Persamaan Non Linier
Mencari Akar-akar Persamaan

Muh. Arif Rahman
Program Studi Ilmu Komputer/T.Informatika
Jurusan T.Informatika
Fakultas Ilmu Komputer
Universitas Brawijaya
2017

Materi

- Metode Bracketing
  - Grafis
  - Bisection
  - Regula Fasi
- Metode Terbuka
  - Fixed Iteration
  - Newton Method
  - Secant Method

Persamaan Linier dan Non Linier

- Persamaan Linier
  - Persamaan dimana pangkat variable nya maksimal hanya 1
  - Contoh:
    - $y = 2x + 1$

- Persamaan Non Linier
  - Persamaan dimana pangkat variable nya maksimal n serta bisa juga mengandung fungsi trigonometri (sin, cos dll) serta fungsi lainnya
  - Contoh:
    - $y = 2x^2 + 1$
    - $y = 2x + \sin^2 x$

Akar akar persamaan Non Linier

- Misal $f(x)$ adalah persamaan non linier
- Jika $f(x) = 0$ maka akan dicari nilai x sedemikian hingga $f(x) = 0$

Akar akar Persamaan Kuadrat

- Persamaan kuadrat mempunyai bentuk umum sbb:

$$f(x) = ax^2 + bx + c = 0$$

- Untuk mencari akar akar persamaan $f(x)$ bisa menggunakan:
  - Rumus: $x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$ dan $x_2 = \frac{-b+\sqrt{b^2-4ac}}{2a}$ dimana $a \neq 0$
  - Memfaktorkan persamaan

Akar akar persamaan Non Linier

Metode Numerik

- Persamaan Non Linier

$$f(x) = 0$$

- Untuk mencari x sedemikian hingga $f(x) = 0$ tidak selalu mudah jika menggunakan cara memfaktorkan
- Metode Numerik dapat digunakan untuk mencari penyelesaian aka akar persamaan non linier
- Ada 2 cara yaitu menggunakan
  - Metode Bracketing
  - Metode Open

Metode Bracketing (mengurung)

Metode Numerik

- Metode untuk mencari akar persamaan non linier denga cara "menebak" 2 nilai (mengurung) dan memprosesnya hingga diperoleh nilai x sedemikian hingga $f(x) = 0$
- Yang termasuk dalam kategori metode bracketing adalah
  - Metode Grafis
  - Metode Biseksi (membagi dua)
  - Metode False Position (Kesalahan posisi)

