# Pendahuluan Metode Numerik

1

## Obyektif :

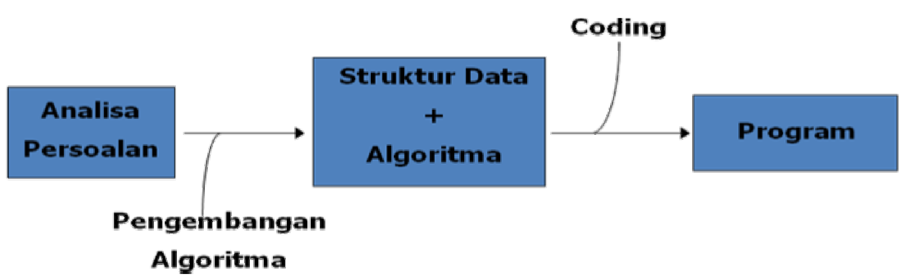
1. Mengerti Penggunaan metode numerik dalam penyelesaian masalah.
2. Mengerti dan memahami penyelesaian masalah menggunakan grafik maupun metode numeric.

### Pendahuluan Metode Numerik (Algoritma)

Salah satu topic penting dalam studi computer adalah topic mengenai Metode Numerik. Metode Numerik juga sangat mendukung berbagai bidang serta disiplin lain, terutama dalam menyelesaikan berbagai komputasi.

Dalam bidang teknologi misalnya, berbagai model matematika seperti Model Persamaan Nonlinier, Model Matriks dan Persamaan Linier, Integral dan juga Persamaan Diferensial sangat sering muncul, dan harus dilakukan komputasi terhadap mereka. Tinggal kita memakainya secara efektif serta efisien.

***Pemrograman Terstruktur***




***Analisa Persoalan***

Persoalan apa yang perlu diselesaikan =>tujuan/program.

Bagaimana program harus berbuat untuk menyelesaikan persoalan.

***Struktur Data***

Bagaimana menyusun data yang ada (struktur data) sehingga program menjadi efisien :

- mudah dalam pengambilan data.
- mudah dalam memperbarui data (insert, delete, update).
- mudah algoritmanya.

***Garis Besar***

Tahap struktur data & algoritma :

- Rancang struktur data yang digunakan, sedemikian rupa sehingga mudah merancang algoritmanya dan memberikan proses yang efisien.
- Dengan memahami masalah yang dimiliki, rancang setiap langkah yang diperlukan untuk menyelesaikan (bisa dalam bentuk flowchart ataupun pseudocode) dan pilih teknik-teknik yang efisien => design algorithm.
- Mencoba algoritma yang dibuat dengan data sederhana.

***Tahap Program :***

- Memahami syntax dari bahasa yang akan digunakan.
- Inti : INPUT => PROSES => OUTPUT.
- Coding : susun pengkodean-nya sesuai dengan algoritma dan struktur data yang telah dibuat beserta Keterangannya.
- Debugging : memperbaiki kesalahan syntax (tahap compiling), kesalahan logika dan kesalahan run-time (tahap execution).
- Testing : menguji dengan data yang telah diketahui hasil outputnya.
- Dokumentasi.

***Di dalam menyusun algoritma, hanya dikenal 3 jenis struktur proses :***

| | | |
|---|---|---|
| 1. Struktur sederhana (sequence) | : | Perintah-perintah dilakukan secara berurutan |
| 2. Struktur berulang (repetition/loop) | : | Satu/sekumpulan perintah dilakukan berulang-ulang |
| 3. Struktur bersyarat | | Satu/sekumpulan perintah dilakukan jika suatu kondisi dipenuhi |

Algoritma yang disusun berdasarkan struktur di atas disebut : algoritma terstruktur, sedangkan program yang dibuat berdasarkan algoritma terstruktur dikatakan sebagai program terstruktur

***Alasan menggunakan program terstruktur :***

- Jika kita telah terbiasa menganalisa dan menyusun program untuk suatu masalah dengan menggunakan teknik yang sama, maka pemecahan masalah (analisa, design, penyusunan program) akan menjadi lebih cepat dan mengurangi error.
- Jika persoalan dilakukan oleh suatu team work, dan semua programer menggunakan teknik yang sama, maka akan lebih mudah bagi seorang programmer untuk dapat membaca pekerjaan programmer yang laindan mudah dalam pengembangan /perbaikan program nantinya.

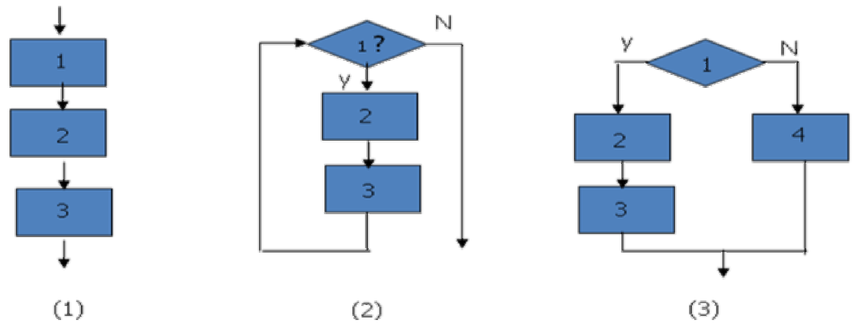
***Struktur Logika (3 struktur kendali dasar)***

Untuk menghindari terjadinya “spaghetti code” yang menyebabkan suatu program sulit untuk ditelusuri baik dalam penulisan maupun debugging maka digunakanlah 3 struktur kendali dasar yang dapat mengatur jalannya program :

- Struktur Sederhana (berurutan) : jika perintah-perintah dilakukan secara berurutan.
- Struktur Berulang (loop) : jika sekumpulan perintah diulang beberapa kali.

- Struktur Bersyarat (selection) : jika sekumpulan perintah dilakukan pada kondisi tertentu.

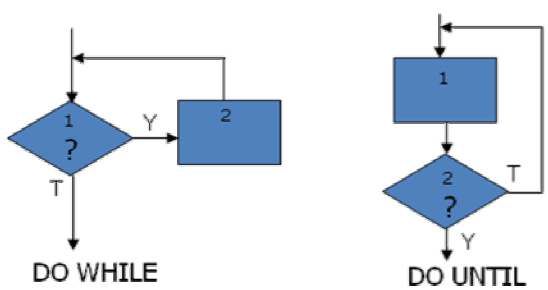
***Struktur Logika***




Bentuk umum dari 3 struktur kendali proses

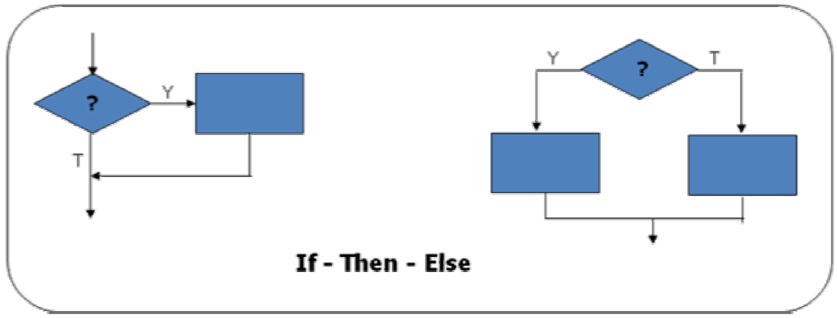
a. Terdapat tiga macam struktur berulang :

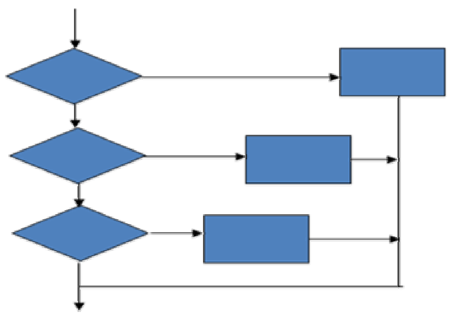
- FOR LOOP : badan loop dilakukan untuk sejumlah tertentu pengulangan.
- DO WHILE : lakukan pengujian, selama kondisi masih dipenuhi, lakukan badan loop.
- DO UNTIL : lakukan badan loop satu kali, uji kondisi dan lakukan badan loop sampai kondisi tidak dipenuhi.



