DERIVATIF NUMERIK: MENENTUKAN PERKIRAAN SOLUSI NUMERIK DENGAN METODE BEDA TENGAH (DIFERENSIASI PUSAT)

Deva Anjani Khayyuninafsyah (122450014), Chevando Daffa Pramanda (122450095), Rafly Prabu Darmawan (122450140), Novelia Adinda (122450104), Nabiilah Putri Karnaia (122450029)

Fakultas Sains, Program Studi Sains Data, Institut Teknologi Sumatera
Jl. Terusan Ryacudu, Way Huwi, Kec. Jatiagung, Kabupaten Lampung Selatan, Lampung 35365

Email:
deva.122450014@student.itera.ac.id, chevando.122450095@student.itera.ac.id, rafly.122450140@student.itera.ac.id, novelia.122450104@student.itera.ac.id, nabiilah.122450029@student.itera.ac.id

1. Pendahuluan

Derivatif numerik adalah metode-metode numerik yang digunakan untuk menghitung hampiran nilai turunan (derivatif) dari suatu fungsi matematika. Turunan adalah ukuran perubahan fungsi terhadap variabelnya. Derivatif numerik berfokus pada pendekatan komputasional untuk menghitung turunan, yang seringkali sulit atau bahkan tidak mungkin dihitung secara analitis. Teknik pengoptimalan turunan numerik dapat diterapkan di berbagai bidang akademik dan industri, seperti pembelajaran mesin, ilmu fisika, kimia, rekayasa, analisis keuangan, pengoptimalan jaringan listrik, dan logistik.

Terdapat tiga pendekatan untuk menghitung turunan numerik, yaitu Metode Beda Hingga dari Maju (Forward Difference), Metode Beda Hingga dari Mundur (Backward Difference), dan Metode Beda Hingga dari Tengah/Terpusat (Central Difference). Ketiga pendekatan tersebut digunakan dalam konteks derivatif numerik untuk memperkirakan nilai turunan suatu fungsi matematika di titik tertentu. Setiap pendekatan mempunyai kelebihan dan kekurangannya tergantung pada jenis fungsi yang dianalisis serta konteks penerapannya.

Dalam kesempatan kali, akan digunakan metode beda terpusat untuk menghitung turunan suatu fungsi di suatu titik dengan cara membandingkan nilai fungsi di titik tersebut dengan nilai fungsi di titik sebelum dan sesudahnya. Dibandingkan dengan metode beda maju atau mundur, metode ini memberikan hasil yang lebih akurat untuk fungsi kompleks atau tidak linier karena memperoleh informasi dari kedua sisi titik yang diinginkan (Chapra & Canale, 2010).

2. Metode

Artikel ini mengambil topik terkait Derivatif Numerik pada tahapan Advanced dalam Tugas Individu Modul 3 Praktikum Pemrograman Berbasis Fungsi. Dalam kasus ini, digunakan metode beda hingga yang diukur dari tengah (pusat) untuk mengembalikan nilai derivatif numerik f di titik x di mana f merupakan fungsi polinomial berorde dua. Metode beda hingga merupakan metode numerik yang digunakan untuk menyelesaikan permasalahan matematis dengan pendekatan turunan, dalam kasus ini dilakukan pengukuran dari tengah (Mufida, 2022, 13-14). Melalui metode beda hingga dari tengah ini, akan diperoleh solusi numerik persamaan fungsi yang digunakan (Nikmah, 2015, 5). Perkiraan yang mendekati solusi numerik tersebut nantinya dapat diimplementasikan dalam berbagai pengimplementasian lainnya.

```
def derivative(f, x, h=1e-5):
    return (f(x + h) - f(x - h)) / (2 * h)
```

Gambar 1 Kode Metode Beda Tengah

Kode di atas menggunakan metode beda hingga yang diukur dari tengah untuk turunan biasa. Secara matematis, fungsi tersebut berasal dari pengurangan rumus berikut ini.

$$f(x + h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x)$$

$$f(x - h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x)$$

Diperoleh,

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

Dengan demikian, metode beda hingga dari tengah (metode beda pusat) dapat dirumuskan seperti berikut yang kemudian diimplementasikan dalam kode pemrograman di atas.

