# LAPORAN PRAKTIKUM METODE NUMERIK

# INTERPOLASI

**Ditulis Oleh :**

**Ilham Nur Romdoni**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS SEBELAS MARET**

**SURAKARTA**

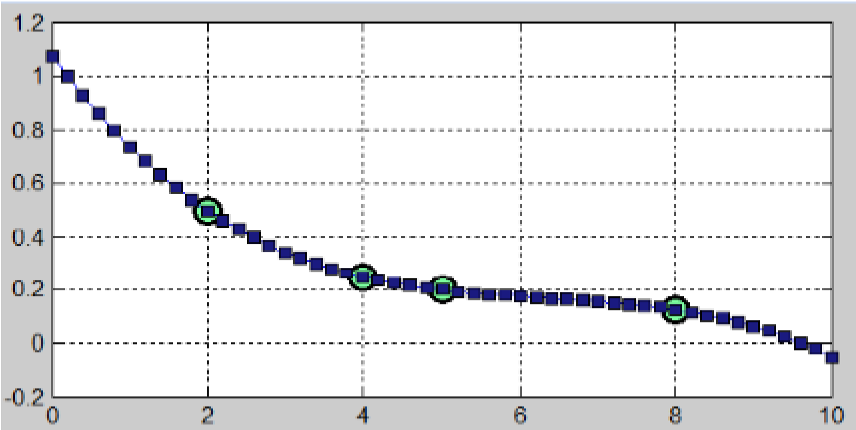
# BAB I
# PENDAHULUAN

## 1.1 Tujuan Praktikum

1. Mengetahui bentuk algoritma dalam penentuan P(x) dengan teknik Polinomial Newton dari fungsi f(x).
2. Mengetahui bentuk grafik yang diperoleh dari fungsi f(x) dan P(x) dari teknik interpolasi.

## 1.2 Dasar Teori

Teknik Interpolasi merupakan teknik untuk mendapatkan fungsi yang melewati semua titik dari sebuah set data diskrit.




Perbedaan interpolasi dan aprosimaksi adalah interpolasi melewati semua titik data yang ada dengan tepat sedangkan pada aprosimaksi, semua titik data tidak dilewati dengan tepat.

Ada beberapa teknik interpolasi untuk menghitung fungsi polinomial :

1. Polinomial Lagrage
2. Polinomial Newton
3. Polinomial Hermite
4. Polinomial Taylor
5. Polinomial Rasional

Salah satu teknik yang sering digunakan adalah Polinomial Newton. Bentuk formulanya dapat dirumuskan di bawah ini :

$$P(x) = f[x_1] + f[x_1, x_2](x - x_1) + f[x_1, x_2, x_3](x - x_1)(x - x_2) + \cdots$$
$$+ f[x_1, \ldots , x_3](x - x_1) \ldots (x - x_n)$$

Untuk menentukannya dapat dihitung dengan teknik selisih bagi dua atau *divide difference* :

$$\begin{matrix} f[x_1] & 0 & 0 & \cdots & 0 \\ f[x_2] & f[x_1,x_2] & 0 & \cdots & 0 \\ f[x_3] & f[x_2,x_3] & f[x_1,x_2,x_3] & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ f[x_k] & f[x_{k-1},x_k] & f[x_{k-2},x_{k-1},x_k] & \cdots & f[x_1,x_2,\ldots,x_k] \end{matrix}$$

***Pretest***

1. Apa yang Anda ketahui tentang teknik interpolasi ?

   Teknik Interpolasi merupakan teknik untuk mendapatkan fungsi yang melewati semua titik dari sebuah set data diskrit.
2. Apa bedanya antara teknik interpolasi dengan aprosikmasi ?

   Perbedaan interpolasi dan aprosimaksi adalah interpolasi melewati semua titik data yang ada dengan tepat sedangkan pada aprosimaksi, semua titik data tidak dilewati dengan tepat.
3. Sebutkan 3 teknik interpolasi untuk fungsi polinomial !

   Ada beberapa teknik interpolasi untuk menghitung fungsi polinomial :

   a. Polinomial Lagrage

   b. Polinomial Newton

   c. Polinomial Hermite
4. Menurut Anda, semisal ada 50 data untuk menghitung nilai dari fungsi f(x) dengan teknik interpolasi. Apakah teknik tersebut tepat ? Berikan alasan !