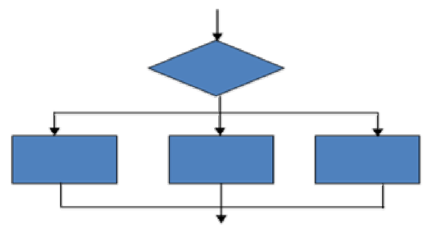

b. Struktur Bersyarat dapat dibedakan :

- Struktur 2 pilihan : IF – THEN – ELSE.
- Struktur lebih dari 2 pilihan : NESTED IF/ DO CASE.



Nested If



Do Case

## Pengertian Metode Numerik

### *Metode Analitik vs Metode Numerik*

Metode analitik metode sebenarnya dapat memberikan solusi sebenarnya (exact solution) solusi yang memiliki galat/error = 0.

Metode analitik hanya unggul pada sejumlah persoalan matematika yang terbatas.

Metode numeric teknik yang digunakan untuk memformulasikan persoalan matematik sehingga dapat dipecahkan dengan operasi hitungan / aritmatika biasa. Solusi angka yang didapatkan dari metode numeric adalah solusi yang mendekati nilai sebenarnya / solusi pendekatan (approximation) dengan tingkat ketelitian yang kita inginkan. Karena tidak tepat sama dengan solusi sebenarnya, ada selisih diantara keduanya yang kemudian disebut galat/error.

Metode numeric dapat menyelesaikan persoalan didunia nyata yang sering kali non linier, dalam bentuk dan proses yang sulit diselesaikan dengan metode analitik.

### *Mengapa Menggunakan Metode Numerik*

Tidak semua permasalahan matematis atau perhitungan dapat diselesaikan dengan mudah.

Dibutuhkan metode yang menggunakan analisis-analisis pendekatan persoalan2 non linier untuk menghasilkan nilai yang diharapkan.

Kesulitan menggunakan metode analitik untuk mencari solusi exact dengan jumlah data yang besar, diperlukan perhitungan komputer,-metode numerik menjadi penting untuk menyelesaikan permasalahan ini.

Pemakaian metode analitik terkadang sulit diterjemahkan kedalam algoritma yang dapat dimengerti oleh komputer. Metode numerik yang memang berangkat dari pemakaian alat bantu hitung merupakan alternative yang baik dalam menyelesaian persoalan-persoalan perhittungan yang rumit.

## Pendekatan dan Kesalahan

### *Beberapa criteria penyelesaian perhitungan matematika*

Bila persoalan merupakan persoalan yang sederhana atau ada theorem analisa matematika yang dapat digunakan untuk menyelesaikan persoalan tersebut, maka penyelesaian matematis (metode analitik) adalah penyelesaian exact yang harus digunakan. Penyelesaian ini menjadi acuan bagi pemakaian metode pendekatan.

Bila persoalan sudah sangat sulit atau tidak mungkin diselesaikan secara matematis (analitik) karena tidak ada theorem analisa matematik yang dapat digunakan, maka dapat digunakan metode numerik.

Bila persoalan sudah merupakan persoalan yang mempunyai kompleksitas tinggi, sehingga metode numerik pun tidak dapat menyajikan penyelesaian dengan baik, maka dapat digunakan metode-metode simulasi.

### *Prinsip-Prinsip Metode Numerik*

Metode numerik kini disajikan dalam bentuk algoritma-algoritma yang dapat dihitung secara cepat dan mudah.

Pendekatan yang digunakan dalam metode numerik merupakan pendekatan analisis matematis, dengan tambahan grafis dan teknik perhitungan yang mudah.

Algoritma pada metode numerik adalah algoritma pendekatan maka dalam algoritma tersebut akan muncul istilah *iterasi* yaitu pengulangan proses perhitungan.

Dengan metode pendekatan, tentunya setiap nilai hasil perhitungan akan mempunyai *nilai error* (nilai kesalahan).

# PERSAMAAN NON LINIER

2

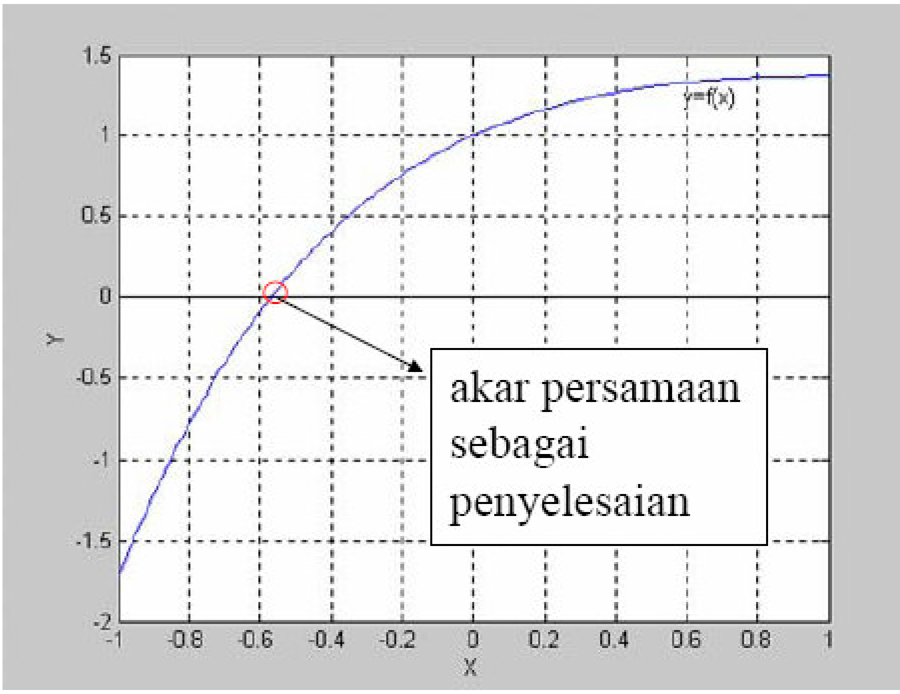
**Obyektif :**

1. Mengerti penggunaan solusi persamaan non linier
2. Mengerti metode biseksi dan regulafalsi
3. Mampu menggunakan metode biseksi dan regula falsi untuk mencari solusi

## PENGANTAR PERSAMAAN NON LINIER

Beberapa contoh permasalahan yang memerlukan penyelesaian persamaan non linier sebagai kuncinya adalah sebagai berikut:

- Penentuan nilai maksimal dan minimal fungsi non linier
- Perhitungan nilai konstanta pada matrik dan determinan, yang biasanya muncul dalam permasalahan sistem linier, Bisa digunakan untuk menghitung nilai eigen
- Penentuan titik potong beberapa fungsinon linier, yang banyak digunakan untuk keperluan perhitungan-perhitungan secara grafis.
- Penyelesaian persamaan non linier adalah penentuan akar – akar persamaan non linier.
- Akar sebuah persamaan f(x)=0 adalah nilai – nilai x yang menyebabkan nilai f(x) sama dengan nol.
- Akar persamaan f(x) adalah titik potong antara kurva f(x) dan sumbu X.



