

# Modul Praktikum Metode Numerik



I Made Suarsana, S.Pd. M.Si.

SINGARAJA

**2020**



# DAFTAR ISI

<b>DAFTAR ISI .....</b>	<b>I</b>
<b>CONTOH FORMAT LAPORAN PRAKTIKUM.....</b>	<b>II</b>
<b>PENDEKATAN AKAR PERSAMAAN TAK LINIER.....</b>	<b>1</b>
1.1. TUJUAN PEMBELAJARAN .....	1
1.2. DASAR TEORI.....	1
1.2.1. <i>Metode Bagi Dua</i> .....	1
1.2.2. <i>Metode Iterasi Titik Tetap</i> .....	6
1.2.3. <i>Metode Posisi Palsu</i> .....	10
1.2.4. <i>Metode Newton-Raphson</i> .....	13
1.2.5. <i>Metode Tali Busur</i> .....	18
1.3. PERCOBAAN.....	20
<b>SISTEM PERSAMAAN LINIER .....</b>	<b>21</b>
2.1. TUJUAN PEMBELAJARAN .....	21
2.2. DASAR TEORI.....	21
2.2.1. <i>Eliminasi Gauss Sederhana</i> .....	21
2.2.2. <i>Eliminasi Gauss dengan Pivoting Parsial</i> .....	24
2.2.3. <i>Invers Matriks</i> .....	25
2.2.4. <i>Dekomposisi Segitiga</i> .....	28
2.2.5. <i>Iterasi Jacobi</i> .....	30
2.2.6. <i>Iterasi Gauss-Seidel</i> .....	34
2.2.7. <i>SPL Tridiagonal</i> .....	37
2.3. PERCOBAAN.....	39
<b>PENDEKATAN INTEGRAL DAN TURUNAN.....</b>	<b>40</b>
3.1. TUJUAN PEMBELAJARAN .....	40
3.2. DASAR TEORI.....	40
3.2.1. <i>Metode Trapesium</i> .....	40
3.2.2. <i>Metode Simpson 1/3</i> .....	44
3.2.3. <i>Pendekatan Turunan</i> .....	47
3.3. PERCOBAAN.....	50
<b>INTERPOLASI.....</b>	<b>52</b>
4.1. TUJUAN PEMBELAJARAN .....	52
4.2. DASAR TEORI.....	52
4.2.1. <i>Interpolasi Polinom Lagrange</i> .....	52
4.2.2. <i>Interpolasi Beda Terbagi Newton</i> .....	57
4.3. PERCOBAAN.....	60
<b>REGRESI .....</b>	<b>61</b>
5.1. TUJUAN PEMBELAJARAN .....	61
5.2. DASAR TEORI.....	61
5.2.1. <i>Regresi</i> .....	61
5.2.2. <i>Regresi Linier</i> .....	62
5.2.3. <i>Regresi Ekspensial</i> .....	66
5.2.4. <i>Regresi Polinomial</i> .....	68
5.3. PERCOBAAN.....	71

## Contoh Format Laporan Praktikum

Judul Percobaan : **Metode Bagi Dua**  
 Nama : .....  
 NIM : .....

### A. Algoritma

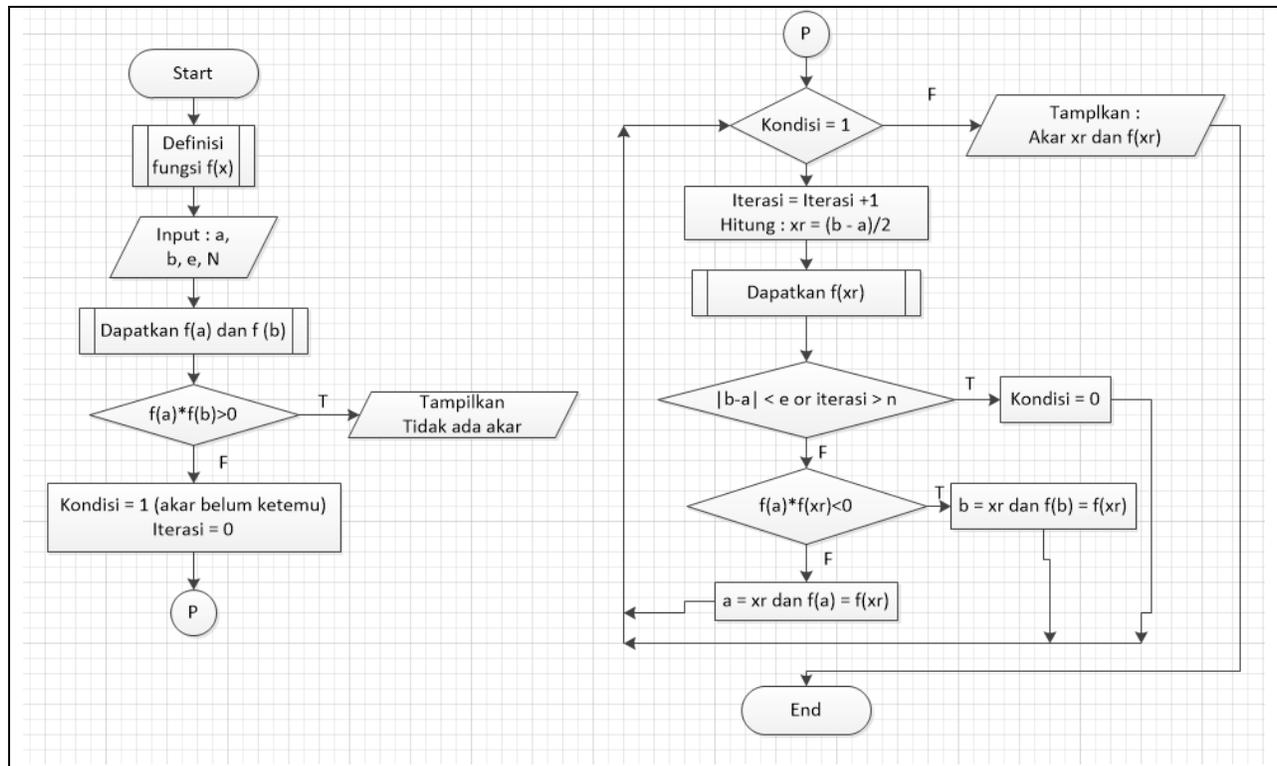
Input:  $f(x)$ ,  $a$ ,  $b$ ,  $tol$

Output; akar

Langkah-langkah.

- 1)  $T1 = (a + b)/2$
- 2) Jika  $f(a).f(T) < 0$  maka  $b = T$ . Jika tidak  $a = T$
- 3) Jika  $abs(b-a) < tol$  maka akar =  $T$ . Selesai
- 4) Kembali ke langkah 1)

### B. Flow Chart (Diagram Alir) dengan Ms. Visio



### C. Listing Program yang Sudah Benar

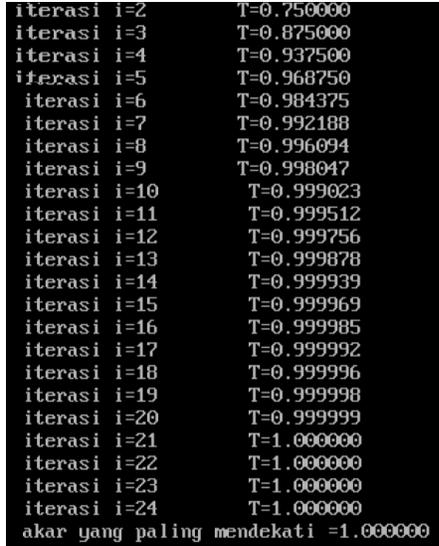
```

- Syntax:
program bagi_dua;
uses crt;
var a,b,T,tol:real;
i:integer;
label 5;
function f(x:real):real;
begin
  f:=(x-1)*(x-3);
end;
begin
  clrscr;
  i:=1;
  writeln('masukkan a=');readln(a);
  writeln('masukkan b=');readln(b);
  tol:=0.0000001;
  Repeat
  T:=(a+b)/2;
  writeln('iterasi i=',i,'      T=',T:2:6);
  if f(a)*f(T)<0 then b:=T;
  if f(a)*f(T)=0 then
    begin
      writeln('akar=',T:2:6);
      goto 5;
    end;
  If f(a)*f(T)>0 then a:=T;
  i:=i+1
  until abs(b-a)<tol;
  writeln('akar yang paling mendekati =',T:2:6);readln;
5:end.

```

### D. Hasil Percobaan

- Print Screen



```

iterasi i=2      T=0.750000
iterasi i=3      T=0.875000
iterasi i=4      T=0.937500
iterasi i=5      T=0.968750
iterasi i=6      T=0.984375
iterasi i=7      T=0.992188
iterasi i=8      T=0.996094
iterasi i=9      T=0.998047
iterasi i=10     T=0.999023
iterasi i=11     T=0.999512
iterasi i=12     T=0.999756
iterasi i=13     T=0.999878
iterasi i=14     T=0.999939
iterasi i=15     T=0.999969
iterasi i=16     T=0.999985
iterasi i=17     T=0.999992
iterasi i=18     T=0.999996
iterasi i=19     T=0.999998
iterasi i=20     T=0.999999
iterasi i=21     T=1.000000
iterasi i=22     T=1.000000
iterasi i=23     T=1.000000
iterasi i=24     T=1.000000
akar yang paling mendekati =1.000000

```

- Ulasan: Pada percobaan diatas soalnya adalah  $f(x) = (x - 1) * (x - 3)$  Kemudian hasil yang didapat akar = 1 pada iterasi ke 24 dengan toleransi 0.0000001



## PRAKTIKUM 1

### PENDEKATAN AKAR PERSAMAAN TAK LINIER

#### 1.1. TUJUAN PEMBELAJARAN

Mahasiswa dapat memahami dan menerapkan:

- i. metode bagi dua
- ii. metode posisi palsu
- iii. metode newton-raphson
- iv. metode tali busur

untuk menentukan pendekatan akar persamaan tak linier.

#### 1.2. DASAR TEORI

##### 1.2.1. Metode Bagi Dua

Dalam matematika terapan seringkali harus mencari selesaian persamaan yang berbentuk  $f(x) = 0$  yakni bilangan  $x_0$  sehingga  $f(x_0) = 0$ . Dalam hal ini  $f$  adalah persamaan/fungsi tak linier. Nilai-nilai  $x$  yang memenuhi persamaan tersebut disebut dengan akar atau titik nol dari persamaan.

Persamaan  $f(x)$  dapat berbentuk sebagai berikut :

##### 1. Persamaan Aljabar

Contoh : persamaan polinom berordo lebih dari 2.

$$a_n x^n + a_{(n-1)} x^{(n-1)} + \dots + a_2 x^2 + a_1 x^1 + a_0 = 0 \text{ dengan } a_n \neq 0, n > 2$$

##### 2. Persamaan Transenden

Persamaan yang memuat fungsi-fungsi trigonometri, logaritma atau eksponen.

Contoh :

$$e^{-x} + \sin x = 0$$

##### 3. Persamaan campuran : memuat baik persamaan polinom maupun persamaan transenden.

Contoh :

$$x^2 \sin x + 3 = 0$$

$$x^2 + \ln x = 0$$

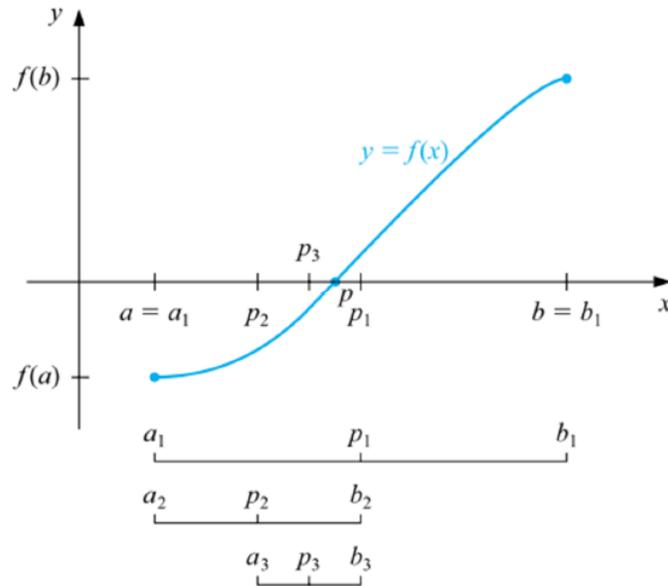
Dari contoh-contoh di atas terlihat bahwa rumus-rumus yang memberikan nilai eksak dari penyelesaian eksplisit hanya akan ada untuk kasus-kasus yang sederhana. Dalam banyak kasus, harus menggunakan metode hampiran, khususnya pada kasus dimana secara aljabar solusinya sulit ditemukan.

Metode iterasi numeris adalah metode dimana kita memilih sebarang  $x_0$  sebagai tebakan awal dan secara beruntun menghitung barisan  $x_0, x_1, x_2, \dots$  secara rekursif dari relasi berbentuk  $x_{n+1} = g(x_n), n = 0, 1, 2, 3, \dots$

Dengan  $g$  didefinisikan dalam selang yang memuat  $x_0$  dan rentang  $g$  terletak dalam selang tersebut. Metode yang demikian khususnya cocok untuk komputer karena metode tersebut melibatkan pengulangan satu proses komputasi  $g(x)$ .

Metode ini adalah metode untuk menentukan titik nol (akar) dari  $f$  bila  $f$  kontinu di suatu selang. Metode ini sangat sederhana tetapi kekonvergenannya lambat. Metode bagi dua didasarkan pada teorema nilai antara untuk fungsi kontinu, yaitu suatu selang  $[a, b]$  harus memuat suatu titik nol bila  $f(a)$  dan  $f(b)$  berlawanan tanda, misalnya  $f(a) < 0$  dan  $f(b) > 0$ . Hal ini menyarankan metode pengulangan pembagiduaan selang dan dalam setiap langkah mengambil setengah selang yang juga memenuhi persyaratan  $f(a).f(b) < 0$ .

Metode bagi dua memerlukan dua nilai sebagai tebakan awal. Sebut  $a$  dan  $b$  dengan  $a < b$  dan harus memenuhi syarat  $f(a).f(b) < 0$ . Selang  $[a, b]$  memuat satu akar. Mula-mula ditentukan titik tengah selang  $[a, b]$ , misal titik  $T$  dengan  $T = \frac{a+b}{2}$ . Dua selang baru diperoleh, yakni  $[a, T]$  dan  $[T, b]$ . Salah satu dari selang ini diantaranya pasti memuat akar. Berikutnya yang ditinjau adalah selang yang memuat akar tersebut. Proses diulangi dengan membagi dua selang tersebut dan memeriksa setengah selang yang memuat akar. Pembagi-duaan selang ini dilanjutkan sampai lebar selang yang ditinjau cukup kecil.



Gambar 1. Metode Bagi Dua

Dari gambar di atas titik  $p$  merupakan akar dari  $f(x)$ . Titik  $p_1$  merupakan titik tengah selang  $[a, b]$ . Oleh karena  $f(a) \cdot f(p_1) < 0$ , maka akar  $f(x)$  terletak di selang  $[a, p_1]$ . Titik  $p_2$  merupakan titik tengah  $[a, p_1]$ . Oleh karena  $f(p_2) \cdot f(p_1) < 0$ , maka selang  $[p_2, p_1]$  memuat akar  $f(x)$ . Proses ini berlangsung secara terus menerus dan berhenti apabila mencapai eror yang telah ditentukan.

Dari uraian di atas, penentuan setengah selang yang memuat akar dilakukan dengan memeriksa tanda dari hasil kali  $f(a) \cdot f(T)$  atau  $f(b) \cdot f(T)$ .

$$f(a) \cdot f(T) \begin{cases} < 0, \text{ berarti akar pada } (a, T) \\ = 0, \text{ berarti akar : } T \\ > 0, \text{ berarti akar pada } (T, b) \end{cases}$$

Dalam algoritma digunakan variabel-variabel sebagai berikut:

$a$  sebagai ujung kiri selang

$b$  sebagai ujung kanan selang

$T$  sebagai titik tengah

Karena metode ini selalu menghasilkan akar, maka dikatakan bahwa metode ini selalu konvergen. Besarnya epsilon tergantung pada ketelitian yang diinginkan. Semakin kecil

epsilon semakin teliti hampiran akan yang diperoleh. Bila proses dilakukan sebanyak  $n$  iterasi, maka toleransi eror (Tol) yang diberikan adalah

$$\frac{b-a}{2^n} < Tol$$

### Contoh 1.

Gunakan metode bagi dua untuk menentukan akar dari  $f(x) = e^{-x} - x$ .

Karena interval awal dapat dipilih dari  $x_l = 0$  hingga  $x_u = 1$ .

Dengan sendirinya, taksiran awal akar terletak di tengah interval tersebut :

$$x_r = \frac{0+1}{1} = 0,5$$

Taksiran ini menunjukkan kesalahan dari (harga sebenarnya adalah 0,56714329...):

$$E_t = 0,56714329 - 0,5 = 0,06714329$$

atau dalam bentuk relatif :

$$|\varepsilon_t| = \left| \frac{0,06714329}{0,56714329} \right| \times 100\% = 11,8\%$$

dimana indeks  $t$  menunjukkan bahwa kesalahan diacu terhadap harga sebenarnya.

Sekarang kita hitung :

$$f(0).f(0,5) = 1.(0,10653) = 0,10653 > 0$$

Tidak ada perubahan tanda terjadi antara  $x_l$  dan  $x_r$ . akar selanjutnya terletak pada interval antara  $x = 0,5$  dan  $1,0$  ( $x_l = 0,5$ )

$$x_r = \frac{0,5+1,0}{2} = 0,75 \text{ dan } \varepsilon_t = 32,2\%$$

Proses dapat dilanjutkan lagi agar mendapatkan taksiran yang lebih halus. misalnya untuk iterasi ketiga adalah :

$$f(0).f(0,5) = -0,030 < 0$$

Karena akar terletak di antara 0,5 dan 0,75 ( $x_u = 0,75$ ) :

$$x_r = \frac{0,5+0,75}{2} = 0,625 \text{ dan } \varepsilon_t = 10,2\%$$

Dan iterasi keempat adalah :

$$f(0,5).f(0,625) = -0,010 < 0$$

Karena akar terletak di antara 0,5 dan 0,625 ( $x_u = 0,625$ ) :

$$x_r = \frac{0,5 + 0,625}{2} = 0,5625 \text{ dan } \varepsilon_t = 0,819\%$$

sampai seterusnya diulangi untuk memperoleh taksiran yang lebih halus.

Contoh 2.

Carilah akar dari  $x^3 + 4x^2 - 10 = 0$  pada interval  $[1,2]$ .