Metode Grafis

Metode Numerik

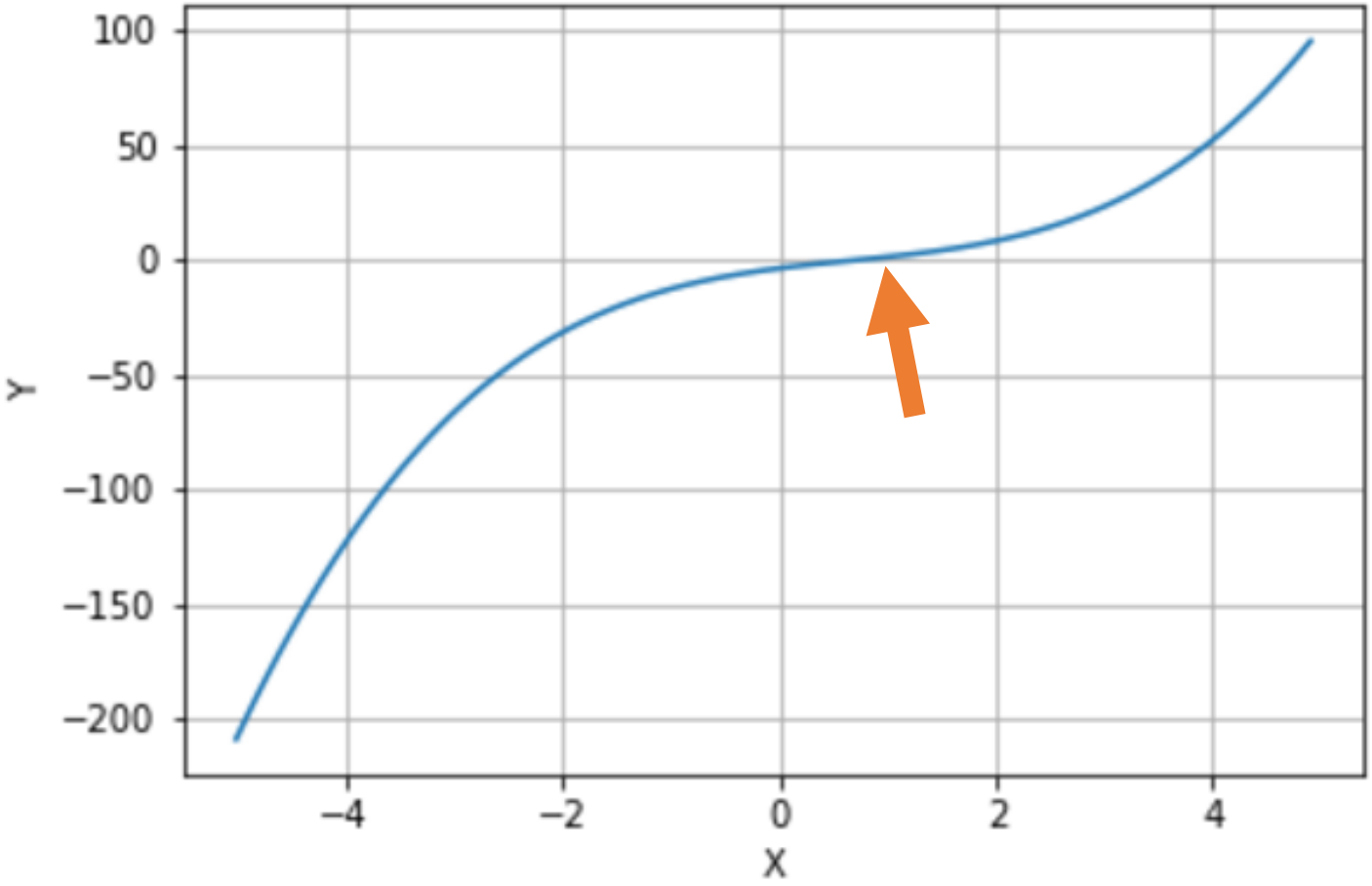
- Metode ini cukup mudah untuk digunakan sepanjang kita masih bisa menggambar fungsinya.
- Dari gambar kemudian ditebak, kira kira nilai x berapakah sedemikian hingga $f(x) = 0$

Contoh: $y = x^3 - 2x^2 + 6x - 4$

- Di tentukan 2 nilai yaitu a dan b dimana $a < b$ bahwa
- Misal $a = -5$ dan $b = 5$

```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(-5.0, 5.0, 0.1)
y = x**3 - 2*(x**2) + 6*x - 4
plt.plot(x,y,'-')
plt.xlabel('X'); plt.ylabel('Y'); plt.grid(True)
plt.show()
```