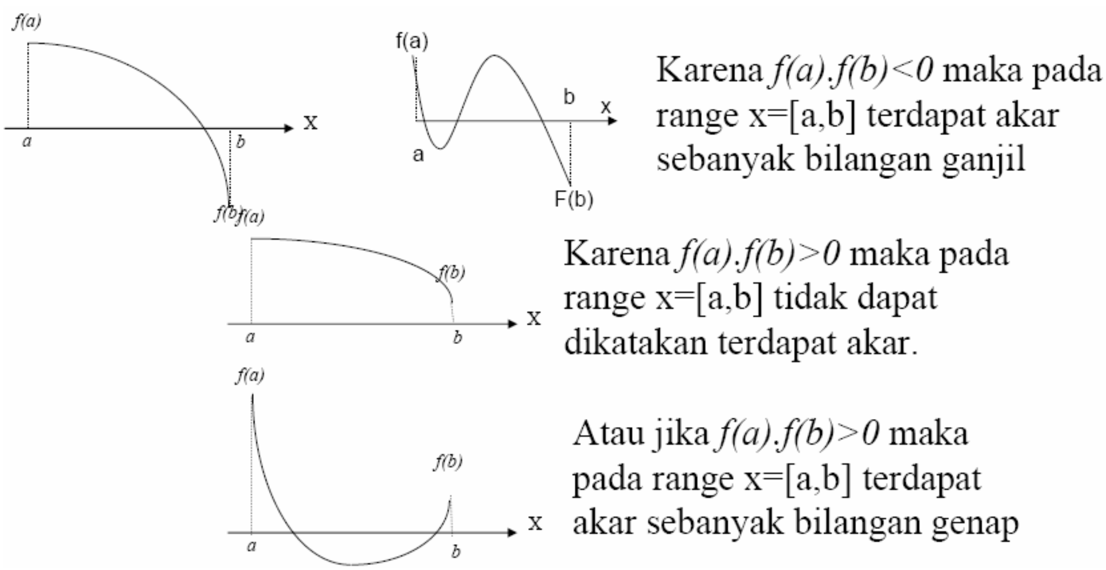

Contoh kurva $y = xe^{-x} + 1$

Titik potong kurva dengan sb x ada diantara x = - 0.5 dan x = -0.6 sehingga akar atau penyelesaian persamaan $y = xe^{-x} + 1$ juga berada di x = -0.5 dan x = -0.6

## Teorema

## PenyelesaianPersamaanNon Linier

*Suatu range x=[a,b] mempunyai akar bila f(a) dan f(b) berlawanan tanda atau memenuhi f(a).f(b)<0.*



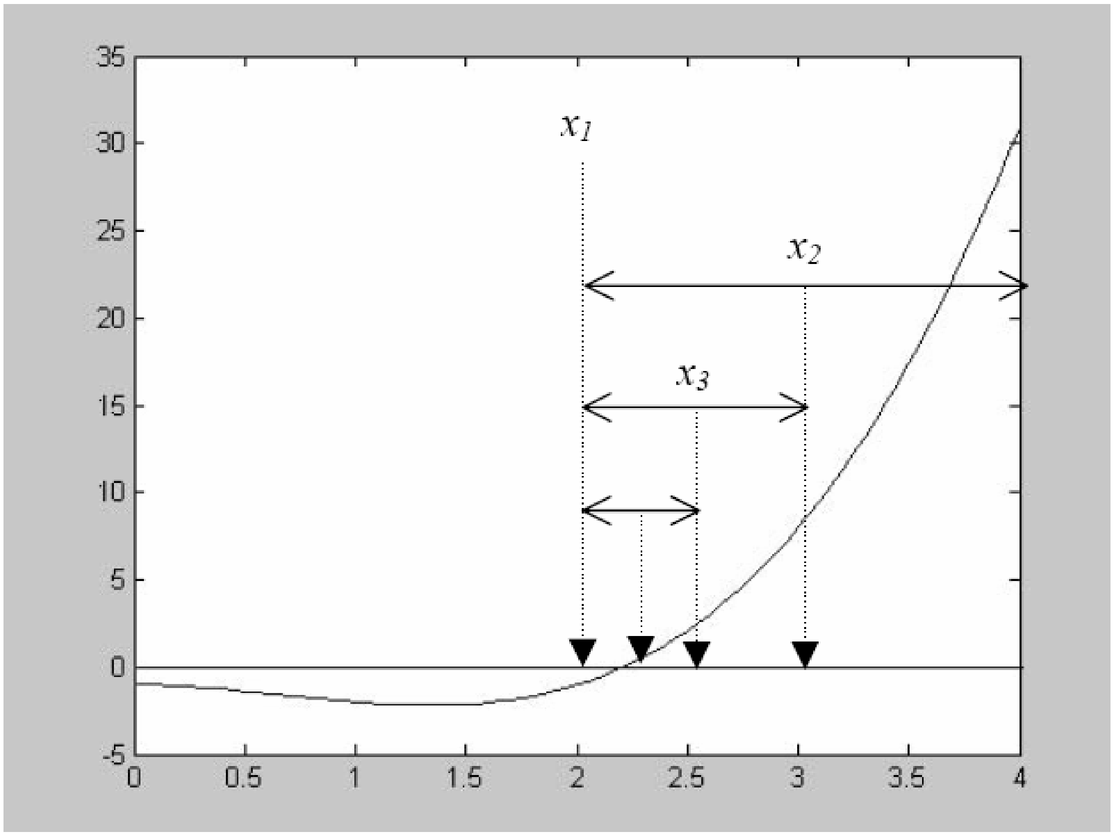

## 1. Metode Biseksi

- Metode biseksi ini membagi range menjadi 2 bagian, dari dua bagian ini dipilih mana yang mengandung dan bagian yang tidak mengandung akar dibuang. Hal ini dilakukan berulang - ulang hingga diperoleh akar persamaan.
- Untuk menggunakan metode biseksi, tentukan batas bawah (a) dan batas atas (b). Kemudian dihitung nilai tengah : x = (a +b )/2
- Dari nilai x ini perlu dilakukan pengecekan keberadaan akar :

  f(a). F(b) < 0, maka b=x, f(b)=f(x), a tetap

  f(a). F(b) > 0, maka a=x, f(a)=f(x), b tetap

- Setelah diketahui dibagian mana terdapat akar, maka batas bawah & batas atas di perbarui sesuai dengan range dari bagian yang mempunyai akar.




Metode Biseksi

**Algoritma Metode Biseksi :**

1. Definisikan fungsi *f(x)* yang akan dicari akarnya
2. Tentukan nilai *a* dan *b*
3. Tentukan torelansi e dan iterasi maksimum *N*
4. Hitung *f(a)* dan *f(b)*
5. Jika *f(a).f(b)>0* maka proses dihentikan karena tidak ada akar, bila tidak dilanjutkan
6. Hitung x= $\frac{a+b}{2}$ Hitung *f(x)*
7. Bila *f(x).f(a)<0* maka *b=x* dan *f(b)=f(x)*, bila tidak *a=x* dan *f(a)=f(x)*
8. Jika |b-a|<e atau iterasi>iterasi maksimum maka proses dihentikan dan didapatkan akar = x, dan bila tidak, ulangi langkah 6.

**ALGORITMA BISEKSI (1)**

```
INPUT X0 ,X1 ,F(X),T

WHILE [(X1 – X0) ≥T OR F(X0)*F(X1) ≠0] DO

        X2 = (X0 + X1)/2

        IF F(X0)*F(X2) >0 THEN

             X0 = X2

        ELSE

             X1 = X2

        ENDIF

ENDWHILE
```

**ALGORITMA BISEKSI (2)**

```
IF F(X0)=0 THEN
           OUTPUT (X0)
ELSE IF F(X1) = 0 THEN
            OUTPUT (X1)
ELSE
            OUTPUT (X2)
ENDIF
```

**KEUNTUNGAN BISEKSI**

- Selalu berhasil menemukan akar (solusi) yang dicari, atau dengan kata lain **selalu konvergen**.

**KELEMAHAN BISEKSI**

- Bekerja **sangat lambat**. Tidak memandang bahwa sebenarnya akar atau solusi yang dicari telah berada dekat sekali dengan $X_0$ ataupun $X_1$.
- Metode biseksi hanya dapat dilakukan apabila ada akar persamaan pada interval yang diberikan
- Jika ada beberapa akar pada interval yang diberikan maka hanya satu akar saja yang dapat ditemukan.
- Memiliki proses iterasi yang banyak sehingga memperlama proses penyelesaiannya

## 2. METODE REGULAFALSI

Metode regulafalsi adalah metode pencarian akar persamaan dengan memanfaatkan kemiringan dan selisih tinggi dari dua titik batas range.

