

Nama : Annisa Octavyanti Hakim

NIM : F1D019008

PRAKTIKUM MANDIRI III METODE NUMERIK

A. TUJUAN

1. Untuk mengetahui prinsip kerja metode Pencocokan Kurva dan Interpolasi.
2. Untuk mengetahui teknik Pencocokan Kurva dan Interpolasi.
3. Untuk mengetahui prinsip penukaran baris dan kolom dalam pengaruh banyaknya data yang dipertimbangkan dalam interpolasi.

B. PERMASALAHAN

Diberikan pasangan data (x,y)

X	1	2	3	4	5	6	7	8
Y	1.5577	1.2131	0.9447	0.7358	0.5730	0.4462	0.3476	0.2706

1. Tentukan fungsi terbalik yang mewakili data tersebut dengan model Beda Terbagi Newton:
 - a) $y=a_0+a_1x$
 - b) $y=a_0x^2+a_1x+a_2$
 - c) $y=ax^b$
2. Tentukan nilai $x=4.5$ dengan memmpertimbangkan setengah data dan semua data menggunakan metode Beda Terbagi Newton.

C. SOURCE CODE

1. *Source code* permasalahan pendekatan terbalik.

```
package baginewton;
import java.text.DecimalFormat;
public class BagiNewton {
    static DecimalFormat decfor = new DecimalFormat("#.#####");
    private static final double EPSILON = 1e-10;
    static void view(double[] arr) {
        decfor.setMinimumFractionDigits(1);
        for(int i=0;i<arr.length;i++) {
            System.out.format("%11s", decfor.format(arr[i]));
            System.out.println();
        }
    }
    static void view(double[][] arr) {
        decfor.setMinimumFractionDigits(1);
        for(int i=0;i<arr.length;i++) {
            for(int j=0;j<arr[i].length;j++)
                System.out.format("%11s", decfor.format(arr[i][j]));
            System.out.println();
        }
    }
    public static double[] gaussianElimination(double[][] A1, double[] b) {
        int n = b.length;
        for (int x = 0; x < n; x++) {
            int max = x;
            for (int i = x + 1; i < n; i++)
                if (Math.abs(A1[i][x]) > Math.abs(A1[max][x]))
                    max = i;
            double[] temp = A1[x]; A1[x] = A1[max]; A1[max] = temp;
            double t = b[x]; b[x] = b[max]; b[max] = t;
            if (Math.abs(A1[x][x]) <= EPSILON)
```

```

throw new ArithmeticException("Matrix is singular or nearly singular");
    for (int i = x + 1; i < n; i++) {
        double alpha = A1[i][x] / A1[x][x];
        b[i] -= alpha * b[x];
        for (int j = x; j < n; j++)
            A1[i][j] -= alpha * A1[x][j];
    }
    double[] x = new double[n];
    for (int i = n - 1; i >= 0; i--) {
        double sum = 0.0;
        for (int j = i + 1; j < n; j++)
            sum += A1[i][j] * x[j];
        x[i] = (b[i] - sum) / A1[i][i];
    }
    return x;
}

public static double[] pencocokanKurva(double[][] soal, int N) {
    double n = soal.length;
    double[][] A2 = new double[N][N];
    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
            A2[i][j] = 0;
    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
            for (int k = 0; k < n; k++)
                A2[i][j] = A2[i][j] + Math.pow(soal[k][0], i + j);
    A2[0][0] = n;
    System.out.println("Matriks A:");
    view(A2);
    double[] B2 = new double[N];
    for (int i = 0; i < N; i++)
        B2[i] = 0;
    for (int i = 0; i < N; i++)
        for (int k = 0; k < n; k++)
            B2[i] += Math.pow(soal[k][0], i) * soal[k][1];
    System.out.println("Matriks B:");
    view(B2);
    return gaussianElimination(A2, B2);
}

public static void main(String[] args) {
    double[][] dataTabel = {
        {1, 1.5577}, {2, 1.2131}, {3, 0.9447},
        {4, 0.7358}, {5, 0.5730}, {6, 0.4462},
        {7, 0.3476}, {8, 0.2706}};
    System.out.println("Matriks soal:");
    view(dataTabel);
    double[] hasil;
    double err = 0;
    System.out.println("\nPenyelesaian a0+a1x:");
    hasil = pencocokanKurva(dataTabel, 2);
    System.out.format("Persamaan:          (%s)          +          (%s)x",
        decfor.format(hasil[0]), decfor.format(hasil[1]));
    for (int i = 0; i < dataTabel.length; i++)
        err += Math.pow((hasil[0] + hasil[1] * dataTabel[i][0]) -
            dataTabel[i][1], 2);
    err = Math.sqrt(err/dataTabel.length);
    System.out.println("\nDengan error: " + err);
    System.out.println("\nPenyelesaian a0x^2+a1x+a2:");
    hasil = pencocokanKurva(dataTabel, 3);
    System.out.format("Persamaan: (%s)x^2+(%s)x+(%s)",
        decfor.format(hasil[2]), decfor.format(hasil[1]), decfor.format(hasil[0]));
    err = 0;
    for (int i = 0; i < dataTabel.length; i++)
        err += Math.pow((hasil[2] * Math.pow(dataTabel[i][0], 2) + hasil[1] *
            dataTabel[i][0] + hasil[0]) - dataTabel[i][1], 2);
    err = Math.sqrt(err/dataTabel.length);
}