Gambar

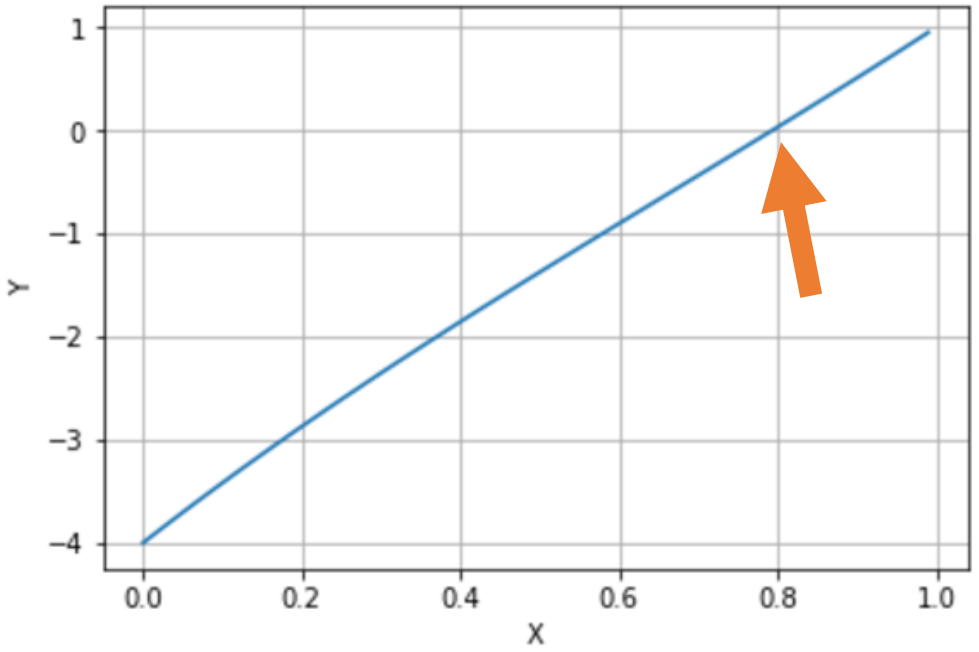


- Dari gambar dapat diketahui nilai x yang menyebabkan $f(x) = 0$ berada disekitar 0 s/d 1

Mencari x sehingga $f(x) = 0$

- Jalankan program dengan batasan $a = 0$ dan $b = 1$

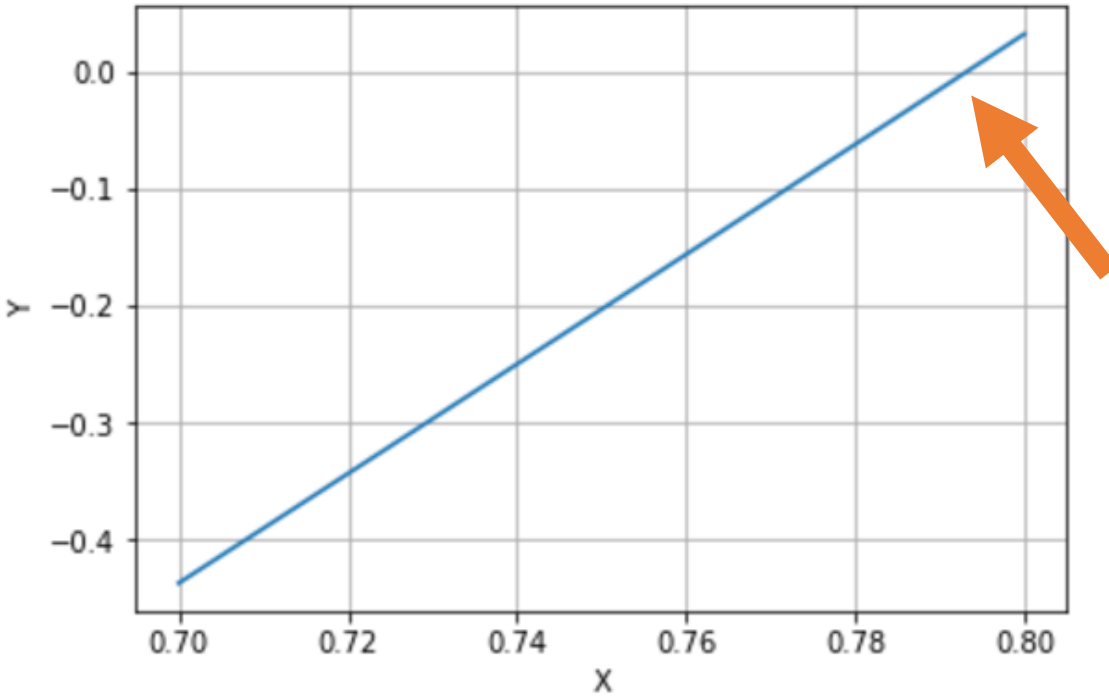
```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0.0, 1.0, 0.01)
y = x**3 - 2*(x**2) + 6*x - 4
plt.plot(x,y,'-')
plt.xlabel('X'); plt.ylabel('Y')
plt.grid(True)
plt.show()
```



Mencari x sehingga $f(x) = 0$

- Jalankan program dengan batasan $a = 0.7$ dan $b = 0.8$

```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0.7, 0.8, 0.01)
y = x**3 - 2*(x**2) + 6*x - 4
plt.plot(x,y,'-')
plt.xlabel('X'); plt.ylabel('Y')
plt.grid(True)
plt.show()
```