Metode ini bekerja secara iterasi dengan melakukan update range. Titik pendekatan yang dipakai adalah

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

Prinsip dari metode ini didasarkan pada interpolasi linier. Perbedaannya dengan metode bagi dua terletak pada pencarian akar persamaan setelah akar tersebut dikurung oleh dua harga taksiran awal. Selanjutnya dilakukan interpolasi linier pada ujung-ujung titik untuk memperoleh pendekatan harga akar. Jadi, jika fungsi tersebut dapat didekati dengan cara interpolasi linier, maka akar-akar taksiran tersebut memiliki ketelitian yang tinggi, akibatnya iterasi dapat mencapai konvergensi ke arah harga akar pendekatan dengan cepat.

Penetapan interval baru:

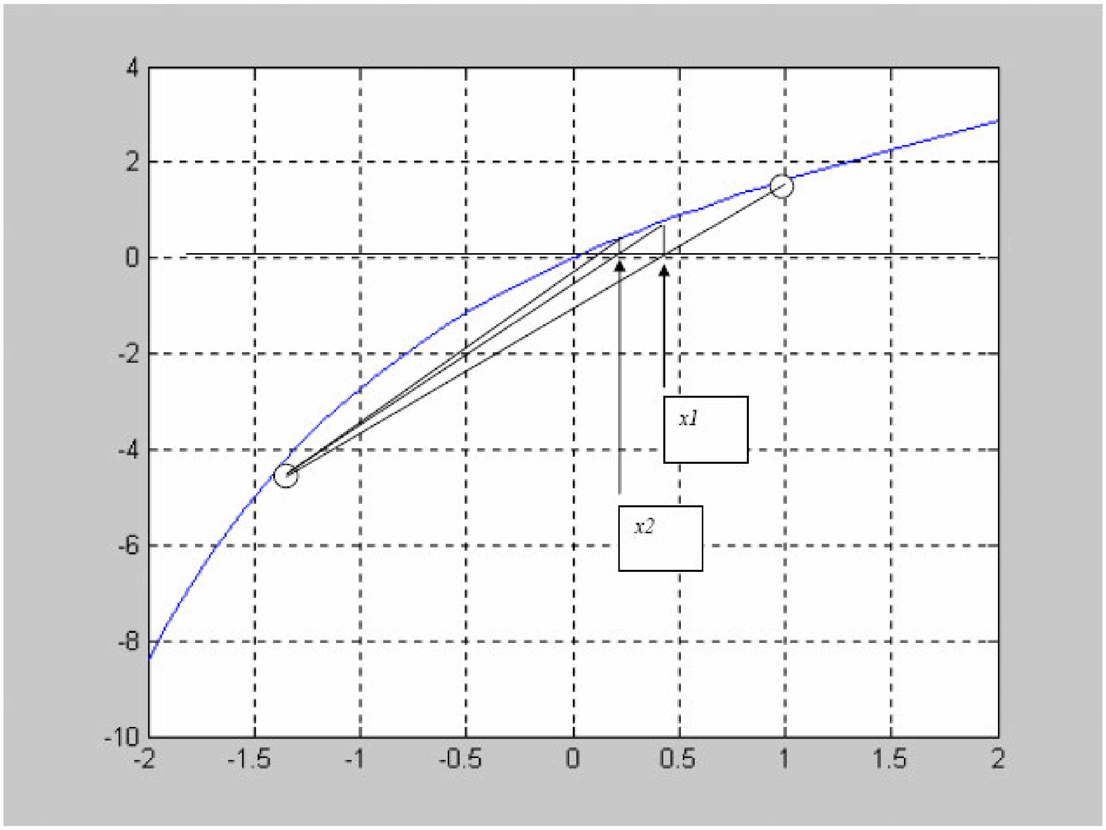
bila $F(X_0)*F(X_2) <0$ maka intervalnya menjadi $[X_0 , X_2]$

bila $F(X_0)*F(X_2) >0$ maka intervalnya menjadi $[X_2 , X_1]$

Pengulangan/iterasi mencari X2 dan interval baru dilakukan berdasarkan nilai toleransi atau bila akarnya belum ditemukan

Sebaiknya nilai toleransi secara relatif mengacu pada : error aproksimasi

**GRAFIK METODE REGULAFALSI**



## Algoritma Metode Regula Falsi

1. Definisikan fungsi f(x)
2. Tentukan batas bawah (a) dan batas atas (b)
3. Tentukan toleransi error (e) dan iterasi maksimum (n)
4. Hitung fa = fungsi(a) dan fb = fungsi(b)
5. Untuk iterasi I = 1 s/d n atau error > e
6. x = $\dfrac{fb.a - fa.b}{fb - fa}$ Hitung fx = fungsi(x)
7. Hitung error = |fx|
8. Jika fx.fa <0 maka b = x dan fb = fx, jika tidak a = x dan fa = fx.
9. Akar persamaan adalah x.

**ALGORITMA REGULAFALSI**

INPUT $X_0$,$X_1$,T,F(X), MAX

I=0; FOUND = false

REPEAT

I=I+1

$X_2 = X_1 - (X_1 - X_0) * F(X_1) / (F(X_1) - F(X_0))$

IF $F(X_0) * F(X_2) < 0$ THEN

$X_1 = X_2$

ELSE

$X_0 = X_2$

ENDIF

IF (|($X_2$ - $X_1$)/ $X_1$|≤T OR I=MAX) THEN

FOUND=true

ENDIF

UNTIL (FOUND=true)

OUTPUT ($X_2$)

Metode regulafalsi hanya membutuhkan kurang dari setengah metode biseksi

**KEUNTUNGAN REGULAFALSI**

Selalu berhasil menemukan akar (solusi) yang dicari, atau dengan kata lain selalu konvergen

**KELEMAHAN REGULAFALSI**

- Hanya salah satu titik ujung interval ($X_0$ atau $X_1$) yang bergerak menuju akar dan yang lainnya selalu tetap untuk setiap iterasi.
- Sehingga mungkin [$X_0$, $X_1$] masih cukup besar jaraknya bila menggunakan batas | $X_1$ - $X_0$| ≤ T padahal $X_0 \rightarrow X_2$ atau
  $X_1 \rightarrow X_2$
- hal tersebut dikenal dengan pendekatan error mutlak.
  ⇒diperbaiki dengan pendekatan Error relatif :

$$\left|\frac{x_1 - x_2}{x_1}\right| \leq T \qquad \text{atau} \qquad \left|\frac{x_0 - x_2}{x_0}\right| \leq T$$

# PERSAMAAN NON LINIER (Lanjutan)

3

**Obyektif :**

1. Mengerti penggunaan solusi non linier

2. Mengerti metode sekan dan iterasi titik tetap

3. Mampu menggunakan metode sekan dan iterasi titik tetap untuk mencari solusi

---

## A. METODE SEKAN

Disebut juga Metode Interpolasi Linear

Dalam prosesnya tidak dilakukan penjepitan akar atau dpl.

[$X_0$, $X_1$] tidak harus mengandung akar yang akan dicari.