**Penyelesaian :**

Dalam penyelesaian ini saya akan menggunakan sampai iterasi ke-10 dan menggunakan 5 angka dibelakang koma.

$$f(x) = x^3 + 4x^2 - 10$$

$$f(1) = (1)^3 + 4(1)^2 - 10 = -5$$

$$f(2) = (2)^3 + 4(2)^2 - 10 = 14$$

$$f(1,5) = (1,5)^3 + 4(1,5)^2 - 10 = 2,375$$

$$f(1,25) = (1,25)^3 + 4(1,25)^2 - 10 = -1,79687$$

$$f(1,375) = (1,375)^3 + 4(1,375)^2 - 10 = 0,16210$$

$$f(1,3125) = (1,3125)^3 + 4(1,3125)^2 - 10 = -0,84838$$

$$f(1,34375) = (1,34375)^3 + 4(1,34375)^2 - 10 = -0,35098$$

$$f(1,35938) = (1,35938)^3 + 4(1,35938)^2 - 10 = -0,09632$$

$$f(1,36719) = (1,36719)^3 + 4(1,36719)^2 - 10 = 0,03239$$

$$f(1,36329) = (1,36329)^3 + 4(1,36329)^2 - 10 = -0,03200$$

$$f(1,36524) = (1,36524)^3 + 4(1,36524)^2 - 10 = 0,000016$$

$$f(1,36426) = (1,36426)^3 + 4(1,36426)^2 - 10 = -0,01601$$

$$f(1,36329) = (1,36329)^3 + 4(1,36329)^2 - 10 = -0,00784$$

$n$	$a$	$b$	$c = \frac{(a+b)}{2}$	$f(a)$	$f(b)$	$f(c)$	$f(a)f(c)$	$f(b)f(c)$
1	1	2	1,5	-	+	+	-	+
2	1	1,5	1,25	-	+	-	+	-
3	1,25	1,5	1,375	-	+	+	-	+
4	1,25	1,375	1,3125	-	+	-	+	-
5	1,3125	1,375	1,34375	-	+	-	+	-
6	1,34375	1,375	1,35938	-	+	-	+	-

7	1,35938	1,375	1,36719	-	+	+	-	+
8	1,35938	1,36719	1,36329	-	+	-	+	-
9	1,36329	1,36719	1,36524	-	+	+	-	+
10	1,36329	1,36524	1,36426	-	+	-	+	-

Jadi akar yang diperoleh dari  $f(x) = x^3 + 4x^2 - 10$  menggunakan 10 iterasi adalah 1.36426

### Algoritma Metode Bagi Dua

Input :  $f(x), a, b$  dan epsilon

Output : hampiran akar

Langkah-langkah:

- i.  $T = \frac{a+b}{2}$
- ii. Jika  $f(a).f(T) < 0$ , maka  $b := T$ . Jika tidak  $a := T$
- iii. Jika  $b - a \leq \text{epsilon}$ , maka hampiran akar := T. Selesai
- iv. Ulangi kembali ke langkah 1.

### 1.2.2. Metode Iterasi Titik Tetap

Metode Iterasi Titik tetap dinamakan juga sebagai metode iterasi sederhana atau metode langsung atau metode substitusi beruntun. Kesederhanaan metode ini karena pembentukan prosedur iterasinya yang mudah dibentuk.

Sebuah nilai  $x$  merupakan solusi dari persamaan  $f(x) = 0$  jika ruas kiri sama dengan ruas kanan. Prinsip ini yang dipakai metode Iterasi titik tetap untuk mencari pendekatan akar. Persamaan  $f(x) = 0$  diubah menjadi persamaan  $x = g(x)$ . Akan dicari nilai  $x = x_0$  sedemikian hingga  $x_0 = g(x_0)$ . Hal ini ditempuh dengan memandang  $g(x_0)$  sebagai  $x_1$ ,  $g(x_1)$  sebagai  $x_2$  ... dst sehingga beda antara  $x_i$  dan  $x_{i+1}$  memenuhi toleransi yang ditetapkan. Namun metode ini tidak menjamin akan mendapatkan hasil yang diharapkan. Perlu diperhatikan bahwa fungsi  $f(x)$  yang dicari pendekatan akarnya haruslah konvergen. Kekonvergenan barisan  $x_0, x_1, x_2, \dots$  sangat dipengaruhi oleh pilihan awal dari nilai  $x_0$ .

#### Contoh.

Lakukan proses iterasi untuk menentukan pendekatan akar persamaan  $f(x) = x^2 - 3x + 1 = 0$  dengan batas toleransi = 0,000001.

Secara aljabar akar persamaan tersebut dapat dihitung menggunakan rumus abc yang hasilnya adalah  $x_1 = 2,6180$  dan  $x_2 = 0,3819$

Penyelesaian.

Persamaan  $f(x) = x^2 - 3x + 1 = 0$  dapat ditulis sebagai  $x = g(x) = \frac{(x^2 + 1)}{3}$

Rumus iterasinya adalah  $x_{i+1} = \frac{(x_i^2 + 1)}{3} : i = 0, 1, 2, \dots$

Pilih tebakan awal, misalnya  $x_0 = 1$ , maka :

$$i = 0, \quad x_1 = \frac{(x_0^2 + 1)}{3} = \frac{((1)^2 + 1)}{3} = 0,667 \text{ (hasil pembulatan)}$$

$$i = 1, \quad x_2 = \frac{(x_1^2 + 1)}{3} = \frac{((0,667)^2 + 1)}{3} = 0,481 \text{ (hasil pembulatan)}$$

$$i = 2, \quad x_3 = \frac{(x_2^2 + 1)}{3} = \frac{((0,481)^2 + 1)}{3} = 0,411 \text{ (hasil pembulatan)}$$

.....

*Ket : iterasi dilanjutkan sampai mendekati nilai akar.*

Nampaknya barisan  $x_i$  akan konvergen menuju ke akar yang terkecil yaitu 0,3819.

Pilih tebakan awal, misalnya  $x_0 = 3$ , maka:

$$i = 0, \quad x_1 = \frac{(x_0^2 + 1)}{3} = \frac{((3)^2 + 1)}{3} = 3,333 \text{ (hasil pembulatan)}$$

$$i = 1, \quad x_2 = \frac{(x_1^2 + 1)}{3} = \frac{((3,333)^2 + 1)}{3} = 4,037 \text{ (hasil pembulatan)}$$

$$i = 2, \quad x_3 = \frac{(x_2^2 + 1)}{3} = \frac{((4,037)^2 + 1)}{3} = 11,415 \text{ (hasil pembulatan)}$$

.....

*Ket : iterasi dilanjutkan sampai mendekati nilai akar.*

Nampaknya barisan  $x_i$  tidak konvergen menuju ke salah satu akar. Tebakan awal  $x_0 = 3$  mengakibatkan barisan  $x_i$  divergen.

**Kesimpulan :** dapat disimpulkan bahwa tebakan awal pada penggunaan metode iterasi titik tetap sangat mempengaruhi kekonvergenan barisan  $x_i$ , sehingga pendekatan akar

menggunakan metode ini dianggap lambat serta tidak menjamin akan memberikan hasil.

Rumus iterasi dari persamaan di atas dapat juga ditulis:  $x_{i+1} = 3 - \frac{1}{x_i}$

Dengan tebakan awal  $x_0 = 1$ , diperoleh hasil sebagai berikut:

$$i = 0, x_1 = \left(3 - \frac{1}{x_0}\right) = \left(3 - \frac{1}{1}\right) = 2,000$$

$$i = 1, x_2 = \left(3 - \frac{1}{x_1}\right) = \left(3 - \frac{1}{2}\right) = 2,500$$

$$i = 2, x_3 = \left(3 - \frac{1}{x_2}\right) = \left(3 - \frac{1}{2,5}\right) = 2,600$$

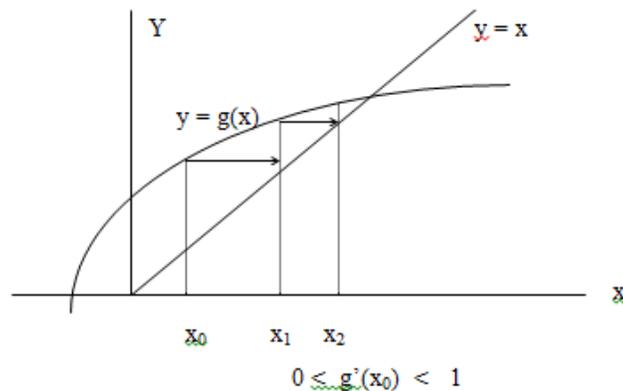
$$i = 3, x_4 = \left(3 - \frac{1}{x_3}\right) = \left(3 - \frac{1}{2,6}\right) = 2,615$$

.....

*Ket : iterasi dilanjutkan sampai mendekati nilai akar.*

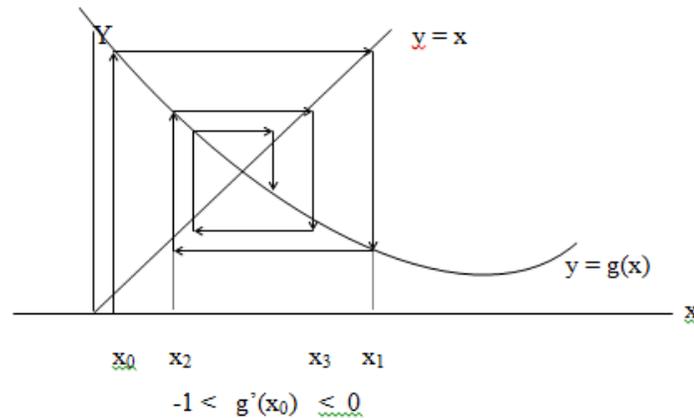
Barisan  $x_i$  nampaknya konvergen ke 2,618. (Coba pilihan tebakan awal yang lain dan bagaimana hasilnya?)

Secara grafik kemungkinan pilihan tebakan awal dan fungsi  $g(x)$  dapat dilihat pada gambar berikut.

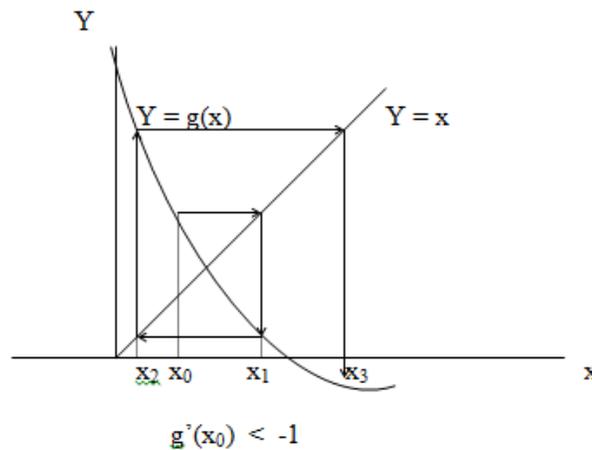


Akar persamaan merupakan titik potong antara  $y = x$  dan  $y = g(x)$ . Tebakan awal  $x_0$  memberikan  $g(x_0) = x_1$ . Dari  $x_0$  ke  $x_1$  arah perpindahan ke kanan menuju ke perpotongan antara  $y = x$  dan  $y = g(x)$ . Ini menunjukkan bahwa barisan yang terjadi

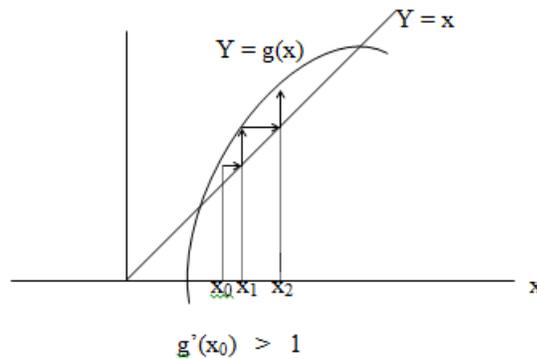
adalah konvergen. Dalam hal ini kemiringan dari  $y = g(x)$  di titik  $x_0$  bernilai antara 0 dan 1.



Mulai dari  $x_0$ , anak panah menunjukkan gerakan yang makin lama mendekati titik potong antara  $y = x$  dan  $y = g(x)$ . Ini berarti barisan  $x_i$  konvergen ke akar persamaan yang dicari.



Pada gambar jelas bahwa anak panah yang menunjukkan hasil iterasi semakin menjauh dari titik potong antara  $y = x$  dan  $y = g(x)$ . Hal ini menyatakan barisan yang dihasilkan oleh iterasi adalah divergen. Perhatikan gradien garis singgung di titik tabakan awal  $g'(x_0) < -1$ .



Seperti pada kasus sebelumnya, gerak panah semakin jauh dari titik potong antara  $y = x$  dan  $y = g(x)$ . Dalam hal ini  $g'(x_0) > 1$ , dan barisan hasil iterasi divergen.

### Algoritma

Input :  $g(x), x_0, tol, n$

Output : akar

Proses :

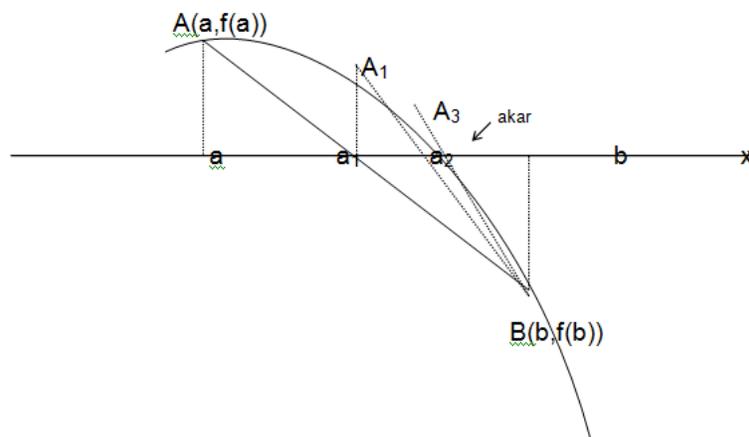
1. Iterasi = 1
2.  $x_1 = g(x_0)$
3. Jika  $\left| \frac{x_1 - x_0}{x_1} \right| \leq tol$  maka akar =  $x_1$ . Selesai.
4.  $x_0 = x_1$
5. Iterasi = iterasi + 1
6. Jika iterasi  $< n$  kembali ke langkah 2
7. Proses belum konvergen.
8. Selesai.

### 1.2.3. Metode Posisi Palsu

Metode posisi palsu ini merupakan alternatif perbaikan metode bagi dua yaitu untuk mempercepat kekonvergenan metode bagi dua. Metode posisi palsu mirip dengan metode bagi dua. Kemiripannya terletak dalam hal diperlukan dua harga taksiran awal pada awal pengurangan akar persamaan. Sedangkan, perbedaannya terletak pada proses pencarian pendekatan akar persamaan selanjutnya setelah pendekatan akar

saat ini ditemukan. Prosedur metode posisi palsu ini mulaidengan memilih dua tebakan awal yaitu dan dimana nilai fungsinya pada kedua tebakanawal ini berbeda tanda. Hubungkan kedua titik yaitu dan dengan garis lurus dan tentukan titik perpotongan garis ini dengan sumbu x. Kenyataan bahwa penggantian kurva oleh garis lurus memberikan “posisi palsu” dari akar yang merupakan asal mula dari nama metodeposisi palsu (method of false position) atau dalam bahasa latinnya regula falsi. Metode ini juga disebut dengan metode interpolasi linear.

Metode posisi palsu adalah metode pencarian akar persamaan dengan memanfaatkan kemiringan dan selisih tinggi dari dua titik batas interval yang mengurung akar. Metode ini merupakan salah satu alternatif untuk mempercepat konvergensi. Seperti halnya metode Bagi Dua, metode Posisi Palsu mensyaratkan diketahuinya interval letak akar yaitu  $(a,b)$ . Hampiran akar persamaan memanfaatkan perpotongan garis yang melalui titik  $(a,f(a))$  dan  $(b,f(b))$  dengan sumbu x. Cara ini akan lebih cepat memperoleh hasil yang mendekati akar yang sebenarnya apabila posisi akar terletak dekat dengan ujung interval seperti pada gambar berikut.



Pengujian letak akar tetap diperlukan untuk perhitungan berikutnya.

Perhatikan gambar. Garis AB memotong sumbu x di titik  $a_1$ . Garis  $A_1B$  memotong sumbu x di  $a_2$ . Garis  $A_3B$  memotong di  $a_4$  dst.

Persamaan garis AB adalah:

$$\frac{y - f(a)}{f(b) - f(a)} = \frac{x - a}{b - a}$$

Titik potong garis dengan sumbu x ( $y=0$ ) adalah  $x_0 = a - f(a) \frac{b-a}{f(b)-f(a)}$

Penghentian perhitungan dilakukan apabila titik potong sudah dekat dengan akar yang ditandai dengan jarak antara dua hasil iterasi berurutan sangat dekat memenuhi toleransi yang diberikan.

**Contoh:**

Hitunglah persamaan salah satu akar dari persamaan  $f(x) = x^3 + x^2 - 3x - 3 = 0$  dengan menggunakan metode posisi palsu sampai dengan 3 kali iterasi!

Penyelesaian:

Ambil tebakan awal yaitu

$$x_1 = 1 \text{ dan } x_2 = 2$$

$$\text{Untuk } x_1 = 1 \Rightarrow f(x_1 = 1) = (1)^3 + (1)^2 - (3 \cdot 1) - 3 = -4$$

$$\text{Untuk } x_2 = 2 \Rightarrow f(x_2 = 2) = (2)^3 + (2)^2 - (3 \cdot 2) - 3 = 3$$

Dengan menggunakan rumus  $x_0 = b - f(b) \frac{b-a}{f(b)-f(a)}$

Iterasi 1

$$x_3 = x_2 - f(x_2) \frac{x_2 - x_1}{f(x_2) - f(x_1)}$$

$$x_3 = 2 - 3 \frac{2-1}{3-(-4)} = 1,5714286$$

$$f(x_3) = (1,5714286)^3 + (1,5714286)^2 - (3 \cdot 1,5714286) - 3 = -1,13644315$$

Iterasi 2

$$x_1 = 1,5714286$$

$$\Rightarrow f(x_1 = 1,5714286) = (1,5714286)^3 + (1,5714286)^2 - (3 \cdot 1,5714286) - 3 = -1,13644315 \quad x_2 = 2$$

$$\Rightarrow f(x_2 = 2) = (2)^3 + (2)^2 - (3 \cdot 2) - 3 = 3$$

$$x_3 = 2 - 3 \frac{2-1,5714286}{3-(-1,13644315)} = 1,7054108$$

$$f(x_3) = (1,7054108)^3 + (1,7054108)^2 - (3 \cdot 1,7054108) - 3 = -0,2477451$$

Iterasi 3

$$x_1 = 1,7054108$$

$$\Rightarrow f(x_1 = 1,7054108) = (1,7054108)^3 + (1,7054108)^2 - (3 \cdot 1,7054108) - 3 = -0,2477451$$

$$\Rightarrow f(x_2 = 2) = (2)^3 + (2)^2 - (3 \cdot 2) - 3 = 3$$

$$x_3 = 2 - 3 \frac{2 - 1,7054108}{3 - (-0,2477451)} = 1,7278827$$

$$f(x_3) = (1,7278827)^3 + (1,7278827)^2 - (1,7278827) - 3 = -0,0393396$$

Diperoleh akar 1,7278827 dengan

$$f(x_3) = (1,7278827)^3 + (1,7278827)^2 - (1,7278827) - 3 = -0,0393396$$

### Algoritma Metode Posisi Palsu

**Input.:**  $f(x), a, b, tol$

**Output:** akar

**Proses:**

- 1)  $x_0 \text{ lama} = b - a$  ( untuk tebakan awal saja)
- 2)  $x_0 = b - f(b) \frac{b - a}{f(b) - f(a)}$
- 3) Jika  $\text{abs } x_0 \cdot x_0 \text{ lama} < \text{tol}$  maka akar =  $x_0$ . Selesai
- 4) Jika  $f(a) \cdot f(x_0) < 0$  maka  $b = x_0$ , jika tidak  $a = x_0$
- 5)  $x_0 \text{ lama} = x_0$ , kembali ke langkah 2).

#### 1.2.4. Metode Newton-Raphson

Pada awalnya, metode Newton Raphson ini lebih dikenal dengan nama metode Newton. Pemberian nama tersebut juga didasarkan atas penemu metode ini, yakni Sir Isaac Newton. Seorang penemu serta ilmuwan dan *saintis* ternama yang berasal dari Lincolnshire. Suatu daerah yang terletak di sebelah timur Inggris. Metode ini diperkenalkan melalui tulisannya yang berjudul *De Analysi Peraequationes Numero Terminorum Infinitas* yang ditulis pada tahun 1669 dan kemudian diterbitkan tahun 1711 oleh William Jones. Serta satu karyanya lagi yang berjudul *De Methodis Fluxionum Et Seri-*

*erum Infinatarium* yang ditulis tahun 1671 yang kemudian diterjemahkan dan diterbitkan sebagai *Method of Fluxions* tahun 1736 oleh John Colson.