   Menurut saya tidak terlalu tepat jika memperhatikan proses waktu yang dibutuhkan. Penghitungan nilai dari fungsi f(x) dengan teknik interpolasi akan membutuhkan waktu yang lebih lama. Karena banyaknya data, algoritma yang dijalankan akan semakin panjang setidaknya lebih panjang daripada sistem persamaan linier. Tetapi walaupun begitu, teknik interpolasi akan melewati semua titik data yang ada dengan tepat.

# BAB II
# PEMBAHASAN

## 2.1 Permasalahan

### 2.1.1 Praktikum

Diberikan bentuk fungsi:

$$f(x) = x^3 - 4x$$

Tentukan Polinomial Newton $P_3(x)$. Dengan $x_1$=1, $x_2$=2, $x_3$=3 dan $x_4$=4

### 2.1.2 *Posttest*

Tentukan P(x) dari fungsi $f(x) = \cos(x)$ dengan menggunakan 5 titik yaitu $x_1 = 0, x_2 = 1, x_3 = 2, x_4 = 3,$ dan $x_5 = 4$ ! Tampilan juga bentuk grafik antara P(x) dengan f(x) di **lampiran** saja !

## 2.2 Algoritma

Untuk menyelesaikan kedua permasalahan di atas akan digunakan salah satu teknik interpolasi yaitu Polinomial Newton. Pada implementasi MATLAB untuk Polinomial Newton, dibuat tiga fungsi dengan nama SelisihBagi, plinom dan PolinomialNewton. *Source code* MATLAB akan disusun dengan algoritma sebagai berikut.

1. Pembuatan fungsi SelisihBagi
   - Inisialisasi nilai x dan y sebagai nilai variabel yang di-*input*-kan.
   - n adalah panjang dari x.
   - D diinisialisasikan sebagai matriks zeros(n) yaitu matriks persegi yang berisi bilangan 0 yang dimensinya sepanjang n.
   - D selanjutnya didefinisikan sebagai matriks yang semua baris pada kolom ke-1 berisi nilai y pada baris pertama sampai kolom ke n.
   - Dilakukan perulangan for untuk menghitung selisih bagi dengan perulangan dari kolom ke-2 hingga n di mana di dalamnya juga terjadi perulangan for dari kolom yang sama dengan baris hingga n. Didefinisikan D(k,j) dengan rumus hasil selisih bagi yaitu (D(k,j-1) – D(k, j-1) – D(k-1,j-1)) /x(k) – x(k-j+1)); Perulangan merupakan penerapan rumus Selisih Bagi.
2. Pembuatan fungsi plinom
   - Inisialisasi nama fungsi plinom dengan parameter. Hasil dari plinom dimasukkan pada variabel ypol.

- Didefinisikan has sama dengan 1.
- Perulangan i dari 2 hingga k. Nilai has yang baru adalah has dikalikan nilai xx dikurangi x dari (i-1).
- Nilai ypol adalah sama dengan has. ypol inilah yang akan digunakan pada fungsi PolinomialNewton.

3. Pembuatan fungsi PolinomialNewton
   - Inisialisasi nama fungsi dan parameter.
   - Didefinisikan variabel xx sebagai *range* dari grafik. Bilangan 0:1:6 menunjukkan *range* grafik dari 0 hingga 6 dengan pertambahan nilai 1. Dan variabel yy sebagai matriks D baris 1 kolm1.
   - Perulangan untuk menutup setiap titik pada grafik. Perulangan dilakukan pada baris ke-2 hingga n di mana variabel baru dari yy didefinisikan sebagai yy ditambah matriks D dari baris k kolom k dikalikan dengan fungsi plinom dengan *input*-an xx, x dan k. Hal ini merupakan penerapan dari rumus polinom newton yaitu

$$P(x) = f[x_1] + f[x_1, x_2](x - x_1) + f[x_1, x_2, x_3](x - x_1)(x - x_2) + \cdots \\ + f[x_1, \dots, x_3](x - x_1) \dots (x - x_n)$$

   - Ditampilkan nilai yy yang merupakan P(x) yang ditemukan.
   - Digunakan perintah plot untuk menampilkan grafik antara P(x) dengan f(x).