Sehingga $f(x_0)$ dan $f(x_1)$ bisa bertanda sama

Untuk mencari $X_2$ , sama dengan metode REGULA FALSI

Untuk iterasi berikutnya akan diperoleh interval baru [$X_0$, $X_1$] dengan cara pergeseran:

$X_0$ menjadi $X_1$ , $X_1$ menjadi$X_2$

Iterasi berlangsung sampai batas maksimum (Max.) atau sampai dipenuhinya batas Toleransi (T):

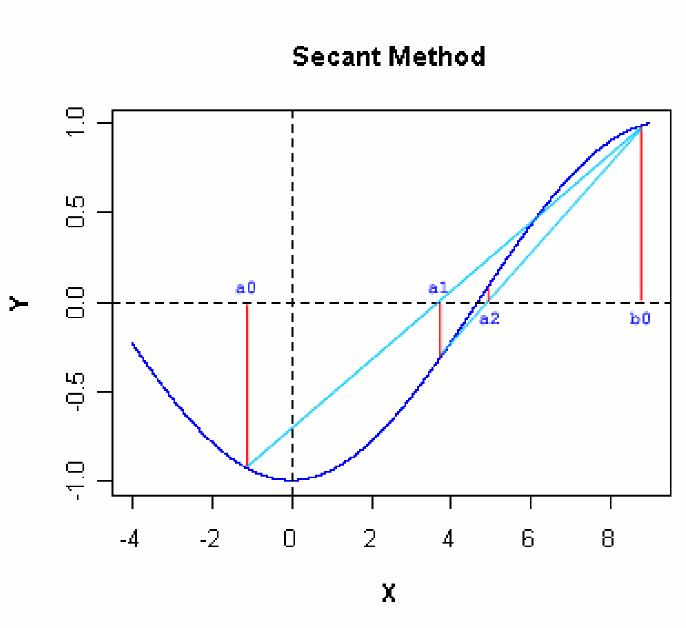
$| (X_1 - X_2 )/ X_1 | \leq T$

Metode ini juga pengembangan dari metode Interpolasi Linier. Metode ini dapat disebut metode Ekstrapolasi Linier. Pada metode ini fungsi $f(x_1)$ tidak perlu berlawanan tanda dengan $f(x_2)$, namun dipilih dua harga yang dekat dengan akar sebenarnya yang ditunjukkan oleh fungsi dari kedua titik tersebut. Algoritma dari metode ini adalah :

1. Memilih harga pendekatan awal, $x_1$ dan $x_2$.
2. Menentukan harga $x_3 = x_2 - (x_2 - x_1) \cdot \dfrac{f(x_2)}{(f(x_2) - f(x_1))}$

3. Jika $|f(x_3)| \leq$ toleransi, maka harga $x_3$ adalah harga x yang dicari, bila tidak dilanjutkan ke tahap 4.
4. Jika $|f(x_1)| > |f(x_2)|$, maka $x_1^{baru} = x_2$, jika tidak maka $x_1^{baru} = x_1$. Kemudian menentukan harga $x_2^{baru} = x_3$, dan kembali ke tahap 2.

Kelebihan Metode Sekan adalah dapat digunakan untuk mencari akar- akar persamaan dari persamaan polinomial kompleks, atau persamaan yang turunan pertamanya sangat sulit didapatkan.




**Contoh:** Selesaikan f(x) = x^3 + x^2 – 3x – 3 = 0 dengan metode secant.

**Penyelesaian:** Dipilih tebakan awal x(1) = 1 dan x(2) = 2;

Untuk x(1) =1, maka f(x(1)) = -4

Untuk x(2) = 2; maka f(x(2)) = 3

Dengan menggunakan (18) maka

Iterasi-1:

$$x(3) = x(2) - \frac{f(x(2))(x(2)-x(1))}{f(x(2))-f(x(1))} = 2 - \frac{3(2-1)}{3-(-4)} = 1.571429$$

Iterasi-2:

Untuk x(2) = 2, maka f(x(2)) = 3

Untuk x(3) = 1.571429, maka f(x(3)) = -1.364431

$$x( \;= x(3) - \frac{f(x(3))(x(3)-x(2)}{f(x(3))-f(x(2))} = 1.571429 - \frac{-1.364431(1.571429-2)}{-1.364431-3} = 1.735136$$

Iterasi-3

Untuk x(3) = 1.571429, maka f(x(3)) = -1.364431

Untuk x(4) = 1.735136, maka f(x(4)) = -0.247745

$$x(5) = x(4) - \frac{f(x(4))(x(4)-x(3)}{f(x(4))-f(x(3))} = 1.735136 - \frac{-1.364431(1.735136-1.571429)}{-0.247745-(-1.364431)} =1.735136$$

Sampai iterasi ke tiga, x(n+1) = 1.735136, untuk n = 4, sehingga dapat disimpulkan hampiran akar pada iterasi ke tiga adalah 1.735136.

**B. METODE ITERASI TITIK TETAP**

**Syaratnya:**

f(x) = 0 dapat diubah menjadi bentuk:

x = g(x) (yang tidak unik)

Cari akar dgn pertidaksamaan rekurens:

$X_{k+1} = g(X_k)$; untuk k = 0, 1, 2, 3, …

dgn $X_0$ asumsi awalnya, sehingga diperoleh

barisan $X_0$, $X_1$, $X_2$, $X_3$, … yang diharapkan

konvergen ke akarnya.

Jika $g'(x)$ ε [a, b] dan $g'(x) \le k$ dgn $k< 1$ Utk setiap

x ε [a, b] , maka titik tetap tersebut tunggal dan

iterasinya akan konvergen menuju akar

Dari bentuk x = g(x), berarti akar dari f(x) tak lain adalah perpotongan antara garis lurus y = x dan kurva y = g(x).

Langkah pertama yang harus dilakukan metode ini adalah mentransformasikan persamaan $f(x) = 0$ secara aljabar ke bentuk $x = g(x)$. Sehingga langkah-langkah iterasi metode ini dapat mengacu hasil transformasi tersebut, yaitu: $x_{n+1} = g(x_n)$(10) dengan g adalah fungsi yang diperoleh dari hasil transformasi $x = g(x)$. Dari (10) dapat dikatakan bahwa suatu selesaian dari bentuk tersebut disebut suatu titik tetap dari g.

Untuk suatu persamaan $f(x) = 0$, mungkin dapat ditransformasikan ke beberapa bentuk $x = g(x)$. Khusus untuk kekonvergensia, barisan iterasi: $x_0, x_1, x_2, \ldots$ (11)

akan berbeda satu sama lainnya, tergantung pada pemilihan $x_0$. Untuk mengetahui iterasi konvergen atau tidak, berikut ini teoremanya.

**Teorema:** Andaikan $x = s$ adalah suatu selesaian $x = g(x)$, dan andaikan $g$ mempunyai turunan kontinu dalam interval I yang memuat $a$, jika $|g'(x)|$ £ $K< 1$ dalam interval $I$, maka proses iterasi yang didefinisikan oleh (11) konvergen untuk sebarang nilai $x_0$ dalam interval I.

**Bukti:** Menurut teorema nilai rata-rata, terdapat suatu t antara x dan s sedemikian sehingga:

$g(x) - g(s) = g'(t)\ (x - s)$ $x$ dan $s$ dalam interval $I$(12)

Karena s suatu selesaian $x = g(x)$, maka $s = g(s)$, dan $x_1 = g(x_0)$, $x_2 = g(x_1)$, …, sehingga diperoleh:

$|x_n - s| = |g(x_{n-1}) - g(s)| = |g'(t)|\ |x_{n-1} - s|$

£ $K\ |x_{n-1} - s| = K\ |g(x_{n-2}) - g(s)| = K\ |g'(t)\ (x_{n-2} - s)|$

£ $K^2 |x_{n-2} – s| = K^2 |g(x_{n-3})– g(s)| = K |g'(t) (x_{n-3} - s)|$

*dst.*

£ $K^n |x_0 – s|$

Karena $K < 1$, maka $K^n$ ® $0$, sehingga $|x_n – s|$ ® $0$ untuk $n$ ® ¥.

# Solusi Persamaan Linier Simultan

4

**Obyektif :**

1. Mengerti penggunaan solusi persamaan linier
2. Mengerti metode eliminasi gauss.
3. Mampu menggunakan metode eliminasi gauss untuk mencari solusi

---

1. Sistem Persamaan Linier
   a. Pendahuluan

Secara umum, sistem persamaan linier dinyatakan sebagai berikut

$$P_n : a_{n1}\,{}_1 + a_{n2}\,{}_2 + ... + a_{nn}\,{}_n = b_n \quad (1)$$

dimana a dan b merupakan konstanta, x adalah variable, n =1 2 3 ....