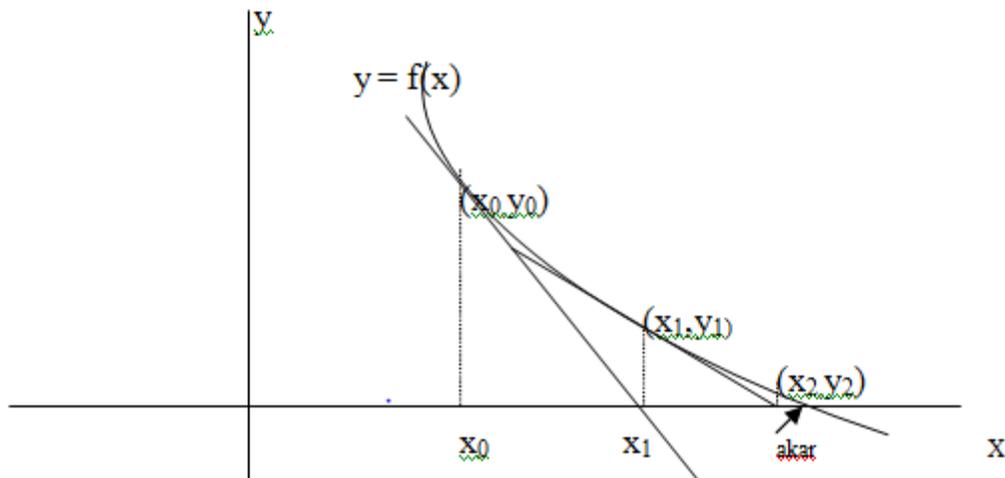
Dalam karyanya tersebut, Newton hanya memandang metode ini untuk fungsi polinomial dan tidak menghitung perkiraan dari  $x_n$ . Metode ini hanya digunakan untuk menghitung barisan dari polinomial dan berakhir pada perkiraan untuk akar  $x$ . Pada akhirnya, Newton hanya menganggap metode ini sebagai metode aljabar murni dan gagal menghubungkannya dengan kalkulus.

Di tahun 1685, seseorang atas nama John Wallis menerbitkan sebuah buku berjudul *A Treatise of Algebra both Historical and Practical* yang kemudian diikuti oleh Joseph Raphson menerbitkan deskripsi yang lebih sederhana dengan judul *Analysis Aequationum Universalis* di tahun 1660. Yang mana dapat mengantarkan Raphson untuk mendapatkan kehormatan sebagai anggota *Royal Society* di tahun 1661. Dalam deskripsinya itu, Raphson juga melihat penggunaan metode Newton murni sebagai metode aljabar dan membatasi kegunaannya pada fungsi polinomial. Namun dia memperkenalkan metode perkiraan suksesif  $x_n$  daripada barisan polinomial yang lebih rumit yang digunakan oleh Newton.

Pada dasarnya buku terbitan Raphson tersebut berfokus terutama pada metode Newton untuk memperkirakan akar dari persamaan yang mana menjelaskan metode yang sama dan contoh-contoh untuk memperkirakan akar persamaan dengan metode Newton mengenai fluks. Dan akhirnya metode ini dinamai sesuai dengan nama dua orang saintis yakni Newton dan Raphson. Sehingga metode ini dikenal dengan nama metode Newton Raphson sampai sekarang.

Metode Newton-Raphson merupakan metode yang didasarkan pada teorema kalkulus yaitu turunan sebuah fungsi. Metode ini dapat digunakan apabila fungsinya mempunyai turunan yang kontinu. Kecepatan metode ini sangat bagus namun ada kemungkinan gagal.

Secara geometri, metode ini memanfaatkan garis singgung pada kurva di sebuah titik. Garis singgung tersebut memotong sumbu  $x$  yang selanjutnya dipakai untuk menentukan garis singgung berikutnya, seperti pada gambar berikut.



Secara sederhana langkah-langkah perhitungan dalam metode ini sebagai berikut:

Dimulai dengan tebakan awal  $x_0$ , dibuat garis singgung pada  $y = f(x)$  di titik  $(x_0, y_0)$  yang memotong sumbu  $x$  di  $x_1$ . Garis singgung kurva di titik  $(x_1, y_1)$  memotong sumbu  $x$  di  $x_2$ , dan seterusnya hingga barisan  $x_i$  yang terbentuk konvergen ke akar yang dicari.

Gradien garis singgung kurva di titik  $(x_0, y_0)$  adalah:

$$f'(x_0) = \frac{f(x_0)}{x_0 - x_1}, \text{ sehingga } x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Berdasarkan bentuk rumus perhitungan untuk  $x_1$  ini maka untuk menghitung  $x_2$  rumusnya adalah :

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

Yang kemudian didapatkan rumus umumnya sebagai berikut :

$$x_{n+1} = x_i - \frac{f(x_n)}{f'(x_n)}$$

Dari rumus di atas jelas bahwa perhitungan dapat dilakukan jika  $f'(x_n) \neq 0$ .

Iterasi Newton Raphson berawal dari sebuah hampiran awal  $r$ , dan kemudian menghitung hampiran selanjutnya dengan cara sebagai berikut.

Misalkan  $x_i$  adalah hampiran awal pada langkah ke- $n$  dengan  $n = 0, 1, 2, \dots$ . Kemudian hitung gradient garis singgung terhadap kurva  $y = f(x)$  di titik  $(x_n, f(x_n))$  yaitu  $f'(x_n)$

dan tentukan persamaan garis singgungnya, yakni  $y = f'(x_n)(x - x_n) + f(x_n)$ . Hampiran berikutnya adalah absis titik potong garis singgung tersebut dengan sumbu- $x$ , yaitu

$$x_{i+1} = x_i - \frac{f(x_n)}{f'(x_n)} \quad (i)$$

Langkah-langkah tersebut diperlihatkan pada gambar

Rumus iterasi (i) juga dapat diturunkan dengan deret Taylor  $f(x)$  disekitar  $x_n$  yaitu:

$$f(x) = f(x_n) + (x - x_n)f'(x_n) + \frac{1}{2}(x - x_n)^2 f''(x_n) + \dots \quad (ii)$$

Dengan mengasumsikan  $x_n$  dan hampiran berikutnya,  $x_n$  cukup dekat ke akar  $r$ , dan mengabaikan suku ke-3 dan seterusnya pada ruas kanan (ii), akan diperoleh (i). Dalam hal ini, fungsi  $f(x)$  telah dihampiri oleh garis singgung di titik  $(x_n, f(x_n))$ . Jadi prinsipnya sama dengan pendekatan geometris sebelumnya.

### Contoh

Tentukan salah satu akar dari persamaan  $f(x) = x^2 - 5x + 6$  dengan metode Newton Raphson.

**Jawab :**

Diketahui :

$f(x) = x^2 - 5x + 6$  maka  $f'(x) = 2x - 5$ . Misalkan nilai  $x_0 = 4$  maka dengan rumus

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

1. Iterasi 1

$$x_0 = 4 \text{ maka } x_{0+1} = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$$x_1 = 4 - \frac{2}{3} = 3,333$$

2. Iterasi 2

$$x_1 = 3,333 \text{ maka } x_{1+1} = x_1 - \frac{f(x_1)}{f'(x_1)}$$

$$x_2 = 3,333 - \frac{0,443889}{1,666} = 3,066560024$$

## 3. Iterasi 3

$$x_2 = 3,066560024 \text{ maka } x_{2+1} = x_2 - \frac{f(x_2)}{f'(x_2)}$$

$$x_3 = 3,066560024 - \frac{0,07099026}{1,133120048} = 3,003909769$$

## 4. Iterasi 4

$$x_3 = 3,003909769 \text{ maka } x_{3+1} = x_3 - \frac{f(x_3)}{f'(x_3)}$$

$$x_4 = 3,003909769 - \frac{0,003925}{1,007819538} = 3,000015$$

## 5. Iterasi 5

$$x_4 = 3,000015 \text{ maka } x_{4+1} = x_4 - \frac{f(x_4)}{f'(x_4)}$$

$$x_5 = 3,000015 - \frac{0,000015}{1,00003} = 3$$

## 6. Iterasi 6

$$x_5 = 3 \text{ maka } x_{5+1} = x_5 - \frac{f(x_5)}{f'(x_5)}$$

$$x_6 = 3 - \frac{0}{1} = 3$$

Karena nilai  $f(3) = 0$  maka salah satu akar dari persamaan tersebut adalah 3.

Iterasi ke-	$x_i$	$f(x)$	$f'(x)$	$x_{i+1}$
1	4	2	3	3,333
2	3,333	0,443889	1,666	3,066560024
3	3,066560024	0,07099026	1,133120048	3,003909769
4	3,003909769	0,003925	1,007819538	3,000015
5	3,000015	0,000015	1,00003	3
6	3	0	1	0

**Algoritma:**

Input:  $f(x)$ ,  $f'(x)$ ,  $x_0$ , tol, M (maksimum iterasi)

Output: akar

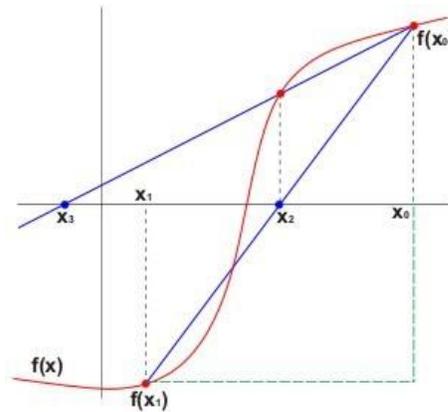
Proses.

1. iterasi  $l = 1$
2. jika  $f'(x_0) = 0$  maka proses gagal.
3. 
$$x_{baru} = x_0 - \frac{f(x_0)}{f'(x_0)}$$
4. jika  $\text{abs}(x_{baru} - x_0) < \text{tol}$  maka akar = x baru
5.  $x_0 = x_{baru}$
6.  $l = l + 1$
7. jika  $l < M$ , maka kembali ke langkah 2.
8. proses belum konvergen

### 1.2.5. Metode Tali Busur

Metode Tali Busur disebut juga Metode Secant, merupakan modifikasi metode Newton-Raphson. Pada metode Newton-Raphson, pada setiap proses iterasinya memerlukan perhitungan nilai dua buah fungsi, yakni  $f(x_n)$  dan  $f'(x_n)$ . Apabila kedua fungsi tersebut tidak rumit, metode tersebut mungkin sangat baik mengingat kekonvergenannya. Akan tetapi, tidak semua fungsi dapat diturunkan dengan mudah, terutama fungsi-fungsi yang rumit. Sedangkan metode Secant hanya memerlukan satu fungsi saja yaitu  $f(x_n)$  yang jika pemrogramannya dilakukan dengan benar, maka nilainya dapat diperoleh dari iterasi sebelumnya  $f(x_{n-1})$ . Sehingga metode Secant akan memerlukan waktu yang lebih sedikit untuk tiap-tiap iterasinya

Pada **Metode Newton-Raphson** memerlukan syarat wajib yaitu fungsi  $f(x)$  harus memiliki turunan  $f'(x)$ . Sehingga syarat wajib ini dianggap sulit karena tidak semua fungsi bisa dengan mudah mencari turunannya. Oleh karena itu muncul ide dari yaitu mencari persamaan yang ekuivalen dengan rumus turunan fungsi. Ide ini lebih dikenal dengan nama **Metode Secant**. Ide dari metode ini yaitu menggunakan gradien garis yang melalui titik  $(x_0, f(x_0))$  dan  $(x_1, f(x_1))$ . Perhatikan gambar dibawah ini.



Persamaan garis  $l$  adalah

$$\frac{x - x_1}{x_0 - x_1} = \frac{y - f(x_1)}{f(x_0) - f(x_1)}$$

Karena  $x = x_2$  maka  $y = 0$  sehingga diperoleh

$$\frac{x_2 - x_1}{x_0 - x_1} = \frac{0 - f(x_1)}{f(x_0) - f(x_1)}$$

$$x_2 - x_1 = -\frac{0 - f(x_1)}{f(x_0) - f(x_1)}$$

$$x_2 = x_1 - \frac{f(x_1)(x_0 - x_1)}{f(x_0) - f(x_1)}$$

$$= x - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}$$

Bentuk umum rumus metode secant ini ditulis

$$x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

### Algoritma

Input :  $f(x), x_0, x_1, \textit{epsilon}, M(\textit{toleransi})$

Output: akar

Proseses :

1.  $l = 1$
2. Jika  $f(x_i) = f(x_i - 1)$  maka proses gagal, selesai
3.  $x_{i+1} = x_i - f(x_i) \frac{x_i - x_{i-1}}{f(x_i) - f(x_i - 1)}$
4. Jika  $\text{abs}(x_{i+1} - x_i) < \text{epsilon}$  maka akar =  $x_{i+1}$ , selesai
5.  $x_i = x_i + 1$
6.  $i = i + 1$
7. Jika  $i < m$  kembali ke langkah 2
8. Proses divergen. Selesai.

### 1.3. PERCOBAAN

Lakukanlah komputasi komputer dengan bahasa pemrograman pascal untuk menentukan pendekatan akar persamaan tak linier dengan langkah-langkah sebagai berikut.

- a. Tuliskan kembali algoritma metode pendekatan akar persamaan tak linier (**metode bagi dua, metode posisi palsu, metode newton-raphson, metode tali busur**) yang akan digunakan.
- b. Gambarkan diagram alir dari algoritma yang telah disusun.
- c. Susunlah bahasa pemrograman berbasis pascal berdasarkan diagram alir yang telah dibuat.
- d. Jalankan program yang dibuat, dan lakukan pengamatan terhadap hasil dengan macam-macam parameter input. (seperti nilai toleransi, jumlah iterasi maksimum, nilai awal dan sebagainya)
- e. Tuliskan hasil percobaan di atas sesuai dengan form laporan percobaan untuk masing-masing metode pendekatan akar persamaan tak linier

## PRAKTIKUM 2

# SISTEM PERSAMAAN LINIER

### 2.1. TUJUAN PEMBELAJARAN

Mahasiswa dapat:

1. melakukan OBE (eliminasi Gauss sederhana)
2. melakukan eliminasi Gauss dengan pivoting parsial
3. melakukan substitusi mundur
4. menggunakan prinsip matriks invers untuk mencari solusi SPL
5. menentukan solusi SPL menggunakan dekomposisi segitiga
6. menentukan solusi SPL menggunakan metode iterasi Jacobi
7. menentukan solusi SPL menggunakan metode iterasi Gauss –Seidel
8. membuat progra komputer untk metode penyelesaian SPL di atas.

### 2.2. DASAR TEORI

#### 2.2.1. Eliminasi Gauss Sederhana

Eliminasi Gauss Sederhana merupakan suatu cara mengoperasikan nilai-nilai di dalam matriks sehingga menjadi matriks yang lebih sederhana. Metode ini ditemukan oleh **Carl Friedrich Gauss**. Prosedur penyelesaian dari metode ini adalah dengan melakukan operasi baris sehingga matriks tersebut menjadi matriks yang Eselon-baris. Ini dapat digunakan sebagai salah satu metode penyelesaian persamaan linier dengan menggunakan matriks. Caranya dengan mengubah persamaan linier tersebut ke dalam matriks dan mengoperasikannya. Setelah menjadi matriks Eselon-baris, lakukan substitusi balik untuk mendapatkan nilai dari variabel-variabel tersebut. Secara umum, sistem persamaan linier adalah sebagai berikut :

---

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ \vdots & \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

Ciri-ciri Eliminasi Gauss Sederhana yaitu jika suatu baris tidak semua nol, maka bilangan pertama yang tidak nol adalah 1; baris nol terletak paling bawah; 1 utama baris berikutnya berada di kanan 1 utama baris di atasnya; di bawah 1 utama harus nol.

### Contoh SPL dan Penyelesaiannya

$$\begin{aligned} x_1 + 2x_2 + 3x_3 + x_4 &= 14 \\ 2x_1 + x_2 + 2x_3 + 2x_4 &= 14 \\ 3x_1 + 2x_2 + 3x_3 + x_4 &= 18 \\ x_1 + 4x_2 + 2x_3 + 3x_4 &= 20 \end{aligned}$$

#### Penyelesaian :

Matriks koefisien yang diperluas sebagai berikut :

$$\begin{pmatrix} 1 & 2 & 3 & 1 & 14 \\ 2 & 1 & 2 & 2 & 14 \\ 3 & 2 & 3 & 1 & 18 \\ 1 & 4 & 2 & 3 & 20 \end{pmatrix}$$

Langkah-langkah :

1. Elemen  $a_{11}$  sebagai pivot untuk membuat nol elemen dibawahnya, yaitu menggunakan keterangan disebelah kanan matriks

$$\begin{pmatrix} 1 & 2 & 3 & 1 & 14 \\ 2 & 1 & 2 & 2 & 14 \\ 3 & 2 & 3 & 1 & 18 \\ 1 & 4 & 2 & 3 & 20 \end{pmatrix} \quad \begin{aligned} b_2 &= b_2 - 2b_1 \\ b_3 &= b_3 - 3b_1 \\ b_4 &= b_4 - b_1 \end{aligned}$$

Hasilnya adalah

$$\begin{pmatrix} 1 & 2 & 3 & 1 & 14 \\ 0 & -3 & -4 & 0 & -14 \\ 0 & -4 & -6 & -2 & -28 \\ 0 & 2 & -1 & 2 & 6 \end{pmatrix}$$

2. Elemen  $a_{22}$  sebagai pivot untuk membuat nol elemen dibawahnya, yaitu menggunakan keterangan disebelah kanan matriks

$$\begin{pmatrix} 1 & 2 & 3 & 1 & 14 \\ 0 & -3 & -4 & 0 & -14 \\ 0 & -4 & -6 & -2 & -28 \\ 0 & 2 & -1 & 2 & 6 \end{pmatrix} \quad \begin{aligned} b_3 &= b_3 - \frac{4}{3}b_2 \\ b_4 &= b_4 + \frac{2}{3}b_2 \end{aligned}$$

Hasilnya adalah

$$\begin{pmatrix} 1 & 2 & 3 & 1 & 14 \\ 0 & -3 & -4 & 0 & -14 \\ 0 & 0 & -\frac{8}{3} & -2 & -\frac{28}{3} \\ 0 & 0 & -\frac{11}{3} & 2 & 6 \end{pmatrix}$$

3. Elemen  $a_{33}$  sebagai pivot untuk membuat nol elemen dibawahnya, yaitu menggunakan keterangan disebelah kanan matriks

$$\begin{pmatrix} 1 & 2 & 3 & 1 & 14 \\ 0 & -3 & -4 & 0 & -14 \\ 0 & 0 & -\frac{8}{3} & -2 & -\frac{28}{3} \\ 0 & 0 & -\frac{11}{3} & 2 & 6 \end{pmatrix} \quad b_4 = b_4 - \frac{33}{24}b_3$$

Hasilnya adalah

$$\begin{pmatrix} 1 & 2 & 3 & 1 & 14 \\ 0 & -3 & -4 & 0 & -14 \\ 0 & 0 & -\frac{8}{3} & -2 & -\frac{28}{3} \\ 0 & 0 & 0 & \frac{57}{12} & \frac{48}{3} \end{pmatrix}$$

Dengan substitusi mundur diperoleh

$$x_4 = 3,367$$

$$x_3 = 0,973$$

$$x_2 = 3,369$$

$$x_1 = 0,92$$

### Algoritma

Input :  $a_{ij}$ ,  $i = 1, 2, 3, \dots, n$       $j = 1, 2, 3, \dots, n+1$

(kolom ke  $n+1$  adalah ruas kanan SPL)

Output :  $x_i, i = 1, 2, 3, \dots, n$

Proses:

Untuk  $k = 1, 2, 3, \dots, n-1$

Jika  $a_{kk} \neq 0$  lanjut ke \*)

Untuk  $i = 1, 2, 3, \dots, n$

Jika  $a_{ik} \neq 0$  maka  $r = i$ , lanjutkan ke \$)

(untuk mencatat nomor baris i)

Jika  $a_{ik} = 0$ , proses gagal

\$) Untuk  $j = k+1, k+2, \dots, n+1$

$$s = a_{kj}, a_{kj} = a_{rj}, a_{rj} = s$$

\*) Untuk  $i = k+1, \dots, n$

$$p = a_{ik} / a_{kk}$$

Untuk  $j = k+1, \dots, n$

$$a_{ij} = a_{ij} - p * a_{kj}$$

Jika  $a_{nn} = 0$ , proses gagal

$$x_n = a_{nn+1} / a_{nn}$$

Untuk  $k = n-1, \dots, 1$

Jumlah = 0

Untuk  $j = k+1, \dots, n$

$$jumlah = jumlah + a_{kj} * x_j$$

$$x_k = (a_{k, n+1} - jumlah) / a_{kk}$$

### 2.2.2. Eliminasi Gauss dengan Pivoting Parsial

Eliminasi Gauss sederhana menggunakan OBE mengolah SPL apa adanya. Dalam eliminasi ini unsur pada diagonal utama dipakai sebagai pedoman (pivot) untuk

mengenolkan unsur di bawahnya. Eliminasi Gauss dengan pivoting parsial menggunakan pivot berupa unsur terbesar absolut dalam setiap kolom saat OBE dilakukan. Dengan demikian urutan kerjanya sbb:

- 1) pilih maks{  $a_{11}, a_{21}, a_{31}, a_{41}, \dots, a_{n1}$  } (kolom 1)
  - a. pindahkan/tukar baris di mana unsur tsb berada dg baris pertama
  - b. lakukan OBE
- 2) pilih maks{  $a_{22}, a_{32}, a_{42}, a_{52}, \dots, a_{n2}$  } (kolom 2)
  - a. mulai  $a_{22}$  ke bawah hingga  $a_{2n}$
  - b. pindahkan/tukar baris di mana unsur tsb berada dg baris ke-dua
  - c. lakukan OBE
- 3) Langkah selanjutnya sama seperti di atas sampai ke baris n-1
- 4) Setelah diperoleh bentuk matrik koefisien berupa segitiga atas, dilakukan substitusi mundur.