Kelemahan metode Grafis

- Hasilnya tidak akurat, karena pengamatan hanya melalui grafis
- Tidak praktis.
  - Menggambar grafis tidak mudah, kecuali menggunakan tool software khusus

Keunggulan metode Grafis

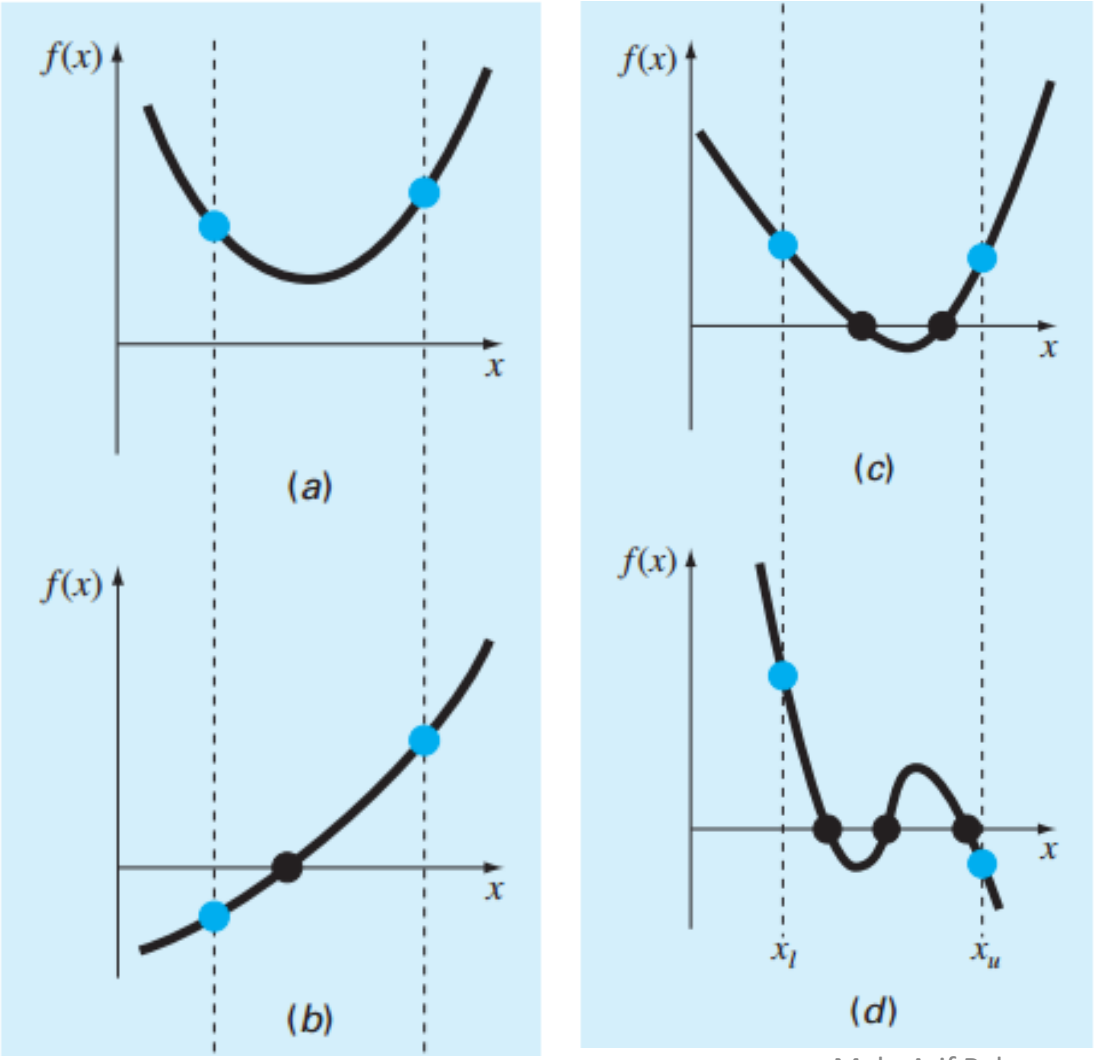
- Membantu memahami perilaku fungsi
- Membantu menentukan nilai a dan b sebagai batas atas dan bawah dalam memprediksi dimana letak x sedemikian hingga $f(x) = 0$