**Contoh pertama**

Misalnya ada sistem persamaan linier yang terdiri dari empat buah persamaan yaitu $P_1$, $P_2$, $P_3$, dan $P_4$ seperti berikut ini:

$$P_1 : x_1 + x_2 + 3_4 = 4$$

$$P_2 : 2_1 + x_2 - x_3 + x_4 = 1$$

$$P_3 : 3_1 - x_2 - x_3 + 2_4 = -3$$

$$P_4 : -x_1 + 2_2 + 3_3 - x_4 = 4$$

Jadi, Sistem Persamaan Linier adalah sebuah persamaan dimana persamaan ini merupakan persamaan yang tetap atau merupakan produk dari persamaan yang variabel berada di dalamnya. Contohnya, sebuah persamaan yang terdiri dari angka puluhan untuk disetarakan dengan angka nol. Persamaan ini dikatakan linier sebab mereka digambarkan dalam garis lurus di koordinat Kartesius.

Bentuk umum untuk persamaan linier adalah

$$y = mx + b.$$

Dalam bentuk ini, konstanta m akan menggambarkan gradien garis, lalu konstanta b akan memberikan titik tempat sumbu-y bersilangan. Persamaan seperti $x^3$, $y^{1/2}$, dan $xy$ bukanlah persamaan linier.

*Contoh sistem persamaan linier dua variabel:*

$$x + 2y = 10,$$

$$3b + 5c = 4d + 20,$$

$$5x - 3y + 6 = -9x + 8y + 4,$$

b. Sistem Persamaan Linier Dua Variabel

Persamaan linier yang rumit, seperti di sebut di atas, bisa ditulis dengan menggunakan hukum aljabar agar menjadi bentuk yang lebih sederhana. Seperti, contoh, huruf besar di persamaan merupakan konstanta, dan x dan y adalah variabelya.

Bentuk Umumnya :

$$Ax + By + C = 0,$$

dimana konstanta A dan B bila dijumlahkan, hasilnya bukan angka nol. Konstanta dituliskan sebagai $A \geq 0$, seperti yang telah disepakati ahli matematika bahwa konstanta tidak boleh sama dengan nol. Grafik persamaan ini bila digambarkan, akan menghasilkan sebuah garis lurus dan setiap garis dituliskan dalam sebuah persamaan seperti yang tertera diatas. Bila $A \geq 0$, dan x sebagai titik potong, maka titik koordinat-

*x*adalah ketika garis bersilangan dengan sumbu-x (*y* = 0) yang digambarkan dengan rumus -*c/a*. Bila *B*≥ 0, dan y sebagai titik potong, maka titik koordinat- *y* adalah ketika garis bersilangan dengan sumbu-y (*x* = 0), yang digambarkan dengan rumus -*c/b*.

Bentuk Standar :

$$Ax + By = C,$$

dimana, *A* dan *B* jika dijumlahkan, tidak menghasilkan angka nol dan A bukanlah angka negatif. Bentuk standar ini dapat dirubah ke bentuk umum, tapi tidak bisa diubah ke semua bentuk, apabila *A* dan *B* adalah nol.

Bentuk Titik Potong Gradien

Sumbu –y :

$$y = mx + b,$$

dimana m merupaka gradien dari garis persamaan, dan titik koordinat *y* adalah persilangan dari sumbu-*y*. Ini dapat digambarkan dengan *x = 0*, yang memberikan nilai *y = b*. Persamaan ini digunakan untuk mencari sumbu-*y*, dimana telah diketahui nilai dari x. *Y* dalam rumus tersebut merupakan koordinat *y* yang anda taruh di grafik. Sedangkan *X* merupakan koordinat *x* yang anda taruh di grafik.

Sumbu –x :

$$x = \frac{y}{m} + c,$$

dimana m merupakan gradien dari garis persamaan, dan *c* adalah titik potong-*x*, dan titik koordinat *x* adalah persilangan dari sumbu-*x*. Ini dapat digambarkan dengan *y = 0*, yang memberikan nilai *x = c*. Bentuk *y/m* dalam persamaan sendiri berarti bahwa membalikkan gradien dan mengalikannya dengan *y*. Persamaan ini untuk mencari titik koordinat *x*, dimana nilai *y* sudah diberikan.

c. Sistem Persamaan Linier lebih dari dua variable

   Sebuah Persamaan linier lebih daru dua variable seperti berikut ini :

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b.$$

dimana dalam bentuk ini, digambarkan bahwa $a_1$ adalah koefisien, $x$ dan $n$ merupakan variabel dan $b$ adalah konstanta.

d. Susunan Persamaan Linier


1. Susunan Persamaa Linier Homogen

   Kalau semua konstanta $b_i = 0$, persamaan menjadi AX = 0.

   Susunan :

$$\left.\begin{array}{ccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & 0 \\
a_{21}x_1 & + & a_{22}x_{22} & + & \ldots & + & a_{2n}x_n & = & 0 \\
\ldots & & \ldots & & \ldots & & \ldots & & . \\
\ldots & & \ldots & & \ldots & & \ldots & & . \\
\ldots & & \ldots & & \ldots & & \ldots & & . \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & = & 0 \\
 & & & & & & & & \\
A_1x_1 & + & A_2x_2 & + & \ldots & + & A_nx_n & = & 0
\end{array}\right\} \text{atau}$$

Disebut susunan persamaan linier homogen.

2. Susunan Persamaan Linier NonHomogen

   Pandang susunan persamaan linier AX = B, dimana B ≠ 0.

$$\left.\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & .... & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & .... & + & a_{2n}x_n & = & b_2 \\ .... & & .... & & .... & & .... & & .... \\ a_{i1}x_1 & + & a_{i2}x_2 & + & .... & + & a_{in}x_n & = & b_i \\ .... & & .... & & .... & & .... & & .... \\ a_{m1}x_1 & + & a_{m2}x_2 & + & .... & + & A_m x_n & = & b_m \end{array}\right\} \text{, atau :}$$

   $A_1x_1 + A_2x_2 + \ldots + A_nx_n = B$ dimana $A_1$, $A_2$, …, $A_n$ adalah vector-vektor kolom dari matriks koefisien $A_1$. Susunan persamaan linier di atas disebut nonhomogen.

2. Metode Eliminasi Gauss

   a. Pendahuluan

   Metode yang dikembangkan dari metode eliminasi, yaitu menghilangkan atau mengurangi jumlah variable sehingga dapat diperoleh nilai dari suatu variable bebas. Metode eliminasi gauss : metode dimana bentuk matrik augmented, pada bagian kiri diubah menjadi matrik segitiga atas /segitiga bawah dengan menggunakan **OBE** ( **Operasi Baris Elementer** ).

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{bmatrix} \rightarrow \begin{bmatrix} c_{11} & c_{12} & c_{13} & \dots & c_{1n} & d_1 \\ 0 & c_{22} & c_{23} & \dots & c_{2n} & d_2 \\ 0 & 0 & c_{33} & \dots & c_{3n} & d_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & c_{nn} & d_n \end{bmatrix}$$

Sehingga penyelesaian dapat diperoleh dengan :

$$x_n = \frac{d_n}{c_{nn}}$$

$$x_{n-1} = \frac{1}{c_{n-1,n-1}}\left(d_{n-1} - c_{n-1,n}x_n\right)$$

....................................

$$x_2 = \frac{1}{c_{22}}\left(d_2 - c_{23}x_3 - c_{24}x_4 - \dots - c_{2n}x_n\right)$$

$$x_1 = \frac{1}{c_{11}}\left(d_1 - c_{12}x_2 - c_{13}x3 - \dots - c_{1n}x_n\right)$$

Operasi Baris Elementer (OBE) : operasi pengubahan nilai elemen matrik berdasarkan barisnya, tanpa mengubah matriknya. OBE pada baris ke-i + k dengan dasar baris ke i dapat dituliskan dengan :

$a_{i+k,j} = a_{i+k,j} - c.a_{i,j}$ dimana c : konstanta pengali dari perbandingan nilai dari elemen $a_{i,j}$ dan $a_{i+k,i}$.