Pivoting parsial dilakukan untuk mendapatkan hasil yang lebih akurat karena pembagi (pivot) merupakan bilangan yang paling besar( dalam harga mutlak).

Contoh.

$$1.133 X_1 + 5.281 X_2 = 6.414$$

$$24.14 X_1 - 1.210 X_2 = 22.93$$

$a_{11} < a_{21}$  maka baris ke-2 ditukar dengan baris pertama menjadi:

$$24.14 X_1 - 1.210 X_2 = 22.93$$

$$1.133 X_1 + 5.281 X_2 = 6.414$$

Dengan melakukan OBE diperoleh  $X_1 = 1.000$  dan  $X_2 = 1.000$

Bandingkan hasilnya dengan OBE langsung tanpa pertukaran baris.

### 2.2.3. Invers Matriks

Matriks invers adalah suatu matriks yang apabila dikalikan dengan matriks aslinya akan menghasilkan matriks satuan. Metode matriks invers ini hanya bisa dilakukan pada matriks bujur sangkar yang non-singular (matriks yang determinannya tidak sama dengan nol). Jika A merupakan matriks bujursangkar, maka invers dari matriks A adalah

$A^{-1}$ . Apabila matriks  $A^{-1}$  adalah invers dari matrik  $A$  maka  $A \cdot A^{-1} = A^{-1} \cdot A = I$  (matrik identitas). Konsep ini dapat digunakan dalam menyelesaikan SPL sbb.

$$\begin{aligned} A \cdot X = B &\longrightarrow A^{-1}A \cdot X = A^{-1}B \\ I \cdot X &= A^{-1} \cdot B \\ X &= A^{-1}B \end{aligned}$$

Dengan demikian jika  $A^{-1}$  diketahui maka solusi SPL didapatkan.

Untuk mendapatkan invers matrik dapat dilakukan langkah-langkah berikut ini yang secara teoritis dapat dibuktikan melalui aljabar linier.

Lakukan OBE terhadap matrik  $(A \cdot I)$  yaitu matriks koefisien SPL yang diperluas dengan matriks identitasnya di sebelah kanan, sehingga menjadi matrik  $(I \cdot D)$ , maka matrik  $D$  adalah invers dari  $A$ . Dengan melakukan perkalian matrik pada ruas kanan, langsung diperoleh solusi SPL-nya.

**Contoh:**

Matrik koefisien dari sebuah SPL yang diperluas dengan matriks identitasnya  $(A \cdot I)$  adalah sebagai berikut:

$$\begin{array}{cc} & A \quad \quad I \\ \left\langle \begin{array}{ccc|ccc} 1 & -1 & 0 & 1 & 0 & 0 \\ 3 & 2 & 6 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 & 1 \end{array} \right\rangle & \text{baris 2-3 x baris 1, baris 3-2 x baris 1} \end{array}$$

Lakukan OBE pada kolom pertama sehingga diperoleh

$$\left\langle \begin{array}{ccc|ccc} 1 & -1 & 0 & 0 & 0 & 1 \\ 0 & 5 & 5 & 0 & 1 & -3 \\ 0 & 2 & 1 & 2 & 0 & -2 \end{array} \right\rangle$$

Bagilah baris ke 2 dengan 5 agar diperoleh  $a_{22} = 1$

$$\left\langle \begin{array}{ccc|ccc} 1 & -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0.2 & -0.6 \\ 0 & 2 & 1 & 2 & 0 & -2 \end{array} \right\rangle$$

Lakukan OBE untuk membeuat nol elemen pada kolom 2 kecuali  $a_{22}$  sehingga diperoleh bentuk berikut.

$$\left\langle \begin{array}{ccc|ccc} 1 & 0 & 1 & 0 & 0.2 & -0.4 \\ 0 & 1 & 1 & 0 & 0.2 & -0.6 \\ 0 & 0 & -1 & 1 & 0.4 & -0.8 \end{array} \right\rangle$$

Agar bagian kiri menjadi matrik identitas, baris 3 dibagi  $-1$ .

$$\left\langle \begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & -0.2 & -0.4 \\ 0 & 1 & 0 & 1 & -0.2 & -1.4 \\ 0 & 0 & 1 & -1 & -0.4 & 0.8 \end{array} \right\rangle$$

Matrik identitas nampak pada bagian kiri dan bagian sebelah kanan menyatakan invers dari matrik koefisien SPL, yaitu:

$$A^{-1} = \left\langle \begin{array}{ccc} 1 & -0.2 & -0.4 \\ 1 & -0.2 & -1.4 \\ -1 & -0.4 & 0.8 \end{array} \right\rangle$$

Dengan didapatkannya invers dari  $A$  maka solusi SPL adalah  $X = A^{-1} \cdot B$ .

Matriks yang akan dicari inversnya berderajat  $n \times n$ . Apabila diperluas dengan matrik identitas di sebelah kanan menjadi matrik berderajat  $n \times 2n$ . Elemen  $a_{i,j}$ ,  $i$  dan  $j = 1, 2, 3, \dots, n$  dientry langsung tetapi elemen matrik identitasnya tidak perlu. Buatlah program pendek yaitu : untuk  $i = j, a_{n+j} = 1$  dan jika  $i \neq j, a_{n+j} = 0$

**Algoritma :**

Input :  $n$

$$a_{ij}, i = 1, 2, 3, \dots, n; j = 1, 2, 3, \dots, n$$

Output :  $a_{ij}, i = 1, 2, 3, \dots, n; j = n + 1, n + 2, n + 3, \dots, 2n$

Langkah-langkah:

Untuk  $k = 1, 2, 3, \dots, n$

$r = k$  ( untuk mencatat nomer baris)

**Jika**  $k = n$  lanjut ke \*)

Untuk  $i = k + 1, k + 2, \dots, n$

└─ **Jika**  $abs(a_{ik}) > abs(a_{rk})$  maka  $r = i$

\*) Jika  $a_{ik} = 0$  maka matrik singular. Proses dibatalkan.

Jika  $r \neq k$  maka

$$\begin{array}{l}
 \text{Untuk } j = k, k+1, \dots, 2n \\
 \quad \left| \begin{array}{l} \text{--- } t = a_{kj}; a_{kj} = a_{ij}; a_{ij} = t \end{array} \right. \\
 \\
 p = a_{kk} \\
 \\
 \text{Untuk } j = k, k+1, \dots, 2n \\
 \quad \left| \begin{array}{l} \text{--- } a_{kj} = \frac{a_{kj}}{p} \end{array} \right. \\
 \\
 a_{kk} = 1 \\
 \\
 \text{Untuk } i = 1, 2, \dots, n \\
 \quad \left| \begin{array}{l} \text{Jika } i \neq k \\ \quad p = a_{ik} \\ \quad \text{Untuk } j = k+1, k+2, \dots, 2n \\ \quad \quad \left| \begin{array}{l} \text{--- } a_{ij} = a_{ij} - p * a_{kj} \end{array} \right. \\ \quad \text{--- } a_{ik} = 0 \end{array} \right.
 \end{array}$$

#### 2.2.4. Dekomposisi Segitiga

Alan Mathison Turing berasal dari Maide Vale, London, Inggris, ia adalah seorang peneliti matematika dan komputer. Dia adalah seorang dari peneliti-peneliti komputer modern digital pertama. Salah satu penemuannya adalah Dekomposisi Matriks LU atau lebih dikenal Dekomposisi Matriks Segitiga. Dekomposisi matriks LU merupakan salah satu metode numerik untuk menyelesaikan persamaan matriks. Apabila secara analitik, mungkin akan sangat mudah menyelesaikan persamaan matriks seperti ini  $Ax = B$ , dimana kita hanya mengetahui nilai matriks  $A$  dan matriks  $B$  saja, sementara kita tidak tahu nilai dari matriks  $X$ . Secara analitik kita dapat tuliskan bahwa matriks  $X$ , merupakan perkalian dari inverse matriks  $A$  dengan matriks  $B$ , atau dapat ditulis  $X = A^{-1}B$ . Namun bagaimana jika matriks  $A$  merupakan matriks dengan dimensi  $100 \times 100$  atau  $1000 \times 1000$  akan memerlukan waktu yang banyak dengan menyelesaikan secara analitik. Tetapi dengan metode numerik dengan bantuan kemampuan programming hal seperti itu akan lebih mudah dikerjakan.

Matriks  $[A]$  dapat didekomposisi (difaktorisasi) matriks segitiga bawah ( $L$ ) dengan semua elemen diagonal Utama adalah 1 dan ( $U$ ) adalah matrik segitiga atas yang tidak nol, sedemikian sehingga identitasnya adalah :

$$[A] = [L].[U] \quad \text{atau} \quad A = L.U$$

Pemfaktoran matriks ini dimaksudkan untuk menyederhanakan perkalian antara dua matriks. Notasi matriks  $LU$  sebagai dekomposisi matriks  $A$  dapat dituliskan dalam Sistem Persamaan Linier sebagai berikut:

$$[A].[X] = [L].[U].[X] = [b]$$

Sehingga matriks  $A$  difaktorkan menjadi matriks segitiga bawah  $L$  dan matriks segitiga atas  $U$ :

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \ell_{2,1} & 1 & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \ell_{n,1} & \ell_{n,n-1} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ 0 & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{n,n} \end{bmatrix}$$

### Contoh.

Faktorkan matriks  $A$  berikut ini menjadi  $A = LU$

$$\begin{bmatrix} 1 & 2 & 6 \\ -2 & 3 & 5 \\ 4 & 8 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \ell_{2,1} & 1 & 0 \\ \ell_{3,1} & \ell_{3,2} & 1 \end{bmatrix} \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ 0 & u_{2,2} & u_{2,3} \\ 0 & 0 & u_{3,3} \end{bmatrix}$$

Kalikan ruas kanan sehingga diperoleh hubungan sebagai berikut, kemudian gunakan Sifat kesamaan 2 matrik :

$$\begin{aligned} u_{1,1} &= 1 & u_{1,2} &= 2 & u_{1,3} &= 6 \\ \ell_{2,1}u_{1,1} &= -2 & \ell_{2,1}u_{1,1} + u_{2,2} &= 3 & \ell_{2,1}u_{1,3} + u_{2,3} &= 5 \\ \ell_{3,1}u_{1,1} &= 4 & \ell_{3,1}u_{1,2} + \ell_{3,2}u_{2,2} &= 8 & \ell_{3,1}u_{1,3} + \ell_{3,2}u_{2,3} + u_{3,3} &= -1 \end{aligned}$$

Dari persamaan di atas dapat dicari elemen matrik  $L$  dan matrik  $U$  sehingga diperoleh:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 1 & 2 & 6 \\ 0 & 7 & 17 \\ 0 & 0 & -25 \end{bmatrix}$$

Untuk menghitung  $X$ , diperlukan  $Y$  yaitu  $L.Y = B$ . Setelah didapatkan  $Y$ , maka  $X$  dihitung dari  $U.X = Y$ . Misalnya ruas kanan SPL diatas adalah  $B^T = (9 \ 6 \ 11)$ .

$$LY = B \Rightarrow \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_{1,1} \\ y_{2,1} \\ y_{3,1} \end{bmatrix} = \begin{bmatrix} 9 \\ 6 \\ 11 \end{bmatrix}$$

Sehingga diperoleh :

$$y_{1,1} = 9$$

$$-2 \cdot y_{1,1} + y_{2,1} = 6 \quad , \quad y_{2,1} = 24$$

$$4 \cdot y_{1,1} + y_{3,1} = 11 \quad , \quad y_{3,1} = -25$$

$$Y^T = (9 \quad 24 \quad -25)$$

$$U \cdot X = Y$$

$$\begin{bmatrix} 1 & 2 & 6 \\ 0 & 7 & 17 \\ 0 & 0 & -25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 24 \\ -25 \end{bmatrix}$$

$$x_3 = 1 \quad , \quad x_2 = 1 \quad , \quad x_1 = 1$$

### 2.2.5. Iterasi Jacobi

Metode Iterasi Jacobi merupakan salah satu metode yang dapat digunakan untuk menentukan penyelesaian sistem persamaan linier yang sering disebut SPL. Metode ini ditemukan oleh seorang ahli matematika berkebangsaan Jerman yakni Carl Gustav Jacob Jacobi. Jacobi lahir di Postdam pada tanggal 10 Desember 1804 dan meninggal di Berlin pada tanggal 18 Februari 1951. Sejak 1843, Jacobi menjadi dosen di Universitas Berlin. Selain menemukan metode iterasi Jacobi, pria ini juga mengembangkan teori determinan pada tahun 1841. Nama metode ini diambil dari nama belakang penemunya, yakni Jacobi.

Metode iterasi Jacobi tidak menjamin diperolehnya solusi atau hasil karena dalam hal tertentu proses iterasi tidak konvergen (divergen). Untuk melakukan iterasi diperlukan rumus iterasi dari masing-masing variabel. Rumus iterasi bisa dibuat sesuai keperluan tidak harus mengikuti urutan variabel dan urutan persamaannya. Misalnya rumus iterasi untuk  $x_1$  tidak harus ditentukan menggunakan persamaan 1, demikian pula lainnya. Hanya saja untuk mempermudah proses biasanya rumus iterasi ditentukan sesuai

barisnya, misalnya :  $x_1$  ditentukan dengan persamaan 1,  $x_3$  ditentukan dari persamaan ke-3 dan seterusnya.

$$\begin{array}{cccccc} a_{11}x_1 & + a_{12}x_2 & + \dots & + a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + a_{22}x_2 & + \dots & + a_{2n}x_n & = & b_2 \\ a_{31}x_1 & + a_{32}x_2 & + \dots & + a_{3n}x_n & = & b_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1}x_1 & + a_{n2}x_2 & + a_{n3}x_3 & + a_{n4}x_4 & = & b_n \end{array}$$

Dari persamaan 1 diperoleh rumus iterasi untuk  $x_1$  sebagai berikut:

$$x_1 = \frac{(b_1(a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + \dots + a_{1n}x_n))}{a_{11}}$$

Dari persamaan 2 diperoleh rumus iterasi untuk  $x_2$  sebagai berikut:

$$x_2 = \frac{(b_2(a_{21}x_1 + a_{23}x_3 + a_{24}x_4 + \dots + a_{2n}x_n))}{a_{22}}$$

Begitu seterusnya sehingga diperoleh rumus iterasi untuk variabelnya. Untuk memulai iterasi diperlukan tebakan awal yang dipilih secara bebas, misalnya:

$$x_1 = 0, x_2 = 0, \dots, x_n = 0$$

Dengan demikian, dari perhitungan iterasi pertama diperoleh:

$$\begin{array}{l} x_1 = b_1/a_{11} \\ x_2 = b_2/a_{22} \\ \vdots \\ x_n = b_n/a_{nn} \end{array}$$

Selanjutnya, hasil yang diperoleh di atas digunakan sebagai masukan untuk iterasi berikutnya, dan seterusnya hingga dicapai hasil dengan toleransi atau banyaknya iterasi yang ditentukan. Pada pembuatan program komputer diperlukan batasan maksimum iterasi dan/atau toleransi yang digunakan sebagai dasar penghentian proses.

Toleransi tercapai pada iterasi ke- $i$  apabila  $error\ maksimumnya = maks\{|x_1 - x_1'|, |x_2 - x_2'|, \dots, |x_n - x_n'|\}$  kurang dari atau sama dengan toleransi yang diberikan.

Berikut contoh penyelesaian sistem persamaan linier (SPL) dengan metode iterasi Jacobi. Contoh yang digunakan adalah sistem persamaan linier dengan tiga variabel.

**Contoh**

Diketahui SPL:

$$4x - y + z = 7$$

$$4x - 8y + z = -21$$

$$-2x + y + 5z = 15$$

Persamaan di atas dapat dituliskan dengan beberapa cara, salah satunya adalah:

$$x = (7 + y - z)/4$$

$$y = (21 + 4x + z)/8$$

$$z = (15 + 2x - y)/5$$

Rumus iterasi Jacobi yang terbentuk adalah sebagai berikut:

$$x_{i+1} = (7 + y_i - z_i)/4$$

$$y_{i+1} = (21 + 4x_i + z_i)/8$$

$$z_{i+1} = (15 + 2x_i - y_i)/5$$

Untuk memulai perhitungan diperlukan nilai tebakan awal

Misalkan :  $(x_0, y_0, z_0) = (1, 2, 2)$

**Iterasi I.**

Substitusikan nilai tebakan awal ke dalam rumus iterasi

$$x_1 = (7 + 2 - 2)/4 = 1,75$$

$$y_1 = (21 + 4 + 2)/8 = 3,375$$

$$z_1 = (15 + 2 - 2)/5 = 3,0$$

Error yang terjadi adalah

$$\text{Error1} = |(1,75 - 1)| = 0,75$$

$$\text{Error2} = |(3,375 - 2)| = 3,375$$

$$\text{Error3} = |(3,0 - 2)| = 1$$

Jadi diperoleh error maksimum pada iterasi I = 1,375

**Iterasi II**

$$X_2 = (7 + 6,75 - 6,0)/4 = 1,9375$$

$$Y_2 = (21 + 7 + 3)/8 = 3,875$$

$$Z_2 = (15 + 3,5 - 6,75)/5 = 3,025$$

Error yang terjadi adalah

$$\text{Error1} = |(1,75 - 1,9375)| = 0,1875$$

$$\text{Error2} = |(3,375 - 3,875)| = 0,50$$

$$\text{Error3} = |(3,0 - 3,025)| = 0,025$$

Jadi diperoleh error maksimum pada iterasi  $I = 0,5$  kemudian diuji dengan toleransi yang telah ditetapkan. Demikian seterusnya hingga error atau galat kurang dari toleransi yang ditentukan atau iterasi telah mencapai batas maksimum iterasi yang ditetapkan. Jika sampai batas maksimum iterasi masih diperoleh galat lebih dari toleransi (belum memberikan hasil), maka proses gagal. Hal ini bisa terjadi karena metode iterasi Jacobi merupakan metode terbuka.