## 2.3 Implementasi MATLAB

Implementasi algoritma pada MATLAB adalah sebagai berikut.

```
% Ilham Nur Romdoni, M0520038

function D = SelisihBagi(X,Y)
    x = X;
    y = Y;
    n = length(x);
    D = zeros(n)
    D(:,1) = y(1:n)
    for j=2:n
        for k=j:n
            D(k,j) = (D(k,j-1) - D(k-1,j-1))/(x(k) - x(k-j+1));
        end
    end
end
```

```
% Ilham Nur Romdoni, M0520038

function ypol = plinom(xx,x,k)
has = 1;
for i=2:k
    has = has.*(xx - x(i-1));
end
ypol = has;
end
```

```
% Ilham Nur Romdoni, M0520038

function P = PolinomialNewton(D,n,x,y)
% Sekarang akan dihitung sebuah yy=f(xx) dengan rumus polinomial Newton tersebut.
xx = 0:1:5;
yy = D(1,1);
for k=2:n
    yy = yy+D(k,k).*plinom(xx,x,k);
end
yy
plot(x,y,'-wo', 'LineWidth',2, 'MarkerEdgeColor','k', 'MarkerFaceColor',[.49 1 .63], 'MarkerSize',12); hold on;
plot(xx,yy,'-bs','LineWidth',1,'MarkerEdgeColor','k','MarkerFaceColor',[.1 .1 .5],'MarkerSize',6); grid on;
end
```

## 2.4 Analisis

Fungsi yang pertama dijalankan adalah fungsi SelisihBagi. Inisialisasi nilai X dan Y untuk mengisi parameter X dan Y pada fungsi SelisihBagi. Nilai X adalah $x_1$, $x_2$, dan seterusnya sesuai yang ditentukan dengan bentuk matriks. Sedangkan nilai Y adalah nilai fungsi f(x). Lalu buat variabel D untuk menyimpan hasil dari fungsi SelisihBagi sehingga dapat dipanggil pada fungsi PolinomialNewton. Maka akan menampilkan 3 bentuk matriks. Matriks D pertama adalah matriks persegi dengan semua bilangan adalah 0. Matriks D kedua adalah matriks yang kolom ke-1 berisi nilai dari Y. Sedangkan D terakhir adalah hasil dari selisih bagi.

Fungsi selanjutnya adalah PolinomialNewton. *Input*-an parameter untuk pemanggilan fungsinya adalah D yang merupakan hasil akhir fungsi SelisihBagi, n yang merupakan Panjang x, lalu variabel X dan Y yang sudah diinisialisasikan. Lalu akan ditampilkan sebuah grafik penyelesaian. Grafik yang ditampilkan untuk Polinomial Newton adalah garis lurus yang dihubungkan oleh titik. Grafik tidak menampilkan garis yang melengkung. Polinomial P(x) yang ditentukan sudah benar atau sesuai dengan fungsi f(x). Hal ini ditunjukkan di mana *output* dari f(x) yang berupa titik lingkaran selalu dilewati garis dari P(x). Untuk lebih jelasnya, dapat dilihat pada *screenshot output* dari kedua permasalahan yang diberikan pada lampiran. Berikut penyelesaian dari kedua permasalahan di atas.

### 2.4.1 Praktikum

Polinomial Newton $P_3$(x) dari $f(x) = x^3 - 4x$ dengan $x_1$=1, $x_2$=2, $x_3$=3 dan $x_4$=4 adalah

$$P_3(x) = -3x + 15x^3 + 48x^4 + 105x^5$$

### 2.4.2 *Posttest*

P(x) dari fungsi $f(x) = \cos(x)$ dengan menggunakan 5 titik yaitu $x_1 = 0, x_2 = 1, x_3 = 2, x_4 = 3$, dan $x_5 = 4$ adalah

$$P(x) = 1 + 0.5403x - 0.4161x^2 - 0.99x^3 - 0.6536x^4 + 0.7687x^5$$

# BAB III

# PENUTUP

## 3.1 Kesimpulan

Setelah melakukan praktikum penjadwalan CPU, dapat disimpulkan:

1. Untuk menentukan P(x) dari sebuah fungsi dengan teknik interpolasi Polinomial Newton dilakukan dengan *devide difference* yang akan menghasilkan matriks penyelesaian lalu diekseskusi dengan formula rumus Polinimial Newton.
2. Grafik fungsi dari perbandingan f(x) dan P(x) jika dihitung dengan algoritma yang tepat akan menampilkan titik penyelesaian yang sama persis.