Jadi prinsipnya Eliminasi Gauss ( EGAUSS ) : merupakan operasi eliminasi dan substitusi variable-variabelnya sedemikian rupa sehingga dapat terbentuk matriks segitiga atas, dan akhirnya solusinya diselesaikan menggunakan teknik substitusi balik ( backsubstitution ).

b. Algoritma Metode Eliminasi Gauss

Algoritma Metode Eliminasi Gauss adalah sbb :

1. Masukkan matrik A, dan vektor B beserta ukurannya n
2. Buat augmented matrik [A|B] namakan dengan A
3. Untuk baris ke *i* dimana *i=1*s/d n, perhatikan apakah nilai $a_{i,i}$ =0 :

   Bila ya : pertukarkan baris ke *i* dan baris ke *i+k ≤n,*dimana $a_{i+k,i} \neq 0$, bila tidak ada berarti perhitungan tidak bisa dilanjutkan dan proses dihentikan dengan tanpa penyelesaian.

   Bila tidak : lanjutkan

4. Untuk baris ke j, dimana *j =i+1 s/d n*

   Lakukan operasi baris elementer :

   Hitung $$c = \frac{a_{j,i}}{a_{i,i}}$$

   Untuk kolom *k* dimana *k=1 s/d n+1*hitung

   Hitung akar, untuk *i =n s/d 1*( bergerak dari baris ke n sampai baris

   pertama) : $$x_i = \frac{1}{a_{i,i}}\left(b_i - a_{i,i+1}x_{i+1} - a_{i,i+2}x_{i+2} - \ldots - a_{i,n}x_n\right)$$

   dimana nilai i+k ≤ n.

c. Teknik Pivoting dalam Metode Eliminasi Gauss

Dalam beberapa kasus, terutama bila dijumpai matriks-matrik yang bersifat ‘singular’ karena adanya ‘kombinasi linier’, solusi secara langsung menggunakan algoritma metode eliminasi Gauss tidak memberikan hasil dan ketelitian yang baik, bahkan seringkali memberikan hasil yang meleset jauh dari yang diharapkan. Untuk menghindari fenomena tersebut, diperlukan modifikasi dari algoritma eliminasi Gauss. Pada prinsipnya, modifikasi tersebut dilakukan dengan memperhatikan hal-hal berikut:

ð Harga **pivot** diambil yang terbesar dari setiap baris dan kolom yang sesuai, yaitu komponen $a_{ii}$ ,

ð Pemilihan **pivot** dilakukan berdasarkan ‘pembandingan harga terbesar (maksimum)’ dari setiap elemen *j* $a_{ji}$ ³ " i,

ð Untuk hasil terbaik, sebaiknya gunakan variabel ‘presisi ganda’

(DOUBLE PRECISION atau REAL*8).

# Solusi Persamaan linier (Lanjutan)

5

**Obyektif :**

1. Mengerti penggunaan solusi persamaan linier
2. Mengerti metode gauss jordan dan gauss seidel
3. Mampu menggunakan metode gauss jordan dan gauss seidel untuk mencari solusi

---

## Metode Iterasi Gauss-Jordan

Metode ini merupakan pengembangan metode eliminasi Gauss, hanya saja augmented matrik, pada sebelah kiri diubah menjadi matrik diagonal sebagai berikut:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & d_1 \\ 0 & 1 & 0 & \dots & 0 & d_2 \\ 0 & 0 & 1 & \dots & 0 & d_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & d_n \end{pmatrix}$$

Penyelesaian dari persamaan linier simultan diatas adalah nilai d1,d2,d3,…,dn dan atau:

*n n x* = *d* , *x* = *d* , *x* = *d* ,...., *x* = *d* 1 1 2 2 3 3

Teknik yang digunakan dalam metode eliminasi Gauss-Jordan ini sama seperti metode eliminasi Gauss yaitu menggunakan OBE (Operasi Baris Elementer). Hanya perhitungan

penyelesaian secara langsung diperoleh dari nilai pada kolom terakhir dari setiap baris.

## Algoritma Metode Eliminasi Gauss-Jordan adalah sebagai berikut:

(1) Masukkan matrik A, dan vektor B beserta ukurannya n

(2) Buat augmented matrik [A|B] namakan dengan A

(4) Untuk baris ke i dimana i=1 s/d n

(a) Perhatikan apakah nilai $a_{i,i}$ sama dengan nol :
Bila ya :
pertukarkan baris ke i dan baris ke i+k≤n, dimana $a_{i+k,i}$ tidak sama dengan nol, bila tidak ada berarti perhitungan tidak bisa dilanjutkan dan proses dihentikan dengan tanpa penyelesaian.
Bila tidak : lanjutkan

(b) Jadikan nilai diagonalnya menjadi satu, dengan cara untuk setiap kolom k dimana k=1 s/d n+1, hitung $a_{i,j} = a_{i,k}/a_{i,i}$

(6) Untuk baris ke j, dimana j = i+1 s/d n
Lakukan operasi baris elementer: untuk kolom k dimana k=1 s/d n

Hitung $c = a_{j,i}$

Hitung $a_{j,k} = a_{j,k} – c.a_{i,k}$

(7) Penyelesaian, untuk i = n s/d 1 (bergerak dari baris ke n sampai baris pertama)

$X_i = a_{i,n+1}$

**Metode Iterasi Gauss-Seidel**

Metode interasi Gauss-Seidel adalah metode yang menggunakan proses iterasi hingga diperoleh nilai-nilai yang berubah. Bila diketahui persamaan linier simultan:

$a_{11}\ x_1 + a_{12}\ x_2 + a_{13}\ x_3 + \ldots + a_{1n}\ xn = b_1$

$a_{21}\ x_1 + a_{22}\ x_2 + a_{23}\ x_3 + \ldots + a_{2n}\ xn = b_2$

$a_{31}\ x_1 + a_{33}\ x_2 + a_{33}\ x_3 + \ldots + a_{3n}\ xn = b_3$

… … … … … … … … … … … … … … …

$a_{n1}\ x_1 + a_{n2}\ x_2 + an_3\ x_3 + \ldots + ann\ xn = bn$

Berikan nilai awal dari setiap xi (i=1 s/d n) kemudian persamaan linier simultan diatas dituliskan menjadi:

$$X_1 = 1/a_{11}\ (b_1 - a_{12}x_2 - a_{13}x_3 - \ldots - a_{1n}x_n)$$

$$X_2 = 1/\ a_22(b_n - a_{21}x_1 - a_{23}x_3 - \ldots - a_{2n}x_n)$$

………………………………………………………………………..............

$$Xn = 1/a_{nm}\ (b_n - a_{n1}\ X_1 - a_{n2}X_2 - \ \ldots - a_{nn-1}X_{n-1})$$

Dengan menghitung nilai-nilai *xi* (i=1 s/d n) menggunakan persamaan-persamaan di atas
secara terus-menerus hingga nilai untuk setiap *xi* (i=1 s/d n) sudah sama dengan nilai xi pada iterasi sebelumnya maka diperoleh penyelesaian dari persamaan linier simultan tersebut. Atau dengan kata lain proses iterasi dihentikan bila selisih nilai *xi* (i=1 s/d n) dengan nilai *xi* pada iterasi sebelumnya kurang dari nilai tolerasi error yang ditentukan.

***Catatan:***

Hati-hati dalam menyusun sistem persamaan linier ketika menggunakan metode iterasi Gauss-Seidel ini. Perhatikan setiap koefisien dari masing-masing xi pada semua persamaan di diagonal utama (aii). Letakkan nilai-nilai terbesar dari koefisien untuk setiap xi pada diagonal utama. Masalah ini adalah '**masalah pivoting**' yang harus benar

benar diperhatikan, karena penyusun yang salah akan menyebabkan iterasi menjadi divergen dan tidak diperoleh hasil yang benar.

**<u>Algoritma Metode Iterasi Gauss-Seidel adalah sebagai berikut:</u>**

(1) Masukkan matrik **A**, dan vektor **B** beserta ukurannya *n*

(2) Tentukan batas maksimum iterasi *max_iter*

(3) Tentukan toleransi error ε

(4) Tentukan nilai awal dari *xi*, untuk i=1 s/d n

(5) Simpan *xi* dalam *si*, untuk i=1 s/d n

(6) Untuk i=1 s/d n hitung :

$$X_i = \frac{1}{a_{i,}} \left( b_i - \sum_{j \neq 1} a_{i,j} X_j \right)$$

(7) iterasi ← iterasi+1

(8) Bila iterasi lebih dari max_iter atau tidak terdapat *ei*<ε untuk i=1 s/d n maka proses dihentikan dari penyelesaiannya adalah *xi* untuk i=1 s/d n. Bila tidak maka ulangi langkah (5).