### Algoritma:

Input: banyak baris:  $n$

Koefisien SPL:  $a_{ij} : i, j : 1, 2, 3, \dots, n$

Ruas kanan  $c_i : i = 1, 2, 3, \dots, n$

Tebakan awal:  $x_{0i} : i = 1, 2, 3, \dots, n$

Toleransi :  $tol$

Maksimum iterasi:  $M$

Output:  $x_i : i = 1, 2, 3, \dots, n$

Proses:

Iterasi: Untuk  $k = 1, 2, 3, \dots, M$

Error = 0

Untuk  $i = 1, 2, 3, \dots, n$

$$x_i = c_i$$

Untuk  $j = 1, 2, 3, \dots, n$

Jika  $j \neq i$ , maka  $x_i = x_i - a_{ij} x_{0j}$

$$x_i = x_i / a_{ij}$$

Beda =  $\text{abs}(x_i - x_{0i})$

### 2.2.6. Iterasi Gauss-Seidel

Metode Gauss-Seidel merupakan salah satu penemuan terbesar dari Johann Carl Friedrich Gauss (1777-1850). Metode Gauss-Seidel digunakan untuk menyelesaikan Sistem Persamaan Linear (SPL) berukuran besar dan proporsi koefisien nolnya besar, seperti pada sistem-sistem yang banyak ditemukan dalam sistem persamaan diferensial. Penyelesaian SPL dilakukandengan proses iterasi.

Metode iterasi Gauss-Seidel dikembangkan dari gagasan metode iterasi pada solusi persamaan tak linier. Metode iterasi dengan menggunakan Gauss-Seidel adalah metode iterasi yang menghitung nilai hampiran sekarang dengan mengacu pada nilai hampiran terbaru. Pada prinsipnya metode ini mirip dengan metode iterasi Jacobi, hanya saja hasil perhitungan yang baru diperoleh langsung digunakan dalam perhitungan selanjutnya dalam iterasi tsb. Dengan kata lain perhitungan saat ini menggunakan hasil terakhir yang didapat, walaupun masih dalam satu iterasi.

Metode Gauss-Seidel membolehkan pengguna untuk mengontrol *round-off error*. Metode iterasi ini lebih efisien dibandingkan dengan metode langsung, serta dalam hal penggunaan memori komputer maupun waktu komputasi juga lebih efisien. Selain itu bila bentuk dari masalah dapat dipahami, dapat ditentukan nilai perkiraan awal yang lebih dekat, sehingga terjadinya galat dapat diminimalisir.

Proses iterasi dari metode Gauss-Seidel adalah sebagai berikut.

Apabila diketahui sistem persamaan linier:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= b_2 \\ \vdots & \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

Berikan nilai awal darisetiap  $x_i$  ( $i = 1$  s/d  $n$ ), kemudian persamaan diatas dituliskan sebagai berikut:

$$\begin{aligned} x_1 &= \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n) \\ x_2 &= \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3 - \dots - a_{2n}x_n) \end{aligned}$$

$$x_n = \frac{1}{a_{nn}} (b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{n,n-1}x_{n-1})$$

**Algoritma:**

Input: banyakbaris : n

Koefisien SPL:  $a_{ij} : i, j : 1, 2, 3, \dots, n$

Ruaskanan  $c_i : i = 1, 2, 3, \dots, n$

Tebakanawal:  $x_i : i = 1, 2, 3, \dots, n$

Toleransi : tol

Maksimumiterasi : M

Output:  $x_i : i = 1, 2, 3, \dots, n$

Langkah-langkah:

Iterasi: Untuk  $k = 1, 2, 3, \dots, M$

    Error = 0

    Untuk  $i = 1, 2, 3, \dots, n$

        xbaru =  $c_i$

        Untuk  $j = 1, 2, 3, \dots, n$

            Jika  $j \neq i$ , maka xbaru = xbaru -  $a_{ij}x_j$

        xbaru = xbaru /  $a_{ii}$

        Beda = abs( xbaru -  $x_i$ )

        Jika Beda > error, maka error = beda

$x_i =$  xbaru

    Jika error  $\leq$  tol maka selesai

**Contoh**

Misalkan diketahui Sistem Persamaan Linier:

$$4x_1 - x_2 + x_3 = 7$$

$$4x_1 - 8x_2 + x_3 = -21$$

$$-2x_1 + x_2 + 5x_3 = 15$$

Persamaan di atas dapat dituliskan sbb:

$$x_1 = (7 + x_2 - x_3)/4$$

$$\begin{aligned}x_2 &= (21 + 4x_1 + x_3)/8 \\x_3 &= (15 + 2x_1 - x_2)/5\end{aligned}$$

Rumus iterasi Gauss – Seidel yang terbentuk adalah sbb:

$$\begin{aligned}x_{1i+1} &= (7 + x_{2i} - x_{1i})/4 \\x_{2i+1} &= (21 + 4x_{1i} + x_{3i})/8 \\x_{3i+1} &= (15 + 2x_{1i} - x_{2i})/5\end{aligned}$$

Untuk memulai perhitungan diperlukan nilai tebakan awal misalnya

$$(x_1, x_2, x_3) = (0, 0, 0)$$

**Iterasi 1.** Substitusikan nilai tebakan awal ke dalam rumus iterasi.

$$\begin{aligned}x_{11} &= (7 + 0 - 0)/4 = 1,75 \\x_{21} &= (21 + 4 \cdot 1,75 + 0)/8 = 3,50 \\x_{31} &= (15 + 2 \cdot 1,75 - 3,50)/5 = 3,0\end{aligned}$$

Eror yang terjadi adalah:

$$\begin{aligned}eror_1 &= \text{abs}(1,75 - 0) = 1,75 \\eror_2 &= \text{abs}(3,5 - 0) = 3,5 \\eror_3 &= \text{abs}(3,0 - 0) = 3,0\end{aligned}$$

**Iterasi 2:**

$$\begin{aligned}x_{12} &= (7 + 3,5 - 3,0)/4 = 1,875 \\x_{22} &= (21 + 4 \cdot 1,875 + 3,0)/8 = 3,9375 \\x_{32} &= (15 + 2 \cdot 1,875 - 3,9375)/5 = 2,9625\end{aligned}$$

Eror yang terjadi adalah:

$$\begin{aligned}eror_1 &= \text{abs}(1,75 - 1,875) = 0,125 \\eror_2 &= \text{abs}(3,5 - 3,9375) = 0,4375 \\eror_3 &= \text{abs}(3,0 - 2,9375) = 0,0625\end{aligned}$$

Begitu seterusnya, hasil perhitungan terakhir langsung dipakai untuk perhitungan berikutnya sampai tercapai batas toleransi atau maksimum iterasi, seperti pada table berikut.

Iterasi	$x_1$	$x_2$	$x_3$
1	1,75	3,5	3
2	1,875	3,9375	2,9625
3	1,99375	3,99219	2,99906

4	1.99828	3.99902	2.99951
5	1.99988	3.99988	2.99998
6	1.99998	3.99998	2.99999
7	2	4	3
8	2	4	3
9	2	4	3

sehingga diperoleh solusi yaitu:  $x_1 = 2$ ,  $x_2 = 4$ ,  $x_3 = 3$

### 2.2.7. SPL Tridiagonal

Dalam aplikasinya banyak ditemui bentuk SPL khusus yaitu elemen matriks koefisiennya kebanyakan adalah nol dan elemen lainnya membentuk pola tiga jalur. Matriks dengan bentuk ini disebut dengan matriks tridiagonal.

$$\begin{aligned}
 a_1x_1 + b_1x_2 &= d_1 \\
 c_2x_1 + a_2x_2 + b_2x_3 &= d_2 \\
 c_3x_2 + a_3x_3 + b_3x_4 &= d_3 \\
 c_4x_3 + a_4x_4 + b_4x_5 &= d_4 \\
 &\dots\dots\dots
 \end{aligned}$$

Sehingga persamaan matriksnya adalah:

$$\begin{array}{cccccc}
 a_1 & b_1 & 0 & 0 & 0 & \dots 0 & d_1 \\
 c_2 & a_2 & b_2 & 0 & 0 & \dots 0 & d_2 \\
 0 & c_3 & a_3 & b_3 & 0 & \dots 0 & d_3 \\
 0 & 0 & c_4 & a_4 & b_4 & \dots 0 & d_4 \\
 \dots & \dots & \dots & \dots & \dots & \dots \dots & \dots \\
 0 & 0 & 0 & 0 & \dots & \dots a_n & d_n
 \end{array}$$

Jika diselesaikan dengan eliminasi Gauss secara langsung maka banyak perhitungan yang dilakukan sebenarnya tidak perlu dilakukan. Oleh karena itu dilakukan modifikasi.

Ada tiga diagonal dalam SPL ini yaitu:

Diagonal c dengan elemen dari  $c_2$  sampai  $c_n$

Diagonal a dengan elemen dari  $a_1$  sampai  $c_n$

Diagonal b dengan elemen dari  $b_1$  sampai  $b_{n-1}$

Eliminasi Gauss yang dilakukan pada SPL ini cukup sekali saja dan dihasilkan dua diagonal. Selanjutnya adalah dengan substitusi balik.

Contoh:

$$\left| \begin{array}{cccc|c} 2 & -3 & 0 & 0 & -1 \\ 1 & -4 & 7 & 0 & 4 \\ 0 & 2 & 1 & 3 & 6 \\ 0 & 0 & 2 & 1 & 3 \end{array} \right| \quad \text{baris 2} - \frac{1}{2} \text{ baris 1}$$

$$\left| \begin{array}{cccc|c} 2 & -3 & 0 & 0 & -1 \\ 0 & -2.5 & 7 & 0 & 4.5 \\ 0 & 2 & 1 & 3 & 6 \\ 0 & 0 & 2 & 1 & 3 \end{array} \right| \quad \text{baris 3} - \frac{4}{5} \text{ baris 2}$$

$$\left| \begin{array}{cccc|c} 2 & -3 & 0 & 0 & -1 \\ 0 & -2.5 & 7 & 0 & 4.5 \\ 0 & 0 & 6.6 & 3 & 9.6 \\ 0 & 0 & 2 & 1 & 3 \end{array} \right| \quad \text{baris 4} - \frac{2}{6.6} \text{ baris 3}$$

$$\left| \begin{array}{cccc|c} 2 & -3 & 0 & 0 & -1 \\ 0 & -2.5 & 7 & 0 & 4.5 \\ 0 & 0 & 6.6 & 3 & 9.6 \\ 0 & 0 & 0 & 0.909 & 0.909 \end{array} \right|$$

Lalu lakukan substitusi mundur diperoleh:

$$X_4 = \frac{0.909}{0.909} = 1$$

$$X_3 = \frac{9.6-3}{6.6} = 1$$

$$X_2 = \frac{4.5-7}{-2.5} = 1$$

$$X_1 = \frac{-1+3}{2} = 1$$

### Algoritma

Input:  $n, a_i, b_i, c_i, d_i$  dengan  $i = 1, 2, 3, \dots, n$

$$b_n = 0 \text{ dan } c_1 = 0$$

Output:  $x_i : i = 1, 2, 3, \dots, n$

Langkah-langkah:

Untuk  $k = 1, 2, 3, \dots, n - 1$

$$\left. \begin{array}{l} p := \frac{c_{k+1}}{a_i} \\ a_{k+1} := a_{k+1} - p * b_k \\ d_{k+1} := d_{k+1} - p * d_k \\ x_n := c_n / d_n \end{array} \right|$$

Untuk  $k = n - 1, n - 2, \dots, n$

$$\left| \begin{array}{l} x_k := (d_k - b_k * x_{k+1}) / a_k \end{array} \right.$$

### 2.3. PERCOBAAN

Lakukanlah komputasi komputer dengan bahasa pemrograman pascal untuk menentukan pendekatan solusi SPL dengan langkah-langkah sebagai berikut.

- Tuliskan kembali algoritma metode pendekatan solusi SPL (**metode eliminasi gauss sederhana, metode eliminasi gauss dengan pivoting parsial, metode matriks invers, metode dekomposisi segitiga, metode iterasi jacobi, metode iterasi Gauss –Seidel**) yang akan digunakan.
- Gambarkan diagram alir dari algoritma yang telah disusun.
- Susunlah bahasa pemrograman berbasis pascal berdasarkan diagram alir yang telah dibuat.
- Jalankan program yang dibuat, dan lakukan pengamatan terhadap hasil dengan macam-macam parameter input. (seperti nilai toleransi, jumlah iterasi maksimum, nilai awal dan sebagainya)
- Tuliskan hasil percobaan di atas sesuai dengan form laporan percobaan untuk masing-masing metode pendekatan solusi SPL.

## PRAKTIKUM 3

### PENDEKATAN INTEGRAL DAN TURUNAN

#### 3.1. TUJUAN PEMBELAJARAN

Mahasiswa dapat:

1. menentukan pendekatan integral  $f(x)$  dalam interval  $(a,b)$  menggunakan metode trapesium
2. menentukan pendekatan integral  $f(x)$  dalam interval  $(a,b)$  menggunakan metode simpson 1/3
3. menentukan pendekatan turunan menggunakan beda maju, beda mundur dan beda pusat.

#### 3.2. DASAR TEORI

##### 3.2.1. Metode Trapesium

Sebagaimana namanya, metode trapesium merupakan metode integrasi numerik yang didasarkan pada penjumlahan segmen-segmen berbentuk trapesium. Apabila sebuah integral didekati dengan metode trapesium dengan satu segmen saja, maka dapat dituliskan sebagai

$$\int_a^b f(x)dx = \frac{b-a}{2}[f(a) + f(b)] + E \quad (1)$$

Suku pertama pada ruas kanan adalah aturan trapesium yang kita maksudkan, sedangkan suku kedua yang dinyatakan dengan  $E$  adalah kesalahan yang dimiliki oleh metode ini.

Untuk memperoleh ungkapan metode trapesium (1) dan untuk mengetahui seberapa besar kesalahan yang dimiliki oleh metode ini, maka kita perlu melakukan ekspansi deret Taylor luasan  $A(x)$  yang didefinisikan sebagai

$$A(x) = \int_{x_0}^x f(t)dt \quad (2)$$

Ekspnasi deret Taylor untuk luasan  $A(x)$  selanjutnya adalah

$$A(x) = A(x_0) + (x - x_0)A'(x_0) + \frac{(x - x_0)^2}{2} A''(x_0) + \frac{(x - x_0)^3}{6} A'''(x_0) + \dots \quad (3)$$

dengan definisi (2) maka diperoleh

$$A'(x) = f(x), A''(x) = f'(x), A'''(x) = f''(x) \quad (4)$$

Selanjutnya, ungkapan (2) untuk batas bawah integrasi  $x_0$  dan batas atas  $x_0 + h$  menjadi

$$\begin{aligned} \int_{x_0}^{x_0+h} f(x)dx &= 0 + hA'(x_0) + \frac{h^2}{2}A''(x_0) + \frac{h^3}{6}A'''(x_0) + \dots \\ &= hf(x_0) + \frac{h^2}{2}f'(x_0) + \frac{h^3}{6}f''(x_0) + \dots \end{aligned} \quad (5)$$

Dengan mendekati ungkapan turunan pertama dengan beda hingga maju (forward difference)

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h} \quad (6)$$

maka persamaan (2) akan mengambil bentuk

$$I = hf(x_0) + \frac{h^2}{2} \frac{f(x_0 + h) - f(x_0)}{h} + O(h^3) \quad (6)$$

Dengan demikian kita memperoleh pendekatan integral dengan teknik integrasi trapesium adalah

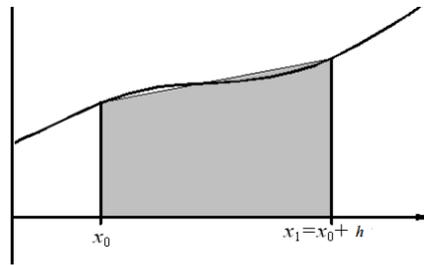
$$\int_{x_0}^{x_0+h} f(t)dt \approx \frac{h}{2} [f(x_0) + f(x_0 + h)] \quad (7)$$

Dari ungkapan (7) dapat diketahui bahwa pendekatan integrasi dengan aturan trapesium memiliki kesalahan yang sebanding dengan  $(h)^3$ . Oleh sebab itu, jika kita membagi dua terhadap  $h$  maka kesalahan hasil integrasi akan tereduksi hingga 1/8 nya. Akan tetapi, ukuran domainnya juga terbagi menjadi dua, sehingga dibutuhkan aturan trapesium lagi untuk mengevaluasinya, selanjutnya sumbangan hasil integrasi tiap domain dijumlahkan. Hasil akhirnya memiliki kesalahan 1/4 nya bukan lagi 1/8 nya.

Untuk memperoleh ungkapan yang lebih teliti mengenai kesalahan pada metode ini, maka marilah kita lakukan perhitungan lebih teliti lagi. Jika kesalahan pendekatan dinyatakan sebagai  $E$ , maka

$$\begin{aligned} E &= \int_{x_0}^{x_0+h} f(t)dx - \frac{h}{2} [f(x_0) + f(x_0 + h)] \\ &= \left[ hf(x_0) + \frac{h^2}{2} f'(x_0) + \frac{h^3}{6} f''(x_0) + \dots \right] - \frac{h}{2} \left[ f(x_0) + f(x_0) + hf'(x_0) + \frac{h^2}{2} f''(x_0) + \frac{h^3}{6} f'''(x_0) + \dots \right] \\ &\approx -\frac{1}{12} h^3 f''(x_0) \end{aligned} \quad (9)$$

Secara grafis ungkapan (8) dapat digambarkan seperti pada gambar dibawah



Ungkapan (8) adalah aturan trapezium untuk satu segmen. Untuk daerah yang dibagi atas  $n$  segmen, maka ungkapan (8) dapat dinyatakan sebagai

$$\int_{x_0}^{x_0+Nh} f(x)dx = \frac{h}{2} [(f_0 + f_1) + (f_1 + f_2) + \dots + (f_{N-2} + f_{N-1}) + (f_{N-1} + f_N)] \quad (10a)$$

atau jika ungkapan (10a) disederhanakan, maka akan menjadi

$$\int_{x_0}^{x_0+Nh} f(x)dx = \frac{h}{2} [f_0 + 2f_1 + 2f_2 + 2f_3 + 2f_{N-3} + 2f_{N-2} + 2f_{N-1} + f_N] \quad (10b)$$

atau secara umum dinyatakan sebagai

$$\int_{x_0}^{x_0+Nh} f(x)dx = \frac{h}{2} \left[ f_0 + f_N + 2 \sum_{n=1}^{N-1} f_n \right] \quad (10c)$$

### Algoritma

- Mendefinisikan fungsi yang akan diintegrasikan
- Menentukan batas bawah  $b$  dan batas atas  $a$  integrasi
- Menghitung lebar segmen yaitu  $h = \frac{b-a}{N}$
- Inisialisasi (memberikan harga awal) fungsi yang diintegrasikan yaitu  $I = f(a) + f(b)$

- Menghitung  $I$  untuk  $n=1$  hingga  $n=N-1$
- Mencetak hasil perhitungan

### Contoh

Gunakan aturan trapesium satu segmen, dua segmen dan empat segmen untuk ungkapan integral

$$\int_0^1 (4x - x^2) dx$$

kemudian hitung kesalahan perhitungan dari masing-masing pendekatan!

### Penyelesaian

Harga eksak untuk ungkapan integral tersebut adalah 1.6667. Harga eksak ini berfungsi untuk memperoleh perbandingan kesalahan antara perhitungan secara analitik dengan hasil pendekatan numerik.