Metode Bisection (Biseksi)

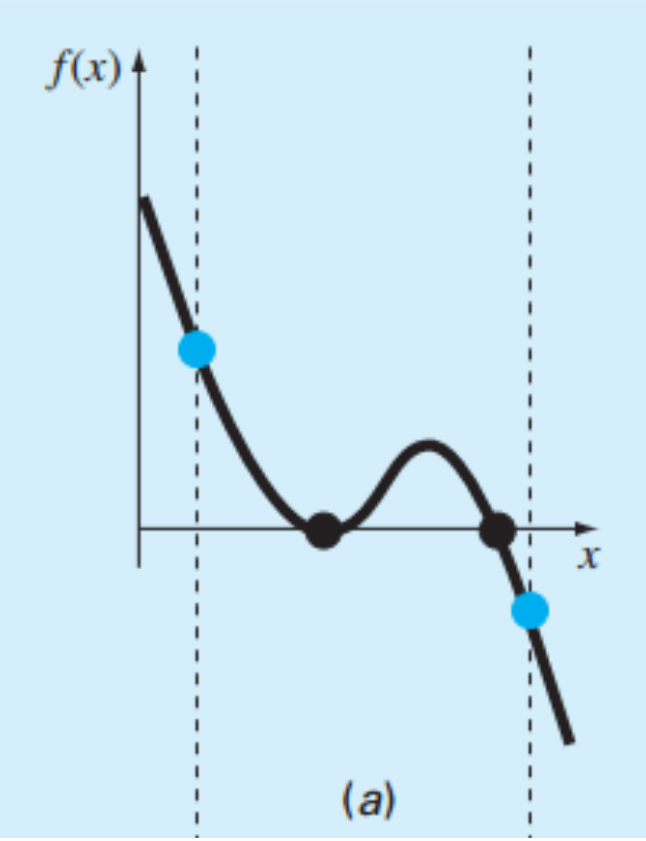
Metode Numerik

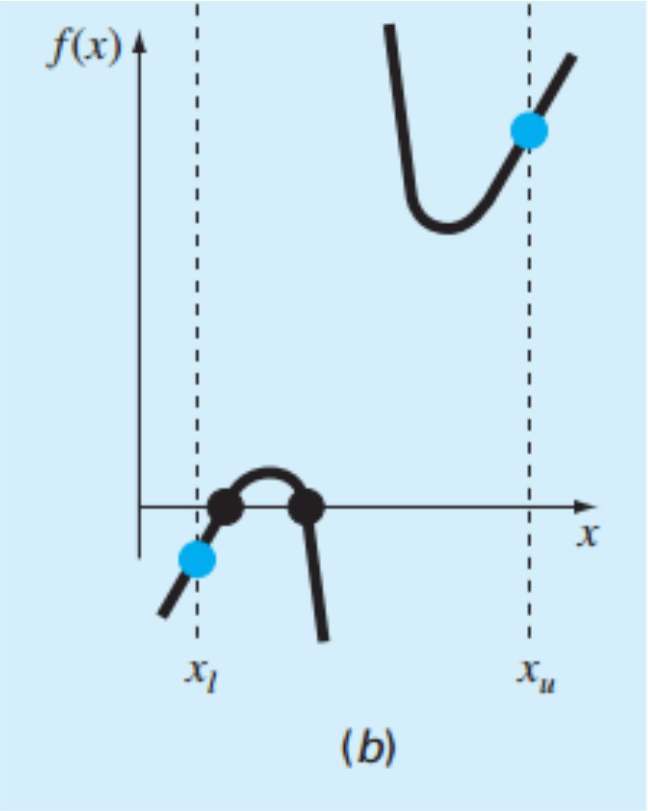
- Membagi menjadi 2 bagian
  - Ada aproksimasi nilai atas (b) dan aproksimasi nilai bawah (a) dengan harapan nilai akar yang sesungguhnya berada di antaranya

Menduga nilai a dan b (1)

