$  : 1

Masukkan tebakan  $x_1$  :

Berikut Algoritmanya

```
# Input Data
x0 = float(input("Masukkan tebakan x0 : "))
x1 = float(input("Masukkan tebakan x1 : "))
```

User diminta untuk menginputkan angka tebakan  $x_0$  dan  $x_1$ , dan inputan dikonversi menjadi *float*. Setelah itu user diminta untuk menginputkan toleransi error dan jumlah iterasi maksimum



Masukkan tebakan  $x_0$  : 1

Masukkan tebakan  $x_1$  : 2

Masukkan toleransi error : 0.000000000001

Masukkan batas max. iterasi :

Berikut algoritmanya

```
# Input toleransi error dan max. iterasi
errorTolerant = float(input("Masukkan toleransi error : "))
maxIteration = int(input("Masukkan batas max. iterasi : "))
print("\n")
```

Setelah user menginputkan data dengan benar, maka program akan menjalankan algoritma Secant dibawah ini

```
# Agar dinamis, kita menggunakan paramter agar fungsi re-usable
def secant(x0, x1, errorTolerant, maxIteration):
    printKolom()
    # Iterasi sebanyak jumlah max. iterasi
    for i in range(maxIteration):
        x2 = x1 - (f(x1)*(x1-x0) / (f(x1) - f(x0)))
        # Tampilkan Value
        print("%i.\t%.9f\t%.9f\t%.9f\t%.9f\t%.9f\t%.9f\n" % (i+1, x0, x1, x2, f(x0), f(x1), f(x2)));

        # Geser value
        x0, x1 = x1, x2

        if (abs(f(x2)) <= errorTolerant): break

    # Kembalikan nilai fungsi
    return x2
```

Pada fungsi diatas menerima 4 parameter yaitu  $x_0$ ,  $x_1$ , *errorTolerant*, dan *maxIteration*.  $x_0$  dan  $x_1$  adalah 2 tebakan awal, lalu 2 selanjutnya adalah toleransi eror dan jumlah iterasi maksimum. Pada permulaan fungsi tersebut memanggil "printKolom()" yang mana untuk menampilkan kolom2 atas, berikut pendefinisianya beserta  $f(x)$ .

```
def f(x):
    return math.pow(x, 2) - math.exp(x) + 3

def printKolom():
    print("No.\txi-2\txi-1\tx\tF(xi-2)\tF(xi-1)\tF(x)")
```

Ini adalah hasil dari "printKolom"

```
Masukkan tebakan x0 : 1
Masukkan tebakan x1 : 2
Masukkan toleransi error : 0.0000000001
Masukkan batas max. iterasi : 15
```

| No. | xi-2 | xi-1 | x | F(xi-2) | F(xi-1) | F(x) |
|-----|------|------|---|---------|---------|------|
|-----|------|------|---|---------|---------|------|

Setelah itu pada bagian selanjutnya di bawah ini, melakukan iterasi sebanyak jumlah Iterasi maksimum yang diinputkan user. Lalu menghitung nilai  $x_2$  atau bisa disebut  $x_{i+1}$  sesuai dengan rumus Secant. Selanjutnya data ditampilkan, lalu menggeser valuenya  $x_0 \rightarrow x_1$ , dan  $x_1 \rightarrow x_2$ . Lalu dilakukan pengecekan apakah  $x_2$  sudah memenuhi eror yang ditoleransi, jika iya maka berhenti, jika tidak maka perulangan berlanjut.

```
# Agar dinamis, kita menggunakan paramter agar fungsi re-usable
def secant(x0, x1, errorTolerant, maxIteration):
    printKolom()
    # Iterasi sebanyak jumlah max iterasi
    for i in range(maxIteration):
        x2 = x1 - (f(x1)*(x1-x0) / (f(x1) - f(x0)))
        # Tampilkan Value
        print("%i.\t%.9f\t%.9f\t%.9f\t%.9f\t%.9f\t%.9f\n" % (i+1, x0, x1, x2, f(x0), f(x1), f(x2)));

        # Geser value
        x0, x1 = x1, x2

        if (abs(f(x2)) <= errorTolerant): break

    # Kembalikan nilai fungsi
    return x2
```

Dan yang terakhir adalah mengembalikan nilai akar itu sendiri

```
# Kembalikan nilai fungsi
return x2
```

Output dari program diatas adalah seperti ini

```
Masukkan tebakan x0 : 1
Masukkan tebakan x1 : 2
Masukkan toleransi error : 0.0000000001
Masukkan batas max. iterasi : 15
```

| No. | xi-2        | xi-1        | x           | F(xi-2)      | F(xi-1)      | F(x)         |
|-----|-------------|-------------|-------------|--------------|--------------|--------------|
| 1.  | 1.000000000 | 2.000000000 | 1.767140238 | 1.281718172  | -0.389056099 | 0.268696519  |
| 2.  | 2.000000000 | 1.767140238 | 1.862265070 | -0.389056099 | 0.268696519  | 0.029727714  |
| 3.  | 1.767140238 | 1.862265070 | 1.874098597 | 0.268696519  | 0.029727714  | -0.002698333 |
| 4.  | 1.862265070 | 1.874098597 | 1.873113871 | 0.029727714  | -0.002698333 | 0.000023969  |
| 5.  | 1.874098597 | 1.873113871 | 1.873122541 | -0.002698333 | 0.000023969  | 0.000000019  |
| 6.  | 1.873113871 | 1.873122541 | 1.873122548 | 0.000023969  | 0.000000019  | -0.000000000 |

Akar dari fungsi  $x^2 - \exp^x + 3$  adalah : 1.873122548

Untuk menjalankan program notebook ini bisa menggunakan Jupyter Notebook (lokal), atau jika online bisa menggunakan Google Colab <https://colab.research.google.com/notebooks/>, lalu upload file *.ipynb* nya

Cukup sekian yang bisa saya sampaikan untuk Analisa Hasil Program Secant ini, jika ada kekurangan saya mohon maaf.

Terima Kasih Bu Yisti atas materi yang telah disampaikan tentang Secant ini, Sekian dan Terima Kasih

Wassalamu'alaikum Warahmatullahi Wabarakatuh