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

Di samping itu, pendekatan metode yang digunakan adalah Pemrograman Berbasis Fungsi dengan mengaplikasikan beberapa fungsi *Python* di bawah ini.

2.1. *Built-In Function*

2.1.1. *def*

Fungsi bawaan *def* digunakan untuk mendefinisikan suatu fungsi. Pada kasus ini, *derivative*() dan f() adalah nama yang diberikan untuk fungsi *def*. Dengan kata lain, *def* mendefinisikan fungsi *derivative*() untuk menemukan perkiraan solusi numerik, sedangkan fungsi f() didefinisikan oleh *def* sebagai fungsi yang memuat persamaan untuk diselesaikan.

2.1.2. *return*

Fungsi *return* digunakan untuk mengembalikan nilai dari sebuah fungsi. Ketika suatu fungsi dijalankan dan mencapai pernyataan "*return*", maka proses tersebut akan berhenti, dan nilai yang ditentukan dalam "*return*" akan dikembalikan kepada pemanggil fungsi. Hal ini mempermudah pemrograman dengan memungkinkan sistem untuk menyederhanakan proses.

2.1.3. *print*()

Fungsi *print*() digunakan untuk menampilkan *output* dari sebuah program, baik berupa angka, teks, atau variabel.

2.2. *User-Defined Function*

2.2.1. *derivative*()

Fungsi *derivative*() adalah nama dari sebuah fungsi *def* yang bertujuan untuk menghitung turunan dari suatu fungsi dengan menggunakan pendekatan numerik.

2.2.2. f()

Fungsi f() adalah nama dari sebuah fungsi *def* biasanya digunakan untuk melakukan operasi atau perhitungan.

3. Pembahasan

Pada deret kode pertama, pendefinisiannya menggunakan fungsi *derivative*() yang merupakan fungsi untuk menghitung turunan numerik pada suatu fungsi f(x) menggunakan pendekatan diferensiasi numerik sentral atau *finite difference method* untuk perkiraan turunan. '*def derivative*(f, x, h=1e-5)' adalah pendefinisian fungsi dengan tiga buah parameter, yaitu:

3.1. 'f' sebagai fungsi yang ingin diturunkan.
3.2. 'x' sebagai titik di mana turunannya diketahui.
3.3. 'h' dengan nilai *default* '1e-5'.

```python
# mendefinisikan fungsi untuk mengembalikan nilai derivatif numerik f di titik x dengan metode perbedaan hingga dari tengah (pusat)
def derivative(f, x, h=1e-5):
  return (f(x + h) - f(x - h)) / (2 * h)

# mendefinisikan fungsi f, dimisalkan kita memiliki fungsi f(x) = x^2 + 2x
def f(x):
  return x ** 2 + 2 * x

# ambil sebarang titik, misal x = 2, lalu hitung derivatif f'(x) dari f(x) = x^2 + 2x pada titik x = 2
x = 2
nilai_derivatif = derivative(f, x)
print(f"Nilai derivatif f'(x) pada x = {x} adalah: {nilai_derivatif}.")
```

Gambar 2 Kode

Ketika dieksekusi, fungsi *derivative*() akan bekerja dengan cara memilih nilai fungsi yang terdapat pada fungsi 'f(x+h)' dan 'f(x-h)' kemudian menghitung selisih dari kedua nilai lalu memperkirakan turunannya dengan cara membagi selisih tersebut dengan 2h. Nilai hasilnya akan dikembalikan oleh fungsi *return* dalam fungsi tersebut.