# INTERPOLASI

6

## Obyektif :

1. Mengerti maksud Interpolasi
2. Mengerti dan memahami bentuk interpolasi polinomial

---

## Pengertian Interpolasi

Interpolasi adalah teknik mencari harga suatu fungsi pada suatu titik diantara 2 titik yang nilai fungsi pada ke-2 titik tersebut sudah diketahui

dpl. : cara menentukan harga fungsi f dititik x* ε [x0,xn] dengan menggunakan informasi dari seluruh atau sebagian titik-titik yang diketahui ( x0, x1, …., xn)

| X | x0 | x1 | x2 | ……. | xn |
|---|---|---|---|---|---|
| f(x) | f(x0) | f(x1) | f(x2) | ……. | f(xn) |

## Teknik Umum yang digunakan

(i) Membentuk polinomial berderajat ≤ n yg mempunyai harga fungsi di titik-titik yang diketahui → Polinomial Interpolasi

(ii) Masukkan titik yang ingin dicari harga fungsinya ke dalam polinomial interpolasi

## Jenis Interpolasi

- Interpolasi Polinomial
- Interpolasi Lagrange

- Interpolasi Newton

## Interpolasi Polinomial

Interpolasi polynomial digunakan untuk mencari titik-titik antara dari n buah titik $P_1(x_1,y_1)$, $P_2(x_2,y_2)$, $P_3(x_3,y_3)$, …, $P_N(x_N,y_N)$ dengan menggunakan pendekatan fungsi polynomial pangkat n-1:

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$$

Masukkan nilai dari setiap titik ke dalam persamaan polynomial di atas dan diperoleh persamaan simultan dengan n persamaan dan n variable bebas:

$$\begin{array}{l}
y_1 = a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 + \ldots + a_{n-1} x_1^{n-1} \\
y_2 = a_0 + a_1 x_2 + a_2 x_2^2 + a_3 x_2^3 + \ldots + a_{n-1} x_2^{n-1} \\
y_3 = a_0 + a_1 x_3 + a_2 x_3^2 + a_3 x_3^3 + \ldots + a_{n-1} x_3^{n-1} \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
y_n = a_0 + a_1 x_n + a_2 x_n^2 + a_3 x_n^3 + \ldots + a_{n-1} x_n^{n-1}
\end{array}$$

Penyelesaian persamaan simultan di atas adalah nilai-nilai $a_0, a_1, a_2, a_3, \ldots, a_n$ yang merupakan nilai-nilai koefisien dari fungsi pendekatan polynomial yang akan digunakan.

Dengan memasukkan nilai x dari titik yang dicari pada fungsi polinomialnya, akan diperoleh nilai y dari titik tersebut.

## Algoritma Interpolasi Polinomial :

(1) Menentukan jumlah titik N yang diketahui.

(2) Memasukkan titik-titik yang diketahui ( , ) *i i i P = x y* untuk i=1,2,3,…,N

(3) Menyusun augmented matrik dari titik-titik yang diketahui sebagai berikut:

$$J = \left[\begin{array}{ccccc|c} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} & y_1 \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} & y_2 \\ 1 & x_3 & x_3^2 & \dots & x_3^{n-1} & y_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} & y_n \end{array}\right]$$

(4) Menyelesaikan persamaan simultan dengan augmented matrik di atas dengan menggunakan metode eliminasi gauss/Jordan.

(5) Menyusun koefisien fungsi polynomial berdasarkan penyelesaian persamaan simultan di atas.

$$a = \{ a \; a = J(i, n), 0 \le i \le n - 1 \}$$

(6) Memasukkan nilai x dari titik yang diketahui

(7) Menghitung nilai y dari fungsi polynomial yang dihasilkan

$$y = \sum_{i=0}^{N-1} a_i x^i$$

(8) Menampilkan titik (x,y)

# INTERPOLASI (Lanjutan)

7

## Obyektif :

1. Mengerti maksud Interpolasi

2. Mengerti dan memahami bentuk interpolasi Lagrange

3 Mampu menginterpolasikan sebuah fungsi menggunakan polinomial lagrange.

---

## Interpolasi Lagrange

Interpolasi Lagrange adalah salah satu formula untuk interpolasi berselang tidak sama selain formula interpolasi Newton umum & metoda Aitken. Walaupun demikian dapat digunakan pula untuk interpolasi berselang sama.

Misalkan fgs. y(x) kontinu & diferensiabel sampai turunan (n+1) dalam interval buka (a,b). Diberikan (n+1) titik (x0,y0), (x1,y1), …, (xn,yn) dengan nilai x tidak perlu berjarak sama dengan yang lainnya, dan akan dicari suatu polinom berderajat n. Untuk pemakaian praktis, formula interpolasi Lagrange dapat dinyatakan sbb. :

Jika y(x) : nilai yang diinterpolasi; x : nilai yg berkorespondensi dg y(x)

x0, x1, …., xn : nilai x dan y0, y1, …., yn : nilai y

$$y(x) = \frac{(x-x_1)(x-x_2)\ldots(x-x_n)}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_n)} y_0 + \frac{(x-x_0)(x-x_2)\ldots(x-x_n)}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_n)} y_1 + \cdot \cdot \frac{(x-x_0)(x-x_1)\ldots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\ldots(x_n-x_{n-1})} y_n$$

## Algoritma Interpolasi Lagrange :

(1) Tentukan jumlah titik (N) yang diketahui

(2) Tentukan titik-titik $P_i(x_i,y_i)$ yang diketahui dengan i=1,2,3,…,N

(3) Tentukan x dari titik yang dicari

(4) Hitung nilai y dari titik yang dicari dengan formulasi interpolasi lagrange

$$y = \sum_{i=1}^{N} y_i \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$$

(5) Tampilkan nilai (x,y)

Nilai yg. berkorespondensi dengan y = 10log x adalah :

| X | 300 | 304 | 305 | 307 |
|---|---|---|---|---|
| 10log x | 2,4771 | 2,4829 | 2,4843 | 2,4871 |

Carilah 10log 301 ?

Untuk menghitung y(x) = 10log 301 dimana x = 301, maka nilai diatas menjadi

| x0 = 300 | x1 = 304 | x2 = 305 | x3 = 307 |
|---|---|---|---|
| y0 = 2,4771 | y1 = 2,4829 | y2 = 2,4843 | y3 = 2,4871 |

$$y(x) = \frac{(301-304)(301-305)(301-307)}{(300-304)(300-305)(300-307)}2,4771 + \frac{(301-300)(301-305)(301-307)}{(304-300)(304-305)(304-307)}2,4829 +$$

$$\frac{(301-300)(301-304)(301-307)}{(305-300)(305-304)(305-307)}2,4843 + \frac{(301-300)(301-304)(301-305)}{(307-301)(307-304)(307-305)}2,4871$$

$$= 1,2739 + 4,9658 - 4,4717 + 0,7106$$

Contoh :

Bila y1 = 4, y3 = 12, y4 = 19 dan yx = 7, carilah x ?

Jawab :

Karena yg ditanyakan nilai x dengan nilai y diketahui, maka digunakan interpolasi invers atau kebalikan yg analog dg interpolasi Lagrange.

$$x = \frac{(7-12)(7-19)}{(4-12)(4-19)}(1) + \frac{(7-4)(7-19)}{(12-4)(12-19)}(3) +$$

$$x = \frac{1}{2} + \frac{27}{14} - \frac{4}{7} = 1,86$$

Nilai sebenarnya dari x adalah 2, karena nilai-nilai atau data diatas adalah hasil dari polinom y(x) = x2 + 3.

Adapun untuk membentuk polinom derajat 2 dengan diketahui 3 titik, dapat menggunakan cara yang sebelumnya pernah dibahas dalam hal mencari persamaan umum polinomial kuadrat