```

```

        System.out.println("\nDengan error: " + err);
        System.out.println("\nPenyelesaian ab^x:");
        double[][] soalLog = new double[dataTabel.length][dataTabel[0].length];
        for(int i=0;i<soalLog.length;i++)
            for(int j=0;j<soalLog[0].length;j++)
                soalLog[i][j] = Math.log(dataTabel[i][j]);
        hasil = pencocokanKurva(soalLog, 2);
        System.out.format("Persamaan:      (%s)x^(%s)",      decfor.format(hasil[0]),
        decfor.format(hasil[1])); err = 0;
        for(int i=0;i<dataTabel.length;i++)
            err += Math.pow((hasil[0] * Math.pow(dataTabel[i][0], hasil[1])) -
            dataTabel[i][1], 2); err = Math.sqrt(err/dataTabel.length);
        System.out.println("\nDengan error: " + err);}

```

2. Source Code permasalahan Beda Terbagi Newton

```

package baginewton2;
import java.lang.Math;
class proses {
    double penyelesaian(double e,double[] x,double[] y, int n, int
    ordoMatriks){
        ordoMatriks+=1;
        double[][] newton = new double[n][n];
        for (int i=0;i<ordoMatriks;i++)
            newton[0][i]=y[i];
        for (int i=1;i<ordoMatriks;i++){
            for (int j=0;j<ordoMatriks-i;j++)
                newton[i][j]=(newton[i-1][j+1]-newton[i-1][j])/(x[j+i]-
                x[j]);}
        double N=newton[0][0];
        for(int i=1;i<ordoMatriks;i++){
            double banding=newton[i][0];
            for(int j=0;j<=i-1;j++)
                banding=banding*(e-x[j]);
            N=N+banding;}
        return N;} }
public class BagiNewton2{
    public static void main(String[] args) {
        proses bagi = new proses();
        double[] x ={1,2,3,4,5,6,7,8};
        double[] y ={1.5576,1.2131,0.9447,0.7358,0.5730,0.4462,0.3476,0.2707};
        System.out.println("Metode Beda Terbagi Newton dengan F(x=4.5)");
        System.out.println("|      x      |      y      |");
        for(int i=0;i<x.length;i++)
            System.out.println("|    "+x[i] + "    |    "+ y[i] + "    |");
        double nilai=bagi.penyelesaian(4.5,x,y,8,7);
        System.out.print("F(x=4.5) Ordo 7 = "+nilai);
        System.out.println();    } }

```

D. HASIL UJI COBA DAN ANALISA SINGKAT

Permasalahan 1, dilakukan perhitungan untuk proses kurva interpolasi terhadap $y=a_0+a_1x$, $y=a_0x^2+a_1x+a_2$, $y=ax^b$. Pada proses pertama, nilai-nilai f_x dimasukkan ke dalam sebuah array 2 dimensi, dimana kolom pertama dari array tersebut bernilai matriks f_x . Kemudian dilakukan perhitungan dengan acara yaitu, pencocokan kurva dan eliminasi gauss sehingga didapat nilai pendekatan dan nilai error dari nilai y yang dicari.