Selanjutnya adalah mendeklarasikan fungsi 'f' dengan sebuah parameter 'x' dan fungsi yang dimiliki adalah $f(x) = x^2 + 2x$ atau pada kode ditulis dengan x**2 + 2 * x. Fungsi 'f(x)' tersebut nantinya akan mengembalikan hasil ekspresi dari x**2 + 2 * x melalui fungsi *return*. Lalu, untuk menentukan turunan dari fungsi 'f(x)' dengan metode ini, maka nilai x harus diinisialisasikan terlebih dahulu, yaitu x = 2. Inisialisasi nilai x tersebut nantinya akan menjadi suatu parameter untuk mengeksekusi nilai derivatif.

Langkah selanjutnya, yakni membuat variabel baru bernama 'nilai_derivatif' untuk menyimpan fungsi *derivative*(). Ini sesuai dengan konsep *higher-order-function* di mana fungsi dapat disimpan dalam sebuah variabel. Langkah ini bertujuan untuk mempermudah proses pemanggilan fungsi ketika hasil ingin dicetak.

Berdasarkan uraian fungsi-fungsi di atas, ketika program dieksekusi secara keseluruhan, dengan nilai 'h' yang sudah ditentukan sebesar 1e-5, maka pada fungsi *derivative*() yang dipanggil melalui variabel 'nilai_derivatif' tinggal digunakan parameter 'f' sebagai fungsi $f(x) = x^2 + 2x$ yang ingin diturunkan dan 'x' sebagai titik tempat untuk menghitung turunan yang telah diinisialisasi sebagai x = 2. Saat fungsi *derivative*() dipanggil melalui variabel 'nilai_derivatif', langkah ini tentunya melibatkan pengurangan nilai 'f(x+h)' dengan 'f(x-h)' di sekitar titik x dan hasil selisihnya dibagi dengan '2h' sehingga didapatkan perkiraan turunan pada titik x. Pada dasarnya, proses pengeksekusian tersebut telah melibatkan semua fungsi yang telah didefinisikan sebelumnya. Kemudian, hasilnya akan dicetak menggunakan fungsi *print*() sebagai berikut.

```
Nilai derivatif f'(x) pada x = 2 adalah: 5.999999999994897.
```

Gambar 3 Hasil

Dengan demikian, perkiraan solusi numerik berupa nilai turunan dari fungsi $f(x) = x^2 + 2x$ pada titik x = 2 adalah 5.999999999994897.

4. Kesimpulan

Dalam artikel ini, dibuat sebuah program terdiri dari fungsi yang menerima fungsi matematika ‘f’ dan titik ‘x’ lalu mengembalikan nilai derivatif numerik ‘f’ di titik ‘x’ menggunakan metode beda hingga dari tengah. Dengan menerapkan *built-in function* dan *user-defined function* serta konsep *higher-order-function*, program tersebut dapat berjalan sebagaimana mestinya dengan melibatkan semua fungsi yang telah didefinisikan. Dari proses eksekusi tersebut, diperoleh bahwa nilai turunan dari fungsi $f(x) = x^2 + 2x$ pada titik x = 2 adalah 5.999999999994897 atau mendekati angka 6. Dengan demikian, tujuan pengimplementasian fungsi ini sudah tercapai.

5. Referensi

Chapra, S. C., & Canale, R. P. (2010). *Numerical Methods for Engineers*. McGraw-Hill Education. https://civil-team.weebly.com/uploads/2/5/8/2/25829430/numerical_methods_for_engineers_chapra_canale_6th_ed.pdf

Mufida, F. I. (2022, Juni). Skema Numerik Persamaan Difusi Dua Dimensi dengan Menggunakan Metode Beda Hingga Pusat-Saul'yev Eksplisit. *Etheses of Maulana Malik Ibrahim State Islamic University*, 13-14. http://etheses.uin-malang.ac.id/37490/1/15610094.pdf

Nikmah, J. (2015, Juni). Penyelesaian Numerik Persamaan Telegraf Menggunakan Metode Beda Hingga Skema Eksplisit. *Etheses of Maulana Malik Ibrahim State Islamic University*, 5. http://etheses.uin-malang.ac.id/6403/1/10610031.pdf