Pendekatan integrasi dengan menggunakan satu segmen

Jika batas bawah  $a = 0$  dan batas atas  $b = 1$ , maka lebar segmen dapat ditentukan dengan

$h = \frac{b-a}{N}$ , karena  $N = 1$  maka lebar segmen  $h = 1$ , sehingga pada

$$x_0 = 0, f_0 = [4(0) - (0)^2] = 0$$

$$x_1 = 1, f_1 = [4(1) - (1)^2] = 3$$

Ungkapan (8) selanjutnya menjadi  $I = \frac{h}{2}[f_0 + f_1]$

$$\text{Jadi } I = \frac{h}{2}[f_0 + f_1] = \frac{1}{2}[0 + 3] = 1.5$$

Kesalahan hasil pendekatan integrasinya :  $\left| \frac{1.6667 - 1.5000}{1.6667} \right| \times 100\% = 10.002\%$

Pendekatan integrasi dengan menggunakan empat segmen

lebar segmen dapat ditentukan dengan  $h = \frac{b-a}{N} = \frac{1-0}{4} = 0,25$ , sehingga pada

$$x_0 = 0, f_0 = [4(0) - (0)^2] = 0$$

$$x_1 = 0 + 0,25 = 0,25, f_1 = [4(0,25) - (0,25)^2] = 0,9375$$

$$x_2 = 0 + 2(0,25) = 0,5, f_1 = [4(0,5) - (0,5)^2] = 1,75$$

$$x_3 = 0 + 3(0,25) = 0,75, f_1 = [4(0,75) - (0,75)^2] = 2,4375$$

$$x_4 = 0 + 4(0,25) = 1, f_1 = [4(1) - (1)^2] = 3$$

Ungkapan (8) selanjutnya menjadi  $I = \frac{h}{2} [f_0 + 2f_1 + 2f_2 + 2f_3 + f_4]$

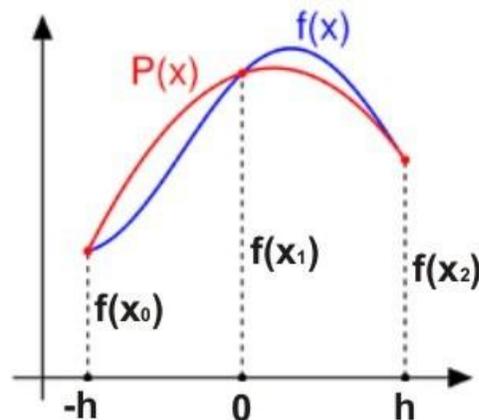
$$\text{Jadi } I = \frac{0,25}{2} [0 + 2(0,9375) + 2(1,75) + 2(2,4375) + 3] = 1.6563$$

$$\text{Kesalahan hasil pendekatan integrasinya : } \left| \frac{1.6667 - 1.6563}{1.6667} \right| \times 100\% = 0,624\%$$

### 3.2.2. Metode Simpson 1/3

Aturan Simpson adalah suatu aturan yang digunakan untuk menghitung luas suatu kurva polinom berderajat dua  $p_2(x)$  atau berderajat tiga  $p_3(x)$  dengan pendekatan yaitu pendekatan menggunakan partisi berbentuk parabola. Metode ini berasal dari matematikawan Thomas Simpson (1710-1761), yang berasal dari Leicestershire, [Inggris](#). Kaidah Simpson banyak digunakan, misalnya oleh [arsitektur perkapalan](#) untuk menghitung kapasitas kapal atau sekoci.

**Aturan Simpson 1 per 3** ini mempartisi kurva polinom berderajat dua  $p_2(x)$  dengan 3 titik, 5 titik, 7 titik dan seterusnya sedemikian sehingga ruang partisi yang dibentuk berjumlah genap. Selanjutnya akan diturunkan rumus Metode Simpson 1/3.



### Gambar 1. Ilustrasi Metode Simpson 1/3

Perhatikan gambar diatas, misal kita menggunakan  $f(x) = Ax^2 + Bx + C$  dengan tiga titik partisi yaitu  $x_0, x_1$ , dan  $x_3$  dengan mengambil  $x_0 = -h$ ,  $x_1 = 0$  dan  $x_2 = h$ . Perlu diingat bahwa partisi yang dilakukan disini dianggap sama besar untuk setiap ruang partisi.

Integralkan  $f(x)$  dengan batas atas masing-masing  $-h$  dan  $h$  sehingga diperoleh luas dibawah kurva.

$$\begin{aligned} \int_{-h}^h (Ax^2 + Bx + C)dx &= \left. \frac{1}{3}Ax^3 + \frac{1}{2}Bx^2 + Cx \right|_{-h}^h = \left[ \frac{1}{3}Ah^3 + \frac{1}{2}Bh^2 + Ch \right] - \left[ -\frac{1}{3}Ah^3 + \frac{1}{2}Bh^2 - Ch \right] \\ &= \frac{2}{3}Ah^3 + 2Ch \\ &= \frac{1}{3}h(2Ah^2 + 6C) \dots (i) \end{aligned}$$

Selanjutnya substitusi nilai  $-h$ ,  $0$  dan  $h$  ke  $f(x)$ , sedemikian sehingga diperoleh

$$(-h, f(x_0)) \rightarrow f(a) = Ah^2 - Bh + c \dots (ii)$$

$$(0, f(x_1)) \rightarrow f(h) = c \dots (iii)$$

$$(h, f(x_2)) \rightarrow f(b) = Ah^2 + Bh + c \dots (iv)$$

Eliminasi (ii) dan (iii) :

$$f(x_0) = Ah^2 - Bh + c$$

$$f(x_1) = c \quad -$$

$$f(x_0) - f(x_1) = Ah^2 - Bh \dots (v)$$

Eliminasi (iv) dan (iii) di dapat

$$f(x_2) - f(x_1) = Ah^2 + Bh \dots (vi)$$

Eliminasi (v) dan (vi) didapat

$$f(x_0) - f(x_1) + (f(x_2) - f(x_1)) = Ah^2 - Bh + (Ah^2 + Bh)$$

$$f(x_0) - 2f(x_1) + f(x_2) = 2Ah^2 \dots (vii)$$

Substisusi (iii) dan (vii) ke (i)

$$\begin{aligned}
&= \frac{1}{3}h((f(x_0) - 2f(x_1) + f(x_2)) + 6f(x_1)) \\
&= \frac{1}{3}h [f(x_0) + 4f(x_1) + 2f(x_2)]
\end{aligned}$$

Dengan kata lain ruus simpson 1 per 3 untuk 1 pias adalah

$$s_2(x) = \frac{1}{3}h [f(x_0) + 4f(x_1) + 2f(x_2)], h = \frac{x_1 - x_0}{2}$$

Berikutnya Rumus Simpson 1 per 3 untuk n pias

$$\begin{aligned}
s_n(x) &= \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx + \dots + \int_{x_{n-2}}^{x_n} f(x)dx \\
&= \frac{1}{3}h [f(x_0) + 4f(x_1) + f(x_2)] + \frac{1}{3}h [f(x_2) + 4f(x_3) + f(x_4)] + \dots \\
&\quad + \frac{1}{3}h [f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)] \\
&= \frac{1}{3}h (f(x_0) + 4f(x_1) + f(x_2) + f(x_2) + 4f(x_3) + f(x_4) + \dots \\
&\quad + f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)) \\
&= \frac{1}{3}h[(f(x_0) + f(x_n)) + 4\sum f(x_i) + 2\sum f(x_{2i})]
\end{aligned}$$

Untuk lebih memahami penggunaan materi simpson 1 per 3, berikut merupakan contoh penggunaannya :

Hitunglah luas dari  $e^x$  dengan batas (0,4) dengan menggunakan 4 pias (  $n = 4$  )

**Penyelesaian :**

$$h = \frac{x_1 - x_0}{n} = \frac{4 - 0}{4} = 1$$

$$x_0 = 0$$

$$x_1 = a + h = 1$$

$$x_2 = a + 2h = 2$$

$$x_3 = a + 3h = 3$$

$$x_4 = a + 4h = 4$$

$$s_4(x) = \frac{1}{3}h [(f(x_0) + f(x_4)) + 4(f(x_1) + f(x_3)) + 2f(x_2)]$$

$$\begin{aligned}
&= \frac{1}{3}(1)[(e^0 + e^4) + 4(e^1 + e^3) + 2e^2] \\
&= \frac{1}{3}[(1 + 54.5981) + 4(2.7182 + 20.0855) + 2(7.3890)] \\
&= \frac{1}{3}[55.5981 + 91.2148 + 14.778] \\
&= 53.8636
\end{aligned}$$

**Algoritma:**

- Definisikan fungsi  $y = f(x)$
- Tentukan batas bawah (a) dan batas atas (b) integrasi
- Tentukan jumlah pembagi (n)
- Jika iterasi habis dibagi 2 dan tidak sama dengan n, maka lakukan perhitungan dibawah ini :

$$2 \sum_{i=1}^{N-1} f(x_i)$$

- Jika tidak maka fungsi  $f(x)$  akan dikalikan 4 kemudian dijumlahkan, dengan rumus :

$$4 \sum_{i=1}^{N-1} f(x_i)$$

- Hitunglah luas daerah :

$$L = \frac{h}{3} ((f(x_0) + f(x_n)) + 4 \sum f(x_i) + 2 \sum f(x_{2i}))$$

### 3.2.3. Pendekatan Turunan

Pendekatan nilai turunan sebuah fungsi secara numeric pada umumnya dikenakan pada fungsi diskrit yang nilainya disajikan dalam bentuk table. Berbeda dengan integral yang pada umumnya diberlakukan untuk fungsi kontinyu, maka hasil pendekatan turunan secara numeric kurang akurat.

Turunan yang didefinisikan sebagai limit hasil bagi selisih nilai fungsi dengan beda ( delta  $x$  ) yang mendekati nol, menghendaki nilai yang sangat kecil. Pembagian dengan bilangan sangat kecil secara numeric sangat riskan, oleh karena itu salah satu

cara untuk penghitungan turunan secara numeric ini dengan membuat barisan 'delta  $x$ ' yang menuju nol.

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Dari definisi ini dibuat rumus turunan untuk pendekatan numeric sebagai berikut:

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

Untuk mencermati keakuratan hasil pendekatan turunan ini di buat barisan  $h_i$  sehingga hasil perhitungannya dapat ditelusuri.

### Contoh

Tentukan hampiran turunan  $y = e^x$  di  $x=1$  dengan langkah  $h_i = 10^{-i}, i=1,2,\dots,10$

Penyelesaian.

Dibuat table nilai  $h_j, f_i = f(1+h_i), f_i - e, f'(1)$  sbb.

$h_i$	$f_i = f(1+h_i)$	$f_i - e$	$f(1)$
0.1	3.004166024	0.285884196	2.858841960
0.01	2.745601015	0.027319187	2.731918700
0.001	2.721001470	0.002719642	2.719642000
0.0001	2.718553670	0.000271842	2.718420000
0.00001	2.718309011	0.000027183	2.718300000
0.000001	2.718284547	0.000002719	2.719000000
0.0000001	2.718282100	0.000000272	2.720000000
0.00000001	2.718281856	0.000000028	2.800000000
0.000000001	2.718281831	0.000000003	3.000000000
0.0000000001	2.718281828	0.000000000	0.000000000

Perhitungan untuk integral numeric, semakin kecil  $h$  semakin akurat hasilnya. Tetapi untuk turunan hal itu tidak berlaku. Semakin kecil  $h$ , semakin dekat jarak antara  $x$  dan  $x+h$ , semakin dekat pula nilai  $f(x+h)$  dan  $f(x)$ . Akibatnya selisihnya sangat kecil. Hasil turunannya juga mendekati nol seperti terlihat pada baris terakhir table di atas.

## Penurunan Rumus-Rumus Hampiran Turunan

Rumus turunan dapat dicari menggunakan deret **Taylor** untuk fungsi  $f(x)$  dengan memperluas dalam bentuk deret seperti berikut ini.

$$f(x+h) = f(x) + f'(x)h + f''(x)h^2/2! + f'''(x)h^3/3! + \dots \quad (1)$$

$$f(x+h) = f(x) + f'(x)h + f''(x)h^2/2! - f'''(x)h^3/3! + \dots \quad (2)$$

Apabila dari persamaan 1) dipotong sampai suku kedua, maka diperoleh

$$f(x+h) = f(x) + f'(x)h \text{ dengan pemotongan } G(h) = f''(x)h^2/2! + \dots$$

Jadi  $f'(x) = \frac{f(x+h_i) - f(x)}{h_i} + G(h)/h$ , yang merupakan **rumus beda maju**.

Galat yang terjadi adalah  $G(h)/h$  yang memiliki pangkat tertinggi untuk  $h$  adalah 1 yaitu  $G(h)/h = f''(x)h^2/2! - f'''(x)h^3/3! h + \dots$

Dikatakan galat rumus beda maju memiliki orde  $h$  atau orde 1.

Apabila 1) - 2) maka diperoleh:

$$f(x+h) - f(x-h) = 2f'(x)h + E(f''', h^3)$$

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

yang merupakan rumus **beda pusat**, yaitu hampiran turunan di satu titik menggunakan titik di depannya dan titik di belakangnya. Disebut beda pusat karena turunan di titik yang menjadi tengah-tengah titik yang digunakan untuk hampiran.

Galat yang terjadi dalam rumus beda pusat adalah orde  $h^2$  atau orde 2. Rumus beda mundur diperoleh melalui persamaan 2) dengan cara memotong deret sampai suku ke-2. Berlawanan dengan beda maju, untuk mencari turunan di sebuah titik, digunakan titik di belakangnya sebagai berikut.

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

Galat yang terjadi memiliki orde  $h$  atau orde 1.

Rumus-rumus lain baik beda pusat, beda maju ataupun beda mundur bisa dikembangkan dengan menggunakan beberapa titik di depan atau dan dibelakangnya. Misalnya  $f(x+2h)$ ,  $f(x-2h)$ ,  $f(x+3h)$  dst.

$$f'(x) = \frac{-f(x+h) + 8f(x) - 8f(x-h) + f(x-2h)}{12h}$$

Untuk mencari rumus turunan ke-2, ke-3 dst lakukan pemotongan deret sesuai keperluan. Misalnya deret 1) dan deret 2) dipotong sampai 3 suku kemudian dijumlahkan maka diperoleh rumus turunan ke-2.

### Algoritma

- Tentukan  $x = x_0$  dan  $y = y_0$
- Tentukan nilai awal  $x_0$  dan nilai akhir  $x_a$  dari variable bebas
- Tentukan nilai  $h$
- Inisialisasi  $i = 0$
- Masukkan persamaan  $f(x, y)$
- Vector  $x(i) = [x_0, x_0 + h, x_0 + 2h, \dots, x_a]$
- Jumlah loop,  $n = (x_a - x_0)/h$
- Untuk  $i = 0$  sampai  $n - 1$  maka :
  - $y_{i+1} = y_i + hf(x_i, y_i)$
  - $x = x + h$
  - Simpan nilai  $x_i, y_i$
  - Lanjutkani

### 3.3. PERCOBAAN

Lakukanlah komputasi komputer dengan bahasa pemrograman pascal untuk menentukan pendekatan integral dan turunan dengan langkah-langkah sebagai berikut.

- a. Tuliskan kembali algoritma metode pendekatan integral dan turunan (**metode trapesium, metode simpson 1/3, metode beda maju, metode beda mundur, dan metode beda pusat**) yang akan digunakan.

- b. Gambarkan diagram alir dari algoritma yang telah disusun.
  - c. Susunlah bahasa pemrograman berbasis pascal berdasarkan diagram alir yang telah dibuat.
  - d. Jalankan program yang dibuat, dan lakukan pengamatan terhadap hasil dengan macam-macam parameter input. (seperti nilai toleransi, jumlah iterasi maksimum, nilai awal dan sebagainya)
  - e. Tuliskan hasil percobaan di atas sesuai dengan form laporan percobaan untuk masing-masing metode pendekatan integral dan turunan.
-

## PRAKTIKUM 4 INTERPOLASI

### 4.1. TUJUAN PEMBELAJARAN

Mahasiswa dapat:

1. menentukan polinom Lagrange dari data yang diberikan
2. menentukan nilai interpolasi di sebuah titik dari sejumlah data yang diberikan.

### 4.2. DASAR TEORI

#### 4.2.1. Interpolasi Polinom Lagrange

Interpolasi lagrange adalah suatu bentuk interpolasi dengan fungsi pendekatan berupa fungsi polinomial lagrange. Pada transformasi lagrange, fungsi polynomial pangkat  $n$  memerlukan  $n+1$  titik. Bila jumlah titiknya 2 buah, maka interpolasi lagrange akan menjadi interpolasi linear. Untuk mencari titik  $(x,y)$  pada nilai  $x$  yang ditentukan dengan diketahui  $n$  buah titik  $(x_1,y_1)$ ,  $(x_2,y_2), \dots, (x_n,y_n)$  menggunakan interpolasi lagrange.

Interpolasi polinomial Lagrange hampir sama dengan polinomial Newton, tetapi tidak menggunakan bentuk pembagian beda hingga. Interpolasi polinomial Lagrange dapat diturunkan dari persamaan Newton. Interpolasi Lagrange diterapkan untuk mendapatkan fungsi polinomial  $P(x)$  berderajat tertentu yang melewati sejumlah titik data. Misalnya, kita ingin mendapatkan fungsi polinomial berderajat satu yang melewati dua buah titik yaitu  $(x_0, y_0)$  dan  $(x_1, y_1)$ .

Diberikan beberapa titik  $(x_i, y_j)$ , kemudian diminta untuk menentukan (memperkirakan) nilai ordinat suatu titik  $y_k$ , jika diketahui  $x_k$  yang berada diantara  $x_i$  yang diketahui. Masalah ini disebut interpolasi. Apabila titik yang akan dicari berada di luar kelompok titik-titik yang diketahui disebut ekstrapolasi.

Interpolasi dapat dilakukan dengan berbagai cara. Metode yang paling sederhana adalah interpolasi linier. Hanya diperlukan dua titik, kemudian dibuat garis lurus melalui kedua titik tsb. Nilai ordinat  $y_k$  yang dicari di titik  $x_k$  menggunakan persamaan garis lurus tersebut.

Metode berikutnya yang masih tergolong sederhana adalah interpolasi kuadrat. Sesuai dengan namanya, interpolasi ini menggunakan fungsi kuadrat untuk menentukan nilai interpolasi yang diinginkan. Hanya diperlukan 3 titik untuk menentukan fungsi kuadrat yang dipakai untuk menginterpolasi di titik  $x_k$  yang berada di antara 3 titik yang diketahui.

### Penurunan Rumus Polinomial Lagrange

Diberikan  $(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n)$  dengan  $\Delta x_i$  sebarang. Lagrange mempunyai pemikiran mengalikan  $f_j$  dengan suatu polinom yang bernilai 1 pada  $x_j$  dan 0 pada  $n$  titik simpul lainnya dan kemudian menjumlahkan  $n+1$  polinom tersebut untuk memperoleh polinom interpolasi tunggal berordo  $n$  atau lebih kecil.

Rumus interpolasi lagrange:

- Polinomial interpolasi mempunyai bentuk :

$$L_n(x) = f_0 b_0(x) + f_1 b_1(x) + f_2 b_2(x) + \dots + f_n b_n(x)$$

Dengan  $b_k(x)$  = suatu polinomial derajat "n"

Polinomial  $b_k(x)$  dapat dicari dengan menggunakan  $n+1$  persamaan constraint.

- Persamaan *constraint* dapat dibuat sebagai berikut :

$$L_n(x_i) = f_i \quad ; \quad i = 0, 1, 2, \dots, n$$

Sehingga :

$$L_n(x_0) = f_0 \rightarrow f_0 b_0(x_0) + f_1 b_1(x_0) + \dots + f_n b_n(x_0) = f_0$$

$$L_n(x_1) = f_1 \rightarrow f_0 b_0(x_1) + f_1 b_1(x_1) + \dots + f_n b_n(x_1) = f_1$$

⋮

$$L_n(x_n) = f_n \rightarrow f_0 b_0(x_n) + f_1 b_1(x_n) + \dots + f_n b_n(x_n) = f_n$$

Untuk mempermudah penyelesaian persamaan constraint, maka dipilih :

$$b_k(x_i) = \begin{cases} 1 & ; \quad i = k \\ 0 & ; \quad i \neq k \end{cases}$$

Persamaan tersebut telah memenuhi persamaan constraint.