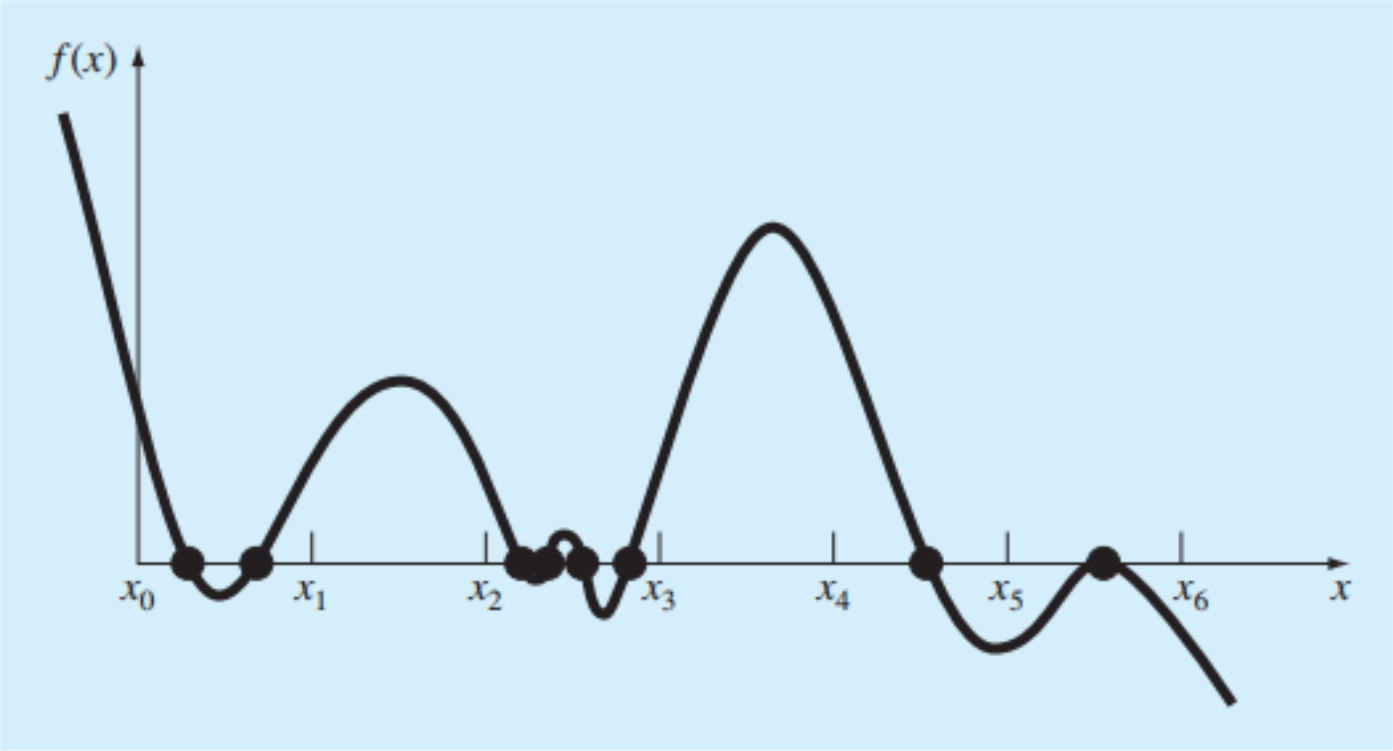


Menduga nilai a dan b (2)