Untuk permasalahan kedua, pada percobaan interpolasi menggunakan BTN (beda terbagi newton) terhadap data $x=\{1, 2, 3, 4, 5, 6, 7, 8\}$ dan $f_x = \{1.5577, 1.2131, 0.9447, 0.7358, 0.5730, 0.4462, 0.3476, 0.2706\}$ di titik $x=4.5$. Pada percobaan ini digunakan *method* “BTN”

yang telah dibuat sebelumnya menerapkan konsep *array* yang menyimpan data untuk setiap ordo yang digunakan. Kemudian untuk melengkapi nilai-nilai dari *array*, pada saat perulangan elemen tabel dengan indeks ditambah 1 akan dikurangi dengan elemen *array* dengan indeks yang dikunjungi, kemudian pada setiap perulangannya matriks “x” akan dikurangi dengan elemen ditambah 1 dikurangi dengan matriks “x” dengan indeks yang dikunjungi pada saat perulangan. Selanjutnya, dilakukan penjumlahan dengan menggunakan perulangan pada semua nilai pada *array*. Pada semua nilai yang melewati perulangan akan dikalikan x dan dikurangi matriks “x”. Proses akan terus menerus dilakukan hingga elemen-elemen pada “a” telah diproses dan akhirnya akan diperoleh persamaan ordo 7 = 0.6493370605468751.

Hasil *running*:

```

Matriks soal:
  1.0  1.5577
  2.0  1.2131
  3.0  0.9447
  4.0  0.7358
  5.0  0.573
  6.0  0.4462
  7.0  0.3476
  8.0  0.2706

Penyelesaian a0+ax:
Matriks A:
  8.0  36.0
 36.0 204.0
Matriks B:
 6.0887 19.9014
Persamaan: (1.5644179) + (-0.1785179)x
Dengan error: 0.10083510913925967

Penyelesaian a0x^2+ax+a2:
Matriks A:
  8.0  36.0 204.0
 36.0 204.0 1296.0
 204.0 1296.0 8772.0
Matriks B:
 6.0887 19.9014 91.4242
Persamaan: (0.0216792)x^2+(-0.3736304)x+(1.8896054)
Dengan error: 0.01541416734888795

Penyelesaian ab^x:
Matriks A:
  8.0 10.6046029
 10.6046029 17.5205499
Matriks B:
-3.4549694 -7.4703975
Persamaan: (0.6744784)x^(-0.8346184)
Dengan error: 0.5613453345309329
BUILD SUCCESSFUL (total time: 0 seconds)

```

Gambar 3.1 *Running* program Fitting interpolasi

```

Metode Beda Terbagi Newton dengan F(x=4.5)
|  x  |  y  |
| 1.0 | 1.5576 |
| 2.0 | 1.2131 |
| 3.0 | 0.9447 |
| 4.0 | 0.7358 |
| 5.0 | 0.573 |
| 6.0 | 0.4462 |
| 7.0 | 0.3476 |
| 8.0 | 0.2707 |
F(x=4.5) Ordo 7 = 0.6493370605468751
BUILD SUCCESSFUL (total time: 0 seconds)

```

Gambar 3.2 *Running* program BTN di titik x=4.5

E. KESIMPULAN

1. Prinsip kerja metode fitting kurva, ialah menggunakan seminimal mungkin parameter bebas dan penyimpanan titik data dibuat minimum. Sedangkan pada metode interpolasi, prinsip

kerja yang digunakan ialah menggunakan fungsi polinomial untuk menginterpolasikan $f(x)$ pada suatu titik.

2. Pencocokan kurva merupakan suatu teknik yang penting dan sangat diperlukan untuk *menghandle* data hasil pengukuran suatu variabel, sehingga diperoleh gambaran yang jelas mengenai sifat-sifat atau perilaku variabel yang diukur. Untuk menentukan teknik pencocokan kurva terhadap data yang diberikan dilakukan dengan membandingkan hasil error pada setiap data yang dihitung.
3. Pada metode Beda Bagi Newton (BTN), prinsip penukaran baris dan kolom digunakan untuk menentukan nilai fungsi dari suatu titik dapat dilakukan berdasarkan data-data yang berada pada sekitarnya, semakin besar orde yang digunakan maka semakin banyak data yang digunakan untuk perhitungan dan nilai interpolasi yang akan didapat akan semakin mendekati nilai sebenarnya, semakin kecil *range* antara titik maka hasil yang didapatkan semakin baik. Secara analitik, semakin tinggi derajat polinom yang digunakan, semakin akurat hasil yang diperoleh.

F. REFERENSI

1. Wijaya, IGP Suta. 2018. *Pencocokan Kurva*. Mataram: Universitas Mataram.
2. Wijaya, IGP Suta. 2018. *Interpolasi*. Mataram: Universitas Mataram.
3. Subakti, Irfan. "Metode Numerik". Surabaya: Institut Teknologi Sepuluh Nopember, 2006.
4. Chapra, Renaldi, "Metode Numerik Untuk Teknik", Jakarta: Universitas Indonesia (UI), 2007.
5. Luknanto, Djoko, "Metoda Numerik", Universitas Gadjah Mada, Yogyakarta, 2001.