LEMBAR PENGESAHAN

KN - 01

DERET TAYLOR

Nama	: Rendy Putra Pratama
NPM	: 140310230037
Hari / Tanggal	: Jumat/ 15 Maret 2024
Waktu / Sesi	: 08.00 WIB / 1
Asisten	: M. Naufaldi D.

Pretest	**Laporan Akhir**

Jatinangor, 15 Maret 2024
Asisten

M. Naufaldi D.

1. Tugas Praktikum

Lakukan ekspansi deret Taylor/Maclaurin dari beberapa fungsi dibawah ini dan buatlah program untuk menghitungnya.

a. $f(x) = sin\ x$ di sekitar $x = 0$ dengan tingkat kesalahan $\varepsilon = 0.0000001$

b. $f(x) = ln\ x$ di sekitar $x = 1$ hingga suku ke-8

c. $f(x) = e^x$ di sekitar $x = 0$ dengan tingkat kesalahan $\varepsilon = 0.000000001$

1.1. Percobaan 1

1.1.1. Listing Program

```
import math as mt
import numpy as np

exactValue = mt.sin(mt.radians(30))
numericValue = 0.0
fx = mt.radians(30)
n = 0
a = 0
while True:
    numericValue += (((-1)**n)*((fx-
a)**(2*n+1))/mt.factorial(2*n+1))
    e = (exactValue-numericValue)
    n += 1
    if n == 4:
        break
    print(f"approx = {numericValue: .8f} \t error = {abs(e):
.8f} \t iterasi = {n}")
print(f"Hasil analitik = {exactValue: .8f}")
print(f"Hasil numerik = {numericValue: .8f} dengan error =
{abs(e): .8f}, pada iterasi = {n}")
```

1.1.2. Output Program

```
approx =  0.52359878      error =  0.02359878     iterasi = 1
approx =  0.49967418      error =  0.00032582     iterasi = 2
approx =  0.50000213      error =  0.00000213     iterasi = 3
Hasil analitik =  0.50000000
Hasil numerik =  0.49999999 dengan error =  0.00000001, pada
iterasi = 4
```

1.1.3. Analisis Program

Pada program diatas praktikan memulai dengan melakukan inisiasi variabel yang diketahui seperti nilai exactValue = 0.5 (hasil dari sin(30)), numericValue dengan nilai awal 0.0, fx = 30 derajat sebagai x yang akan dihampirkan dan a = 0 sebagai hampirannya.

Selanjutnya praktikan mengisi numericValue dengan penjumlahan berulang dari deret :

$$-\frac{1^n(x-a)^{2n+1}}{(2n+1)!}$$

Hasil dari penjumlahan setiap n suku pada deret diatas akan disimpan dalam variabel numericValue. Operasi penjumlahan setiap suku pada deret di bahasakan dengan kode +=. Program ini akan berhenti Ketika diberikan kondisi atau syarat tertentu – dalam kasus ini adalah n = 4.

Kemudian untuk menentukan error, praktikan menggunakan persamaan (1.3) pada modul yaitu :

$$error = exact\ value - numeric\ value$$

Hasilnya praktikan mendapatkan nilai error = 10^{-7} pada iterasi ke-4 dengan nilai numerik = **0.49999999.**

1.2. Percobaan 2

1.2.1. Listing Program

```
from math import *
error = 1
numValue = 0.0 # Inisiasi hasil numerik dari f(x)
exactValue = 0.6931471806
x = 2
n = 0
while True:
    numValue += ((-1)**(n+2)*(2-1)**(n+1)/(n+1))
    error = (exactValue - numValue)
    n+=1
    if n == 8:
        break
    print(f"Hasil perhitungan numerik = {numValue: .8f} \t
Error = {abs(error): .9f} \t Iterasi = {n}")
print(f"Hasil analitik = {exactValue}")
print(f"Aproksimasi : {numValue : .9f}, error =
{abs(error): .9f} iterasi {n:10}")
```

1.2.2. Output Program

```
Hasil perhitungan numerik =  1.00000000    Error =  0.306852819
Iterasi = 1
Hasil perhitungan numerik =  0.50000000    Error =  0.193147181
Iterasi = 2
Hasil perhitungan numerik =  0.83333333    Error =  0.140186153
Iterasi = 3
Hasil perhitungan numerik =  0.58333333    Error =  0.109813847
Iterasi = 4
```

```
Hasil perhitungan numerik =  0.78333333    Error =  0.090186153
Iterasi = 5
Hasil perhitungan numerik =  0.61666667    Error =  0.076480514
Iterasi = 6
Hasil perhitungan numerik =  0.75952381    Error =  0.066376629
Iterasi = 7
Hasil analitik = 0.6931471806
Aproksimasi :  0.63452381, error =  0.05862337 iterasi: 8
```

1.2.3. Analisis Program

Program dilakukan inisiasi seperti pada program percobaan 1. Namun, praktikan mencoba membuat pendekatan numerik dari ln(2) dengan berpusat di sekitar a = 1 (diketahui pada modul).