- Bentuk persamaan polinomial  $b_k(x)$  adalah sebagai berikut :

$$b_k(x) = C_k (x - x_0)(x - x_1)(x - x_2) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_n)$$

Sesuai pilihan di atas yang cocok dengan constraint yaitu  $b_k(x_k) = 1$

Maka konstanta  $C_k$  dapat dicari dengan rumusan berikut :

$$C_k = \frac{1}{(x_k - x_0)(x_k - x_1) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)}$$

Dengan demikian semua polinomial  $b_k(x)$  diperoleh :

$$b_0(x) = C_0 (x - x_1)(x - x_2) \dots (x - x_n)$$

$$b_1(x) = C_1 (x - x_0)(x - x_2)(x - x_3) \dots (x - x_n)$$

$$b_2(x) = C_2 (x - x_0)(x - x_1)(x - x_3) \dots (x - x_n)$$

⋮

$$b_n(x) = C_n (x - x_0)(x - x_1) \dots (x - x_{n-1})$$

di mana :

$$C_0 = \frac{1}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3) \dots (x_0 - x_n)}$$

$$C_1 = \frac{1}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3) \dots (x_1 - x_n)}$$

$$C_2 = \frac{1}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3) \dots (x_2 - x_n)}$$

⋮

$$C_n = \frac{1}{(x_n - x_0)(x_n - x_1)(x_n - x_2) \dots (x_n - x_{n-1})}$$

Jadi polinomial  $b_k(x)$  dapat ditulis secara lengkap :

$$b_0(x) = \frac{(x - x_1)(x - x_2) \dots (x - x_n)}{(x_0 - x_1)(x_0 - x_2) \dots (x_0 - x_n)}$$

$$b_1(x) = \frac{(x - x_0)(x - x_2) \dots (x - x_n)}{(x_1 - x_0)(x_1 - x_2) \dots (x_1 - x_n)}$$

$$b_2(x) = \frac{(x - x_0)(x - x_1)(x - x_3) \dots (x - x_n)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3) \dots (x_2 - x_n)}$$

⋮

$$b_n(x) = \frac{(x - x_0)(x - x_1)(x - x_2) \dots (x - x_{n-1})}{(x_n - x_0)(x_n - x_1)(x_n - x_2) \dots (x_n - x_{n-1})}$$

Sehingga persamaan polinomial dari lagrange interpolation dapat dirumuskan sebagai berikut :

$$L_n(x) = \sum_{k=0}^n f_k \frac{(x-x_0)(x-x_1)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)(x_k-x_1)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)}$$

Atau jika  $l_n(x) = (x-x_0)(x-x_1)(x-x_2)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)$

Lagrange dengan tepat merumuskan bentuk polinom yang melalui titik-titik yang diketahui untuk digunakan meng-interpolasi suatu titik. Sehingga rumus polinom lagrange adalah sebagai berikut.

$$P_n(x) = \sum_{i=0}^n L_i(x) y_i$$

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j} \quad , i = 1,2,3, \dots, n; j = 1,2,3 \dots, n$$

**Contoh :**

1. Tentukan polinom linier menurut Lagrange

$n = 1$  maka

$$P_1(x) = \sum_{i=0}^1 L_i(x) y_i = L_0(x) y_0 + L_1(x) y_1$$

$i = 0$

$$L_0(x) = \prod_{\substack{j=0 \\ j \neq 0}}^1 \frac{x-x_j}{x_0-x_j} = \frac{x-x_1}{x_0-x_1}$$

$i = 1$

$$L_1(x) = \prod_{\substack{j=0 \\ j \neq 1}}^1 \frac{x-x_j}{x_1-x_j} = \frac{x-x_0}{x_1-x_0}$$

$$\text{Jadi } P_1(x) = \frac{x-x_1}{x_0-x_1} y_0 + \frac{x-x_0}{x_1-x_0} y_1$$

Dengan ditemukannya polinom maka  $y_k = P_1(x_k)$  dapat dihitung.

2. Tentukan  $P_2(x)$  !

$n = 2$

$$P_2(x) = \sum_{i=0}^2 L_i(x) y_i = L_0(x) y_0 + L_1(x) y_1 + L_2(x) y_2$$

$$i = 0$$

$$L_0(x) = \prod_{\substack{j=0 \\ j \neq i}}^2 \frac{x - x_j}{x_i - x_j} = \frac{x - x_1}{x_0 - x_1} \frac{x - x_2}{x_0 - x_2}$$

$$i = 1$$

$$L_1(x) = \prod_{\substack{j=0 \\ j \neq i}}^2 \frac{x - x_j}{x_i - x_j} = \frac{x - x_0}{x_1 - x_0} \frac{x - x_2}{x_1 - x_2}$$

$$i = 2$$

$$L_2(x) = \prod_{\substack{j=0 \\ j \neq i}}^2 \frac{x - x_j}{x_i - x_j} = \frac{x - x_0}{x_2 - x_0} \frac{x - x_1}{x_2 - x_1}$$

$$\text{Jadi } P_2(x) = \frac{x - x_1}{x_0 - x_1} \frac{x - x_2}{x_0 - x_2} y_0 + \frac{x - x_0}{x_1 - x_0} \frac{x - x_2}{x_1 - x_2} y_1 + \frac{x - x_0}{x_2 - x_0} \frac{x - x_1}{x_2 - x_1} y_2$$

Jika banyak titik yang diketahui  $n$  buah maka polinomnya berderajat  $n-1$ . Untuk setiap suku dari polinom berderajat  $k$  mengandung sebanyak  $k$  factor

Perkalian berbentuk :

$$\frac{x - x_j}{x_i - x_j}$$

### Kekurangan Metode Polinomial Lagrange

Djuhana (2002) melakukan analisis kesalahan metode Lagrange. Kesalahan itu terjadi saat metode ini ingin memberikan aproksimasi suatu fungsi  $f(x)$  dengan polinomial  $L_n(x)$  yaitu :

$$E_n(x) = f(x) - L_n(x) = \frac{(x - x_0)(x - x_1) \dots (x - x_n)}{(n+1)!} f^{n+1}(\xi)$$

di mana  $\xi$  tergantung pada nilai  $x$  dan tidak diketahui di dalam interval  $x$ ,  $x_0$  dan  $x_n$ .

### contoh soal :

Carilah  $f(9.2)$  dengan interpolasi lagrange dengan  $n = 3$  dan  $f(9.0)=2.19722$ ,  $f(9.5)=2.25129$ ,  $f(10.0)=2.30259$ ,  $f(11.0)=2.39790$

Penyelesaian :

$$L_3(x) = \frac{l_0(x)}{l_0(x_0)} f_0 + \frac{l_1(x)}{l_1(x_1)} f_1 + \frac{l_2(x)}{l_2(x_2)} f_2 + \frac{l_3(x)}{l_3(x_3)} f_3$$

$$l_0(x_0) = (x_0 - x_1)(x_0 - x_2)(x_0 - x_3) = -1.00000$$

$$l_0(x) = (x - x_1)(x - x_2)(x - x_3) = -0.43200$$

$$l_1(x_1) = (x_1 - x_0)(x_1 - x_2)(x_1 - x_3) = 0.37500$$

$$l_1(x) = (x - x_0)(x - x_2)(x - x_3) = 0.28800$$

$$l_2(x_2) = (x_2 - x_0)(x_2 - x_1)(x_2 - x_3) = -0.50000$$

$$l_2(x) = (x - x_0)(x - x_1)(x - x_3) = 0.10800$$

$$l_3(x_3) = (x_3 - x_0)(x_3 - x_1)(x_3 - x_2) = 3.00000$$

$$l_3(x) = (x - x_0)(x - x_1)(x - x_2) = 0.04800$$

$$\begin{aligned} L_3(9.2) &= \frac{-0.43200}{-1.00000} 2.19722 + \frac{0.28800}{0.37500} 2.25129 + \frac{0.10800}{-0.50000} 2.30259 \\ &\quad + \frac{0.04800}{3.00000} 2.39700 \end{aligned}$$

$$= 2.21920 \text{ (eksak sampai 5 angka decimal)}$$

#### 4.2.2. Interpolasi Beda Terbagi Newton

Interpolasi linear dan kuadratik merupakan kasus khusus interpolasi derajat yang lebih tinggi. Dalam hal ini, digunakan konsep beda terbagi sebagai berikut:

- Untuk order pertama dihitung dari derivatif fungsi  $f(x)$  secara diskrit :

$$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0}, \quad x_0 \neq x_1$$

- Untuk order yang lebih tinggi, dipakai beda terbagi order yang lebih rendah secara rekursif:

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}$$

Sehingga:

$$f[x_0, \dots, x_n] = \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0}$$

→ disebut **rumus beda terbagi Newton**.

Dengan demikian, interpolasi beda terbagi Newton dapat dirumuskan sebagai berikut:

$$P_n(x) = f_0 + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] + \dots + (x - x_0) \dots (x - x_{n-1})f[x_0, \dots, x_n]$$

*Contoh.*

Carilah  $f(9.2)$  dari interpolasi menggunakan titik-titik (8.0, 2.079442), (9.0, 2.197225), (9.5, 2.251292), (11.0, 2.397895).

Penyelesaian :

Dengan menggunakan tabel :

i	$x_i$	$y_i = f_i$	$\Delta_1 = f[x_i, x_{i+1}]$	$\Delta_2 = f[x_i, x_{i+1}, x_{i+2}]$	$\Delta_3 = f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
0	8.0	2.0794 42			
1	9.0	2.1972 25	0.117783	-0.006433	
2	9.5	2.2512 92	0.108134 0.097735	-0.005200	0.000411
3	11. 0	2.3978 95			

$$P_3(x) = 2.079442 + 0.117783(x-8.0) - 0.006433(x-8.0)(x-9.0) + 0.000411(x-8.0)(x-9.0)(x-9.5)$$

$$P_3(9.2) = 2.079442 + 0.141340 - 0.001544 - 0.000030 \\ = 2.219208 \rightarrow \text{eksak sampai 6 desimal.}$$

**Algoritma :**

Dari penjelasan dan perumusan di atas, algoritma Interpolasi Beda Terbagi Newton dapat disusun sebagai berikut:

- Tujuan: menghitung  $p_n(x)$  dari  $f(x)$  pada  $x$

- Algoritma interpolasi :

input :  $(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n); x$

output :  $p_n(x)$

Langkah-langkah :

1. Untuk  $j = 0, 1, 2, \dots, n$  lakukan:  $f(x_j) := f_j$

2. Untuk  $m = 1, 2, \dots, n-1$  lakukan

Untuk  $j = 0, 1, 2, \dots, n-m$ , lakukan

$$f[x_j, x_{j+1}, \dots, x_{j+m}] = \frac{f[x_{j+1}, \dots, x_{j+m}] - f[x_j, x_{j+1}, \dots, x_{j+m-1}]}{x_{j+m} - x_j}$$

3.  $p_0(x) := f_0$

4. Untuk  $k = 1, 2, 3, \dots, n$  lakukan

$$p_k(z) = p_{k-1}(z) + (z - x_0) \dots (z - x_{k-1}) f[x_0, \dots, x_k]$$

**Contoh**

Suatu pengusaha baru mencoba memproduksi sejumlah sepatu. Saat ia memproduksi satu pasang sepatu, untung yang diperoleh adalah 2% dari modal usaha. Jika ia memproduksi dua pasang maka tidak ada keuntungan yang diperoleh. Kemudian ia mencoba memproduksi 3 pasang dan rugi 1% dari modal usaha. Jika ia mencoba lagi memproduksi 6 pasang, apakah pengusaha tersebut mengalami keuntungan atau kerugian?

Penyelesaian:

Dengan menggunakan titik (1,2), (2,0) dan (3,-1) dapat dibuat tabel sebagai berikut:

$i$	$x_i$	$y_i = f_i$	$\Delta_1 = f[x_i, x_{i+1}]$	$\Delta_2 = f[x_i, x_{i+1}, x_{i+2}]$
-----	-------	-------------	------------------------------	---------------------------------------

0	1	2	-2	$\frac{1}{2}$
1	2	0		
2	3	-1	-1	

$$P_2(x) = 2 + (-2)(x-1) + \left(\frac{1}{2}\right)(x-1)(x-2)$$

$$P_2(6) = 2 + (-2)(6-1) + \left(\frac{1}{2}\right)(6-1)(6-2)$$

$$= 2$$

Jadi, jika pengusaha memproduksi 6 pasang sepatu akan diperoleh keuntungan sebesar 2% dari modal usaha.

#### **Kekurangan dan kelebihan dari interpolasi beda terbagi newton :**

Kekurangan dari interpolasi beda terbagi newton ini adalah secara teoritis sedikit lebih rumit dari pada interpolasi lagrange. Sementara itu, kelebihan dari interpolasi beda terbagi newton yaitu secara praktis metode ini cukup mudah dilakukan. Setiap suku memiliki pangkat tertinggi berbeda dan pada metode ini dapat digunakan rumus interpolasi linear dalam menentukan interpolasi kuadrat.

#### **4.3. PERCOBAAN**

Lakukanlah komputasi komputer dengan bahasa pemrograman pascal untuk menentukan pendekatan nilai interpolasi di sebuah titik dari sejumlah data dengan langkah-langkah sebagai berikut.

- Tuliskan kembali algoritma metode penentuan polinom yang melalui sejumlah titik (**metode polinom lagrange, metode beda terbagi newton**) .
- Gambarkan diagram alir dari algoritma yang telah disusun.
- Susunlah bahasa pemrograman berbasis pascal berdasarkan diagram alir yang telah dibuat.
- Jalankan program yang dibuat, dan lakukan pengamatan terhadap hasil dengan macam-macam parameter input.
- Tuliskan hasil percobaan di atas sesuai dengan form laporan percobaan untuk masing-masing metode pendekatan integral dan turunan.

## PRAKTIKUM 5

### REGRESI

#### 5.1. TUJUAN PEMBELAJARAN

Mahasiswa dapat:

1. Menentukan regresi linier dari sejumlah data yang diberikan
2. Menentukan regresi eksponensial dari sejumlah data yang diberikan
3. Menentukan regresi polinom dari sejumlah data yang diberikan.

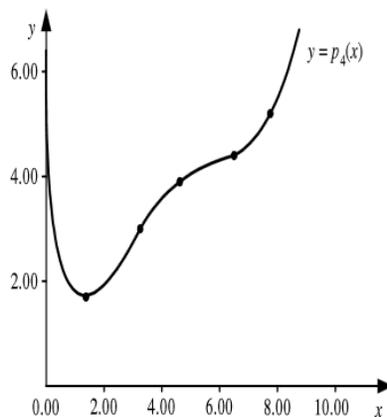
#### 5.2. DASAR TEORI

##### 5.2.1. Regresi

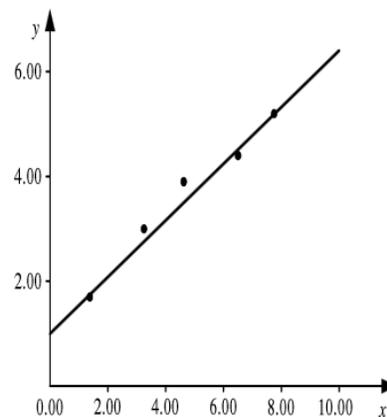
Regresi adalah teknik pencocokan kurva untuk data yang berketelitian rendah. Contoh data yang berketelitian rendah adalah data hasil pengamatan, percobaan di laboratorium, atau data statistik. Data seperti itu kita sebut data hasil pengukuran. Untuk data hasil pengukuran, pencocokan kurva berarti membuat fungsi mengampiri (approximate) titik-titik data. Kurva fungsi hamperan tidak perlu melalui semua titik data tetapi dekat dengannya tanpa perlu menggunakan polinom berderajat tinggi.

Contoh: diberikan data jarak tempuh ( $y$ ) sebuah kendaraan dalam mil setelah  $x$  bulan seperti pada table dibawah ini

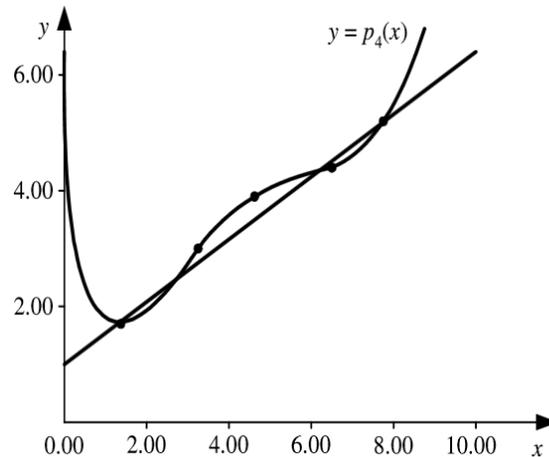
$x$	1.38	3.39	4.75	6.56	7.76
$y$	1.83	2.51	3.65	4.10	5.01



Interpolasi



Regresi



Dari kedua pencocokan tersebut, terlihat bahwa garis lurus memberikan hampiran yang bagus, tetapi belum tentu yang terbaik. Pengertian terbaik disini bergantung pada cara kita mengukur galat hampiran.

Prinsip penting yang harus diketahui dalam mencocokkan kurva untuk data hasil pengukuran adalah:

- ✓ Fungsi mengandung sesedikit mungkin parameter bebas
- ✓ Deviasi fungsi dengan titik data dibuat minimum.

Kedua prinsip diatas mendasari metode regresi kuadrat terkecil. Manfaat pencocokan kurva untuk data hasil pengukuran:

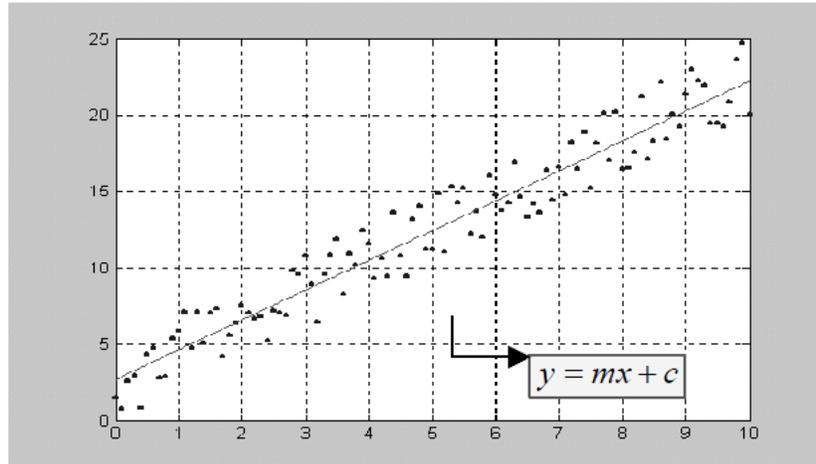
1. Bagi ahli sains/rekayasa: mengembangkan formula empirik untuk system yang diteliti.
2. Bagi ahli ekonomi: menentukan kurva kecenderungan ekonomi untuk “meramalkan” kecenderungan masa depan.

Pada materi ini regresi dapat di bedakan dalam tiga jenis, yaitu:

1. Regresi Linier
2. Regresi Eksponensial, dan
3. Regresi Polinomial

### 5.2.2. Regresi Linier

Regresi linier digunakan untuk menentukan fungsi linier (garis lurus) yang paling sesuai dengan kumpulan titik data hasil pengukuran  $(x_i, y_i)$  yang diketahui.