## 3.2 Daftar Pustaka


[1] JH. Mathews, KK. Fink, *Numerical Methods Using Matlab,* Prentice Hall, 2004

[2] Hidayat, Risanuri. *Interpolasi*. Jurusan Teknik Elektro dan Teknologi Informasi, FT UGM

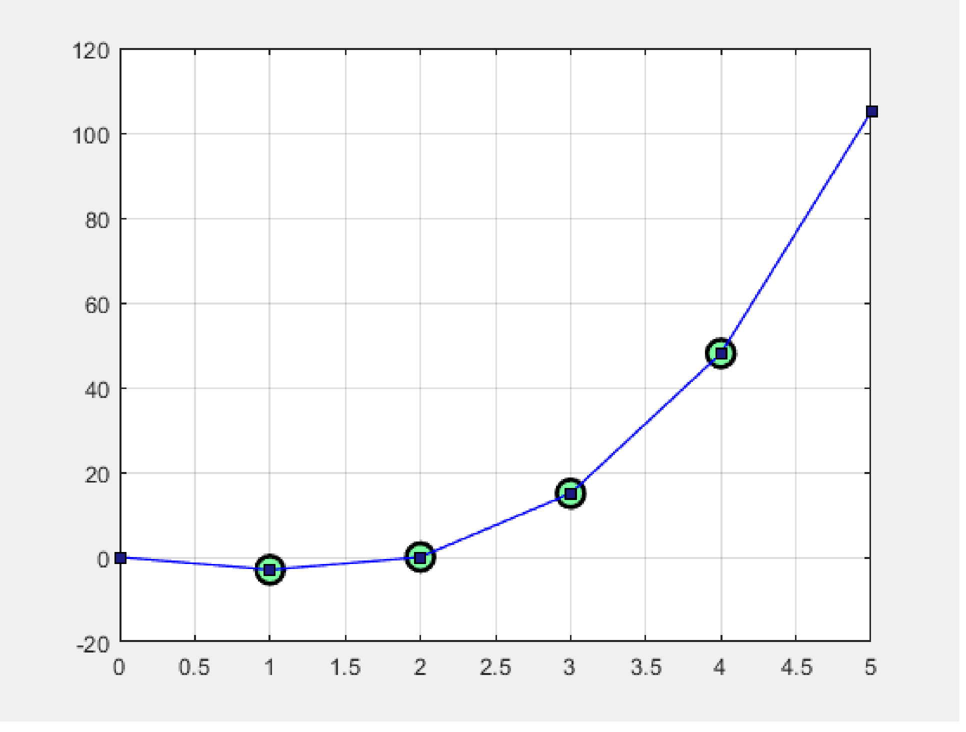
[3] Al-Azizi, Aditya Aulia. (2021). *Praktikum 10 – Interpolasi.* Diakses pada 26 November 2021, dari https://classroom.google.com/u/1/c/NzIzOTUzODA5ODFa/a/NDM3NDc2MTg3Nzk0/details

# LAMPIRAN

## Praktikum

Berikut adalah *output* dari program MATLAB untuk contoh soal praktikum.

```
>> X = [1 2 3 4];
>> Y = X.^3 - 4*X;
>> D = SelisihBagi(X,Y)

D =

     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0


D =

    -3     0     0     0
     0     0     0     0
    15     0     0     0
    48     0     0     0


D =

    -3     0     0     0
     0     3     0     0
    15    15     6     0
    48    33     9     1

>> PolinomialNewton(D,4,X,Y)

yy =

     0    -3     0    15    48   105
```



## *Posttest*

Berikut adalah *output* dari program MATLAB untuk kasus tugas *posttest*.

```
>> X = [0 1 2 3 4];
>> Y = cos(X);
>> D = SelisihBagi(X,Y)

D =

     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0


D =

    1.0000         0         0         0         0
    0.5403         0         0         0         0
   -0.4161         0         0         0         0
   -0.9900         0         0         0         0
   -0.6536         0         0         0         0


D =

    1.0000         0         0         0         0
    0.5403   -0.4597         0         0         0
   -0.4161   -0.9564   -0.2484         0         0
   -0.9900   -0.5738    0.1913    0.1466         0
   -0.6536    0.3363    0.4551    0.0879   -0.0147

>> PolinomialNewton(D,5,X,Y)

yy =

    1.0000    0.5403   -0.4161   -0.9900   -0.6536    0.7687
```