Kemudian, praktikan membuat deret taylor dari ln(x) disekitar a = 1 sebagai berikut:

$$\sum_{n=1}^{8}\left(\frac{(-1)^{n+2}(x-a)^{n+1}}{n+1}\right)$$

Bentuk diatas telah disesuaikan dengan iterasi n dimulai dari nol. Bentuk sebenarnya adalah sebagai berikut:

$$\sum_{n=1}^{8}\frac{-1^{(n-1)}(x-a)^{n}}{n}$$

Sebagaimana persamaan deret diatas, program akan berhenti pada n = 8. Oleh sebab itu, praktikan memberikan kondisi n = 8 untuk *break* pada program agar pengulangan dapat dihentikan.

Selanjutnya, praktikan menghitung error setiap iterasi dengan e = nilai analitik – nilai numerik. Sehingga didapat hasil akhir dari program diatas sebagai berikut :

Hasil analitik = 0.6931471806

Aproksimasi : 0.63452381, error = 0.05862337 iterasi: 8

1.3. Percobaan 3

1.3.1. Listing Program

```
from math import *
exactValue = 0.0
numericValue = 0.0
fx = exp(1) # e**1 = 2.718...
x = 1
error = 1
while error > 0.0000000001:
    numericValue += (x**n/factorial(n))
    error = (fx - numericValue)

    print(f"approx = {numericValue: .8f} \t\t error =
{abs(error)}")
print(f"Approksimasi Deret Taylor dari e**x =
{numericValue: .8f} dengan error = {abs(error): .11f}")
```

1.3.2. Output Program

```
approx =  1.00000000          error = 1.718281828459045
approx =  2.00000000          error = 0.7182818284590451
approx =  2.50000000          error = 0.2182818284590451
approx =  2.66666667          error = 0.05161516179237857
approx =  2.70833333          error = 0.009948495125712054
approx =  2.71666667          error = 0.0016151617923787498
approx =  2.71805556          error = 0.0002262729034896438
approx =  2.71825397          error = 2.7860205076724043e-05
approx =  2.71827877          error = 3.0586177750535626e-06
approx =  2.71828153          error = 3.0288585284310443e-07
approx =  2.71828180          error = 2.7312660577649694e-08
approx =  2.71828183          error = 2.260552189881082e-09
approx =  2.71828183          error = 1.7287637987806193e-10
approx =  2.71828183          error = 1.2285727990501982e-11
Approksimasi Deret Taylor dari e**x =  2.71828183 dengan error =
0.00000000001
```

1.3.3. Analisis Program

Praktikan memulai program menghitung nilai $f(x)$ dengan melakukan inisiasi variabel yang diketahui yaitu : *exactValue, numericValue, error,* $f(x)=e^{1}$, dan $x=1$. Nilai error = 1 diinisiasikan agar syarat pada *while* terpenuhi sehingga program didalamnya dapat dieksekusi.

Ketika ingin menentukan hasil pendekatan numerik, praktikan menginisiasikan numericValue = 0.0 sebagai nilai awal perhitungan. Kemudian didalam while, nilai dari numericValue akan dijumlahkan dari hasil $\frac{x^n}{n!}$. Penjumlahan tersebut didasari oleh bentuk umum deret taylor berikut:

$$e^x = \frac{\sum x^n}{n!}$$

Proses menjumlahkan tiap suku pada ekspansi diatas ialah dengan code +=, maknanya ialah setiap hasil dari perhitungan `x**n/factorial(n))` akan dijumlahkan dengan suku sebelumnya. Hal it uterus berlangsung hingga kondisi yang ditentukan – dalam hal ini `error > 0.0000000001` bernilai *false*.

Setelah mendapatkan nilai dari numericValue program akan menentukan error dengan error = exactValue – numericValue sampai dengan error < 0.0000000001. Setelah kondisi error < 0.0000000001 terpenuhi maka syarat *while* error > 0.0000000001 tidak terpenuhi dan program akan terhenti.

Didapat hasil dari proses diatas adalah Aproksimasi Deret Taylor dari e^x = 2.71828183 dengan error = 0.00000000001.

2. Tugas Akhir

2.1. Mengapa metode numerik penting untuk dipelajari? Padahal hasil yang dihasilkan dari metode tersebut masih memiliki kesalahan. Jelaskan!