Garis lurus tersebut dibuat sedemikian sehingga galatnya sekecil mungkin dengan titik-titik data. Karena data mengandung galat, maka nilai data sebenarnya,  $g(x_i)$ , dapat ditulis sebagai

$$g(x_i) = y_i + e_i \quad i = 1, 2, \dots, n$$

yang dalam hal ini,  $e_i$  adalah galat setiap data, sehingga fungsi linier yang diharapkan berbentuk

$$f(x) = a + bx$$

Untuk mencocokkan data sedemikian sehingga deviasinya adalah

$$r_i = y_i - f(x_i) = y_i - (a + bx_i)$$

Total kuadrat deviasinya adalah

$$R = \sum_{i=1}^n r_i^2 = \sum_{i=1}^n (y_i - a - bx_i)^2$$

Agar  $R$  minimum, maka haruslah

$$\frac{\partial R}{\partial a} = -2 \sum (y_i - a - bx_i) = 0$$

$$\frac{\partial R}{\partial b} = -2 \sum x_i (y_i - a - bx_i) = 0$$

Masing-masing ruas kedua persamaaan dibagi dengan  $-2$ , sehingga didapat

$$\sum (y_i - a - bx_i) = 0 \Rightarrow \sum y_i - \sum a - \sum bx_i = 0$$

$$\sum x_i (y_i - a - bx_i) = 0 \Rightarrow \sum x_i y_i - \sum ax_i - \sum bx_i^2 = 0$$

Selanjutnya

$$\sum y_i = \sum a + \sum bx_i$$

$$\sum x_i y_i = \sum ax_i + \sum bx_i^2$$

atau

$$\sum y_i = na + b \sum x_i$$

$$\sum x_i y_i = a \sum x_i + b \sum x_i^2$$

Kedua persamaan terakhir ini dinamakan **persamaan normal**, dan dapat dituliskan dalam bentuk persamaan matriks sebagai berikut.

$$\begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix} = \begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

Solusi untuk nilai a dan b dapat dicari dengan metode eliminasi Gauss atau langsung dengan rumus

$$b = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}$$

$$a = \bar{y} - b\bar{x}$$

Untuk menentukan seberapa bagus fungsi hampiran mencocokkan data, kita dapat mengukurnya dengan **Galat RMS** (*Root-mean-square error*):

$$E_{RMS} = \left( \frac{1}{n} \sum_{i=1}^n |f(x_i) - y_i|^2 \right)^2$$

Semakin kecil nilai  $E_{RMS}$ , semakin bagus fungsi hampiran mencocokkan titik-titik data.

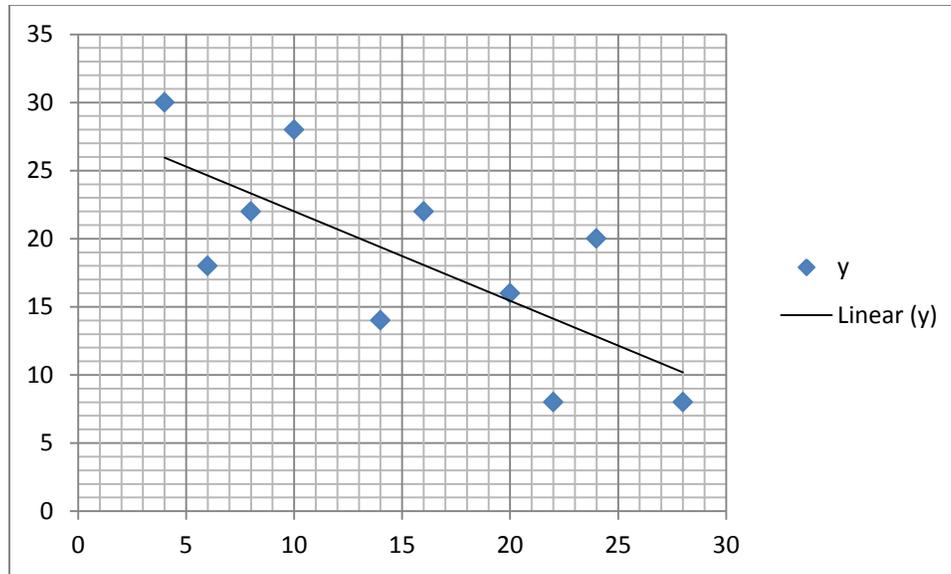
**Contoh :**

Tentukan persamaan garis yang mewakili data berikut:

<b>x</b>	4	6	8	10	14	16	20	22	24	28
<b>y</b>	30	18	22	28	14	22	16	8	20	8

**Penyelesaian :**

Penggambaran titik-titik data pada siste koordinat x-y diberikan dalam Gambar 1, yang dapat diwakili oleh garis lurus.



Gambar.1

$$\bar{x} = \frac{\sum x}{n} = \frac{152}{10} = 15,2$$

$$\bar{y} = \frac{\sum y}{n} = \frac{186}{10} = 18,6$$

Persamaan garis yang mewakili titik-titik data tersebut adalah  $y = a + bx$

No	$x_i$	$y_i$	$x_i y_i$	$x_i^2$
1	4	30	120	16
2	6	18	108	36
3	8	22	176	64
4	10	28	280	100
5	14	14	196	196
6	16	22	352	256
7	20	16	320	400
8	22	8	176	484
9	24	20	480	576
10	28	8	224	784
$\Sigma$	152	186	2432	2912

$$b = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n(\sum x_i^2) - (\sum x_i)^2}$$

$$= \frac{(10 \times 2432) - (152 \times 186)}{(10 \times 2912) - (152)^2} = -\frac{3952}{6016} = -0,6569$$

$$a = \bar{y} - b\bar{x} = 18,6 - (-0,6569 \times 15,2) = 28,5849$$

Jadi persamaan garis adalah

$$y = 28,5849 - 0,6569 x$$

**Algoritma:**

- (1) Tentukan n titik data yang diketahui dalam  $(x_i, y_i)$  untuk  $i = 1, 2, 3, \dots, n$
- (2) Hitung nilai a dan b dengan menggunakan formulasi dari regresi linier di atas
- (3) Tampilkan fungsi linier
- (4) Hitung fungsi linier tersebut dalam range x dan step dx tertentu
- (5) Tampilkan hasil tabel  $(x_n, y_n)$  dari hasil fungsi linier tersebut.

### 5.2.3. Regresi Eksponensial

Regresi eksponensial digunakan menentukan fungsi eksponensial yang paling sesuai dengan kumpulan titik data  $(x_n, y_n)$  yang diketahui. Regresi eksponensial ini merupakan pengembangan dari regresi linier dengan memanfaatkan fungsi logaritma. Perhatikan bahwa

$$y = e^{ax+b}$$

dengan melogaritmakan persamaan di atas akan diperoleh:

$$\ln y = \ln(e^{ax+b})$$

$$\ln y = ax + b$$

atau dapat ditulis

$$z = ax + b \text{ dimana } z = \ln y$$

Dengan demikian dapat digunakan regresi linier dalam menentukan fungsi eksponensial.

**Contoh:**

Carilah persamaan kurva eksponensial jika diketahui data x dan y sebagai berikut

$x_i$	$y_i$	$z_i = \ln y$
1	0,5	0,6931
2	1,7	0,5306
3	3,4	1,2238
4	5,7	1,7405
5	8,4	2,1282

$$\sum x_n = 15$$

$$\sum x_n z_n = 21,6425$$

$$\sum z_n = 4.93 \quad \sum x_n^2 = 55$$

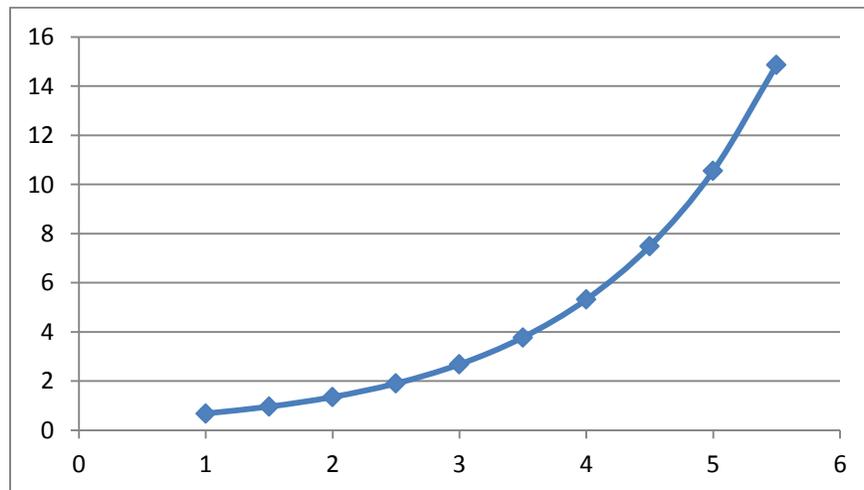
$$a = \frac{5 \times 21,6465 - 15 \times 4,93}{5 \times 55 - 15^2}$$

$$b = \frac{4,93}{5} - 0,685 \times \frac{15}{5} = -1,069$$

$$y = e^{0,685x - 1,069}$$

i	x	y
1	1	0.68113
2	1.5	0.95935
3	2	1.35121
4	2.5	1.90313
5	3	2.68049
6	3.5	3.77538
7	4	5.31748
8	4.5	7.48948
9	5	10.5487
10	5.5	14.8574

Berikut ini gambar grafik eksponensial  $y = e^{0,685x - 1,069}$



### Algoritma Regresi Eksponensial

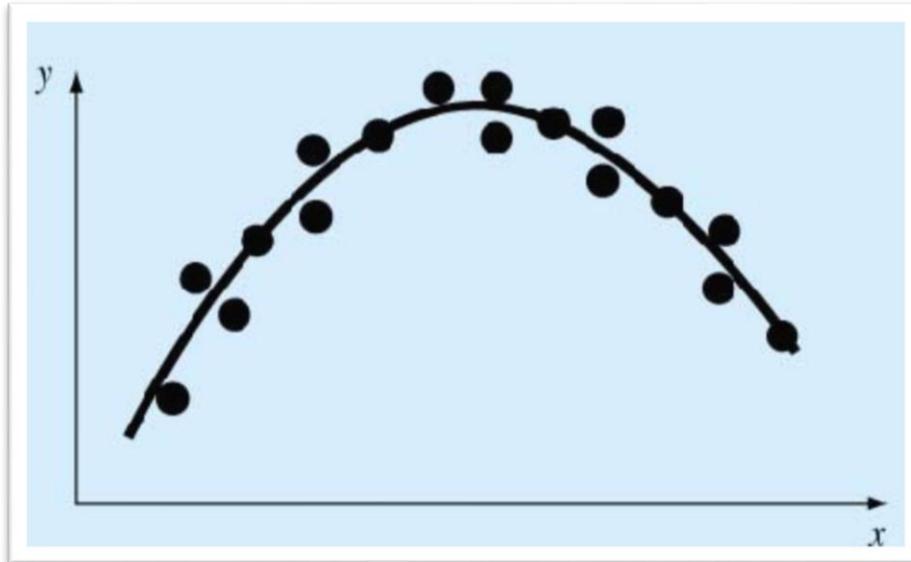
1. Tentukan N titik data yang diketahui dalam  $(x_i, y_i)$  untuk  $i = 1, 2, 3, \dots, N$
2. Ubah nilai y menjadi z dengan  $z = \ln y$
3. Hitung nilai a dan b dengan menggunakan formulasi dari regresi linier di atas
4. Tampilkan fungsi eksponensial  $y = e^{-ax+b}$

5. Hitung fungsi eksponensial tersebut dalam range  $x$  dan step  $dx$  tertentu
6. Tampilkan hasil tabel  $(x_n, y_n)$  dari hasil fungsi eksponensial tersebut.

#### 5.2.4. Regresi Polinomial

Pada pembahasan sebelumnya, regresi garis bekerja dengan baik jika data hasil pengukuran hampir linier. Namun untuk kasus-kasus lain, hasil yang lebih baik dapat diperoleh dengan mencocokkan suatu polinom berderajat lebih tinggi pada himpunan data tersebut. Untuk menangani kasus-kasus yang demikian, maka digunakan regresi polinomial.

Regresi polinomial digunakan menentukan fungsi polinomial yang paling sesuai dengan kumpulan titik data  $(x_n, y_n)$  yang diketahui.



Adapun fungsi pendekatan dari regresi polinomial untuk suatu polinom berderajat  $n$  adalah sebagai berikut :

$$g(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

Simpangan kurva dari masing-masing titik data adalah

$$d_i = y_i - g(x_i), \quad i = 1, 2, \dots, L$$

Jumlah kuadrat simpangan-simpangannya adalah sebagai berikut :

$$D = \sum_{i=1}^L (d_i)^2$$

Kita tetapkan turunan-turunan parsial dari D terhadap koefesien-koefesien polinom sama dengan 0 untuk meminimumkan D.

$$\frac{\partial D}{\partial a_i} = 0, \quad i = 1, 2, \dots, L$$

atau secara setara

$$\sum_{n=0}^N \left( \sum_{i=1}^L x_i^{n+k} \right) a_n = \sum_{i=1}^L x_i^k y_i, \quad \text{untuk } k = 0, 1, \dots, N$$

Yang secara lebih jelas dapat dituliskan dalam bentuk persamaan matriks sebagai berikut.

$$\begin{pmatrix} L & \sum x_i & \dots & \sum x_i^N \\ \sum x_i & \sum x_i^2 & \dots & \sum x_i^{N+1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \sum x_i^N & \sum x_i^{N+1} & \dots & \sum x_i^{2N} \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_N \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum x_i y_i \\ \cdot \\ \cdot \\ \cdot \\ \sum x_i^N y_i \end{pmatrix}$$

Koefesien-koefesien  $a_i, i = 0, 1, 2, \dots, N$  ditentukan dengan metode eliminasi Gauss.

### Contoh:

Lakukan pencocokan data hasil pengukuran seperti terlihat pada tabel 6.6 kepada polinomial kuadrat.

### Penyelesaian

Persamaan simultan untuk menemukan harga koefisien-koefisien kuadrat dalam masalah ini secara umum mengambil bentuk

$$\begin{bmatrix} N & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}$$

Untuk masing-masing elemen matriks persamaan simultan di atas dapat dilihat pada tabel berikut.

$i$	$x_i$	$x_i^2$	$x_i^3$	$x_i^4$	$y_i$	$x_i y_i$	$x_i^2 y_i$
1.	0.1400	0.0196	0.0027	0.000	4.0964	0.5735	<b>0.0803</b>

2.	0.4300	0.1849	0.0795	0.034	4.7284	2.0332	<b>0.8743</b>
3.	0.5800	0.3364	0.1951	0.113	5.2231	3.0294	<b>1.7571</b>
4.	0.9100	0.8281	0.7536	0.685	5.9984	5.4585	<b>4.9673</b>
5.	1.3000	1.6900	2.1970	2.900	6.8989	8.9686	<b>11.6591</b>
6.	2.0000	4.0000	8.0000	16.400	7.2307	14.4614	<b>28.9228</b>
7.	2.2000	4.8400	10.6480	23.400	7.3306	16.1273	<b>35.4801</b>
8.	2.5000	6.2500	15.6250	39.100	7.8756	19.6890	<b>49.2225</b>
9.	2.7000	7.2900	19.6830	53.100	7.9908	21.5752	<b>58.2529</b>
10.	3.2000	10.2400	32.7680	104.900	8.1303	26.0170	<b>83.2543</b>
11.	3.5000	12.2500	42.8750	150.100	8.4302	29.5057	<b>103.2699</b>
12.	4.1000	16.8100	68.9210	282.600	8.5444	35.0320	<b>143.6314</b>
13.	4.4000	19.3600	85.1840	374.800	8.8931	39.1296	<b>172.1704</b>
14.	4.9000	24.0100	117.6490	576.500	9.0432	44.3117	<b>217.1272</b>
15.	6.3000	39.6900	250.0470	1575.300	9.3240	58.7412	<b>370.0696</b>
<b>Jmlh</b>	<b>39.1600</b>	<b>147.7990</b>	<b>654.6279</b>	<b>3199.000</b>	<b>109.7381</b>	<b>324.6533</b>	<b>1280.007</b>

Secara eksplisit, persamaan linier simultan untuk menemukan  $a_0, a_1, a_2$  selanjutnya dapat ditampilkan dalam bentuk matriks yaitu

$$\begin{bmatrix} 15.0000 & 39.1600 & 147.7990 \\ 39.1600 & 147.7990 & 654.6279 \\ 147.7990 & 654.6279 & 3199.000 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 109.7381 \\ 324.6533 \\ 1280.007 \end{bmatrix}$$

Dengan menyelesaikannya menggunakan metode eliminasi Gauss atau Gauss-Jordan, maka diperoleh harga-harga

$$a_0 = 4.1952$$

$$a_1 = 1.8301$$

$$a_2 = -0.1682$$

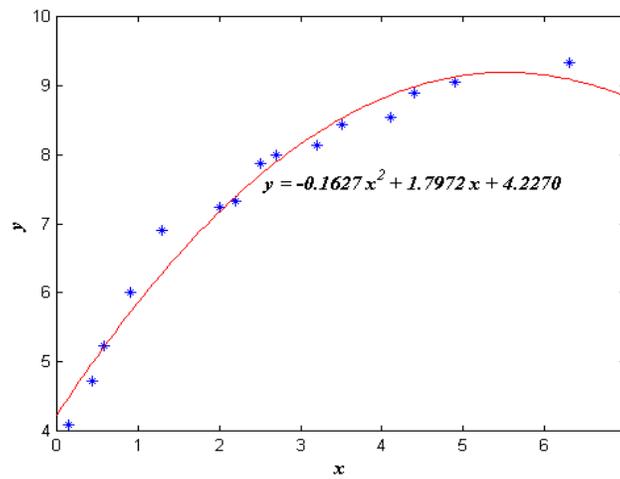
Polinomial kuadratik hasil pencocokan selanjutnya dapat dinyatakan sebagai

$$y = -0.1682 x^2 + 1.8301x + 4.1952$$

seperti diperlihatkan pada gambar.

$i$	$x_i$	$y_i$	Polinomial	Simpangan
1.	0.1400	4.0964	4.4754	<b>-0.3790</b>
2.	0.4300	4.7284	4.9697	<b>-0.2413</b>
3.	0.5800	5.2231	5.2146	<b>0.0085</b>
4.	0.9100	5.9984	5.7277	<b>0.2707</b>
5.	1.3000	6.8989	6.2884	<b>0.6105</b>
6.	2.0000	7.2307	7.1706	<b>0.0601</b>
7.	2.2000	7.3306	7.3934	<b>-0.0628</b>
8.	2.5000	7.8756	7.7032	<b>0.1724</b>
9.	2.7000	7.9908	7.8934	<b>0.0974</b>
10.	3.2000	8.1303	8.3121	<b>-0.1818</b>
11.	3.5000	8.4302	8.5242	<b>-0.0940</b>
12.	4.1000	8.5444	8.8607	<b>-0.3163</b>
13.	4.4000	8.8931	8.9850	<b>-0.0919</b>

14.	4.9000	9.0432	9.1271	<b>-0.0839</b>
15.	<b>6.3000</b>	<b>9.3240</b>	<b>9.0923</b>	<b>0.2317</b>



### Algoritma

- (1) Tentukan N titik data yang diketahui dalam  $(x_i, y_i)$  untuk  $i=1,2,3,\dots,N$
- (2) Hitung nilai-nilai yang berhubungan dengan jumlahan data untuk mengisi matrik normal
- (3) Hitung nilai koefisien-koefisien  $a_0, a_1, a_2, \dots, a_n$  dengan menggunakan eliminasi gauss/jordan
- (4) Tampilkan fungsi polinomial  $g(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$
- (5) Hitung fungsi polinomial tersebut dalam range  $x$  dan step  $dx$  tertentu
- (6) Tampilkan hasil tabel  $(x_n, y_n)$  dari hasil fungsi polinomial tersebut.

### 5.3. PERCOBAAN

Lakukanlah komputasi komputer dengan bahasa pemrograman pascal untuk menentukan regresi linier, eksponensial dan regresi polinom dari sejumlah data dengan langkah-langkah sebagai berikut.

- a. Tuliskan kembali algoritma metode regresi (**metode regresi linier, metode regresi eksponensial, metode regresi polinom**) yang akan digunakan.
- b. Gambarkan diagram alir dari algoritma yang telah disusun.
- c. Susunlah bahasa pemrograman berbasis pascal berdasarkan diagram alir yang telah dibuat.

- d. Jalankan program yang dibuat, dan lakukan pengamatan terhadap hasil dengan macam-macam parameter input.
  - e. Tuliskan hasil percobaan di atas sesuai dengan form laporan percobaan untuk masing-masing metode pendekatan integral dan turunan.
-