Jawab :

Metode numerik penting untuk dipelajari karena metode tersebut dapat mempermudah perhitungan suatu fungsi dengan cara aritmatika – tidak memerlukan kalkulus. Oleh sebab itu, meskipun metode numerik masih memiliki error, error tersebu dapat disesuaikan dengan pendekatan/perhitungan yang berulang (*looping*). Sedangkan, metode analitik memerlukan *skill* khusus dalam perhitungannya yaitu kalkulus. Apabila terdapat kesalahan dalam menggunakan kalkulus, maka perhitungan memiliki probabilitas untuk salah.

2.2. Hampiri persamaan berikut, yaitu

$$f(x) = 0.5x^2 - 0.25x + 1.2$$

Untuk $0 \leq x \leq 1$ dengan menggunakan deret Taylor sampai suku ke-0, suku ke-1, suku ke-2, suku ke-3, suku ke -4 dengan menggunakan nilai nol sebagai garis bilangan. Bandingkan hasil pendekatan numerik dengan hasil perhitungan analitik, hitung error numeriknya.

Jawab :

2.2.1 Listing Program

```
error = 1
numericValue = 0.0
x = 0.5 # 0<= x <= 1
a = 0.6 # 0<=a <= 1
exactValue = 0.95
while error > 0.0000000000001:
    numericValue += (-0.5 * x**2 - 0.25 * x + 1.2 + (-x -
0.25)*(x-a) - 1*(x-a)**2/2)
    error = (exactValue - numericValue)
    print(f"Hasil analitik = {exactValue}")
    print(f"Hasil pendekatan numerik = {numericValue} \t
error = {abs(error)}")
```

2.2.2 Output Program

```
Hasil analitik = 0.95
Hasil pendekatan numerik = 1.02, error =
0.07000000000000006
```

2.2.3 Analisis Program

Pada program diatas berfungsi untuk menentukan nilai $f(x)$ pada interval $0 \leq x \leq 1$, praktikan mengambil nilai uji pada $x = 0.5$ dengan $a = 0.6$. Hal tersebut dilakukan agar perhitungan berada didalam rentang $0 \leq x \leq 1$. Langkah yang dilakukan ialah memulai inisiasi variabel yang diketahui :

$$error = 1, numericValue = 0.0, x = 0.5, a = 0.6, exactValue = 0.95$$

Pada program praktikan memberikan nilai error = 1 sebagai kondisi *true* pada syarat *while error > 0.0000000000001*. Ketika syarat tersebut benar, program akan masuk kedalam *looping* yang ada didalam *while*. Selanjutnya, *numericValue* diinisiasikan dengan 0.0 sebagai wadah menyimpan hasil aproksimasi numerik pada *while*. Untuk mendapatkan output, praktikan menurunkan persamaan $f(x)$ kedalam bentuk deret taylor. Berikut turunan setiap suku ke – n:

$$f(x) = 0.5x^2 - 0.25x + 1.2$$

$$f'(x) = -x - 0.25$$

$$f''(x) = -1$$

$$f'''(x) = 0$$

$$f''''(x) = 0$$

Maka, didapat formasi untuk ekspansi $f(x)$ adalah :

$$f(x) = f(x) + f'(x)(x-a) + \frac{f''(x)(x-a)}{2!} + \frac{f'''(x)(x-a)}{3!} + \frac{f''''(x)(x-a)}{4!} + \cdots.$$

$$f(x) = 0.5x^2 - 0.25x + 1.2 + (-x - 0.25)(x-a) + \frac{(-1)(x-a)^2}{2!} + 0 + 0$$

Setelah didapat formasi ekspansi deret Taylor dari $f(x)$ perhitungan akan dilakukan dengan variabel yang telah ditentukan sampai syarat dari *while* yaitu *error > 0.0000000000001* adalah salah atau *error < 0.0000000000001*.

2.3. Hitung nilai fungsi

$$f(x) = \ln(\cos(x))$$

Di titik $x = \frac{\pi}{4}$ dengan pendekatan deret Taylor sampai suku ke-2, suku ke-3, dan suku ke -4. Bandingkan hasilnya dengan hasil perhitungan kalkulator dan tentukan error numeriknya.

Jawab :

2.3.1 Listing Program

```
# Hitung fungsi f(x) = ln(cos(x)) dengan x disekitar pi/4
from math import *
numValue = 0.0 # Inisiasi hasil numerik dari f(x)
exactValue = -0.3465735903
cos45 = sqrt(2)/2
n = 1
while True:
    numValue += (((-1)**(n+1)*(cos45-1)**n)/n)
    error = (exactValue - numValue)
    n+=1
    if n == 14:
        break
    print(f"Hasil perhitungan numerik = {numValue: .8f} \t
Error = {abs(error): .9f} \t Iterasi = {n-1}")
print(f"Hasil analitik = {exactValue}")
print(f"Aproksimasi : {numValue : .9f}, error = {abs(error):
.9f} iterasi : {n}")
```

2.3.2 Output Program

```
Hasil perhitungan numerik = -0.29289322    Error =  0.053680371
Iterasi = 1
Hasil perhitungan numerik = -0.33578644    Error =  0.010787153
Iterasi = 2
Hasil perhitungan numerik = -0.34416186    Error =  0.002411731
Iterasi = 3
Hasil perhitungan numerik = -0.34600169    Error =  0.000571903
Iterasi = 4
Hasil perhitungan numerik = -0.34643279    Error =  0.000140804
Iterasi = 5
Hasil perhitungan numerik = -0.34653801    Error =  0.000035582
Iterasi = 6
Hasil perhitungan numerik = -0.34656442    Error =  0.000009166
Iterasi = 7
Hasil perhitungan numerik = -0.34657119    Error =  0.000002396
Iterasi = 8
Hasil perhitungan numerik = -0.34657296    Error =  0.000000634
Iterasi = 9
Hasil perhitungan numerik = -0.34657342    Error =  0.000000169
Iterasi = 10
Hasil perhitungan numerik = -0.34657354    Error =  0.000000046
Iterasi = 11
Hasil perhitungan numerik = -0.34657358    Error =  0.000000012
Iterasi = 12
Hasil analitik = -0.3465735903
Aproksimasi : -0.346573587, error =  0.000000003 iterasi: 14
```

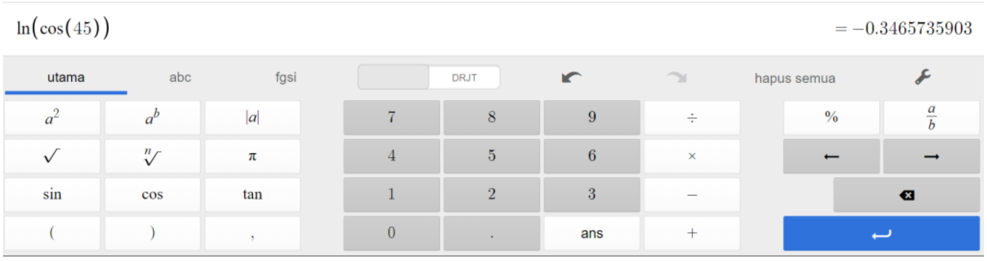
2.3.3 Analisis Program

Pada tugas akhir ketiga, untuk menentukan nilai ln(cos(x)), praktikan memulai program dengan menginisiasikan variabel yang telah diketahui seperti nilai cos45, dan hasil perhitungan analitik dari ln(cos(45)). Selanjutnya, praktikan mencari referensi penjumlahan deret taylor dari ln(cos(x)) yaitu $\sum\left(\frac{(-1)^{n+1}(\cos(x)-1)}{n}\right)$.

Untuk memulai iterasi, praktikan menginisiasikan nilai true pada while, kemudian proses looping akan berulang sampai suku ke-14 dan program akan dihentikan.

Dari program tersebut didapat hasil numerik adalah **-0.346573587** dengan error **0.000000003** pada iterasi ke 14.

Sedangkan hasil kalkulatornya ialah :



3. **Kesimpulan**

Pengetahuan tentang ekspansi deret taylor suatu fungsi dapat sangat memudahkan perhitungan dengan program. Hal ini dikarenakan program computer hanya melakukan perhitungan aritmatika. Perhitungan aritmatika didapat dari hasil ekspansi taylor yang telah dilakukan kepada suatu fungsi. Sehingga hasil dari penurunan setiap suku pada fungsi f(x) dapat dijumlahkan sebagaimana yang telah diformulasikan pada deret taylor.

Tantangan pada modul ini adalah menentukan turunan ke-n pada fungsi yang ingin diaproksimasi menggunakan deret taylor. Terkhusus fungsi-fungsi kompleks seperti pada tugas akhir nomor 3. Langkah penurunan tersebut seringkali menjadi penghambat apabila ekspansi deret taylor harus dilakukan hingga n yang lebih banyak. Maknanya, praktikan harus melakukan penurunan secara berulang hingga n yang diminta pada modul.

DAFTAR PUSTAKA

C. Chapra, S., & P. Canale, R. (2015). *Numerical Methods in Engineering* (7th ed.). McGraw-Hill Education.

Jack. (n.d.). *Looking for Taylor series expansion of*. math.stackexchange.com. Retrieved March 17, 2024, from https://math.stackexchange.com/q/1443789

Kiusalaas, J. (2009). *NUMERICAL METHODS IN ENGINEERING* (2nd ed.). New York: Cambridge University Press.

Munir, Rinaldi. (2017). *Metode Numerik.* Bandung: Penerbit INFORMATIKA.