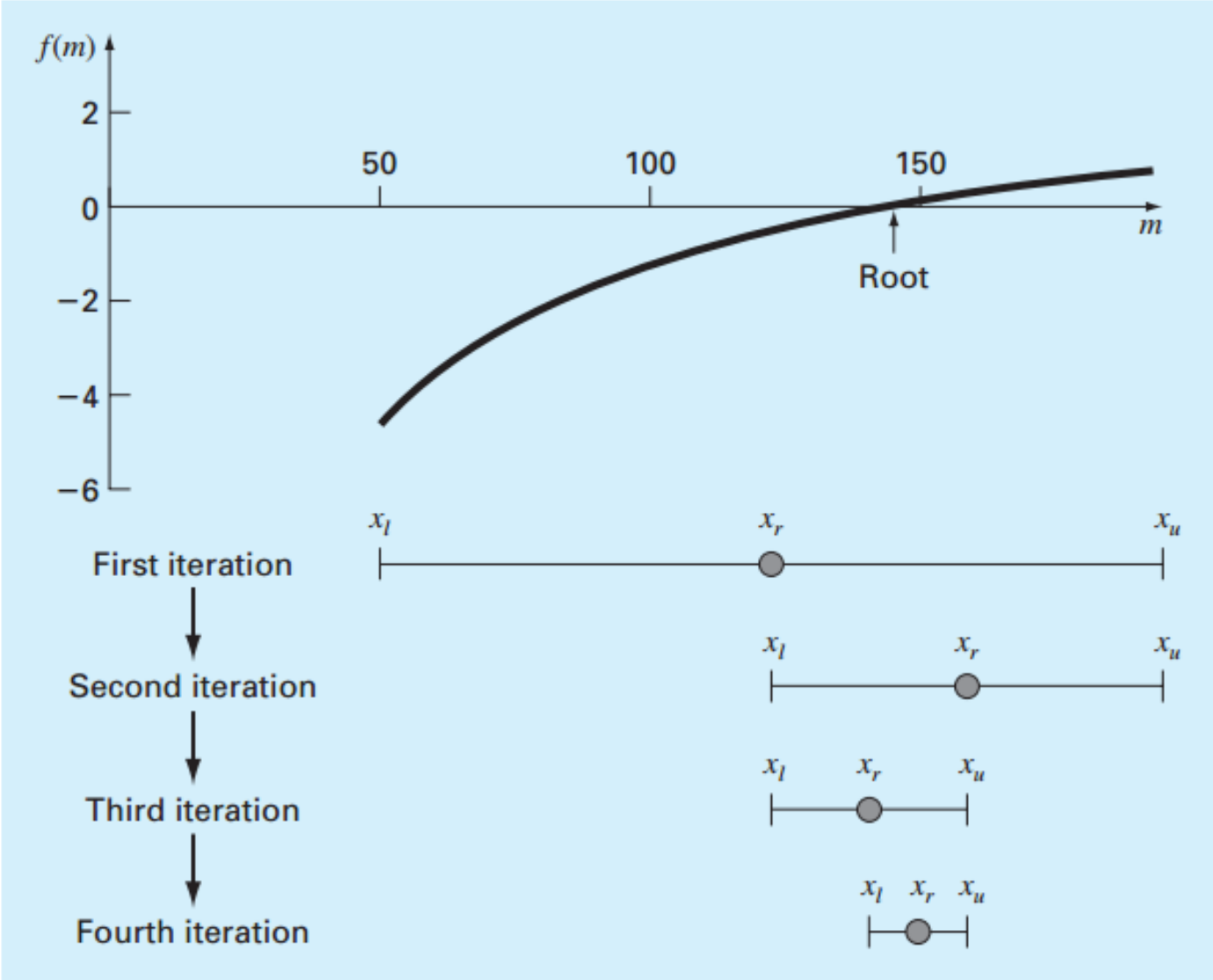


Menduga nilai a dan b (3)



Berapa nilai a dan b nya?

Ilustrasi menggunakan metode biseksi



Algoritma Biseksi

- Tahap 1:
  - Pilih a dan b sedemikian hingga $f(a) * f(b) < 0$
- Tahap 2:
  - Hitung nilai tengah dari a dan b dengan rumus $c = \frac{a+b}{2}$
- Tahap 3:
  - Jika $f(a) * f(c) < 0$ maka $b = c$, kembali ke tahap 2
  - Jika $f(a) * f(c) > 0$ maka $a = c$, kembali ke tahap 2
  - Jika $f(a) * f(c) = 0$, akar ditemukan. Program selesai

Program dan hasilnya


```
fx = lambda x: x**3 - 2*(x**2) + 6*x - 4.0
a=float(input("Masukkan a: "))
b=float(input("Masukkan b: "))
if (fx(a)*fx(b)) < 0:
    while abs(fx(a)*fx(b)) > 0.000001:
        c = (a+b)/2
        if (fx(a)*fx(c) < 0.0):
            b = c
        else:
            a = c
    print("Akar persamaan adalah %f" % (c))
else:
    print("penentuan a dan b salah ")
```

latihan

- Ubah program sehingga output yang dihasilkan adalah:

```
Iterasi          a              b              Nilai Tengah
-------------------------------------------------------------
0                0.0            1.0            …
…                …              …              …
```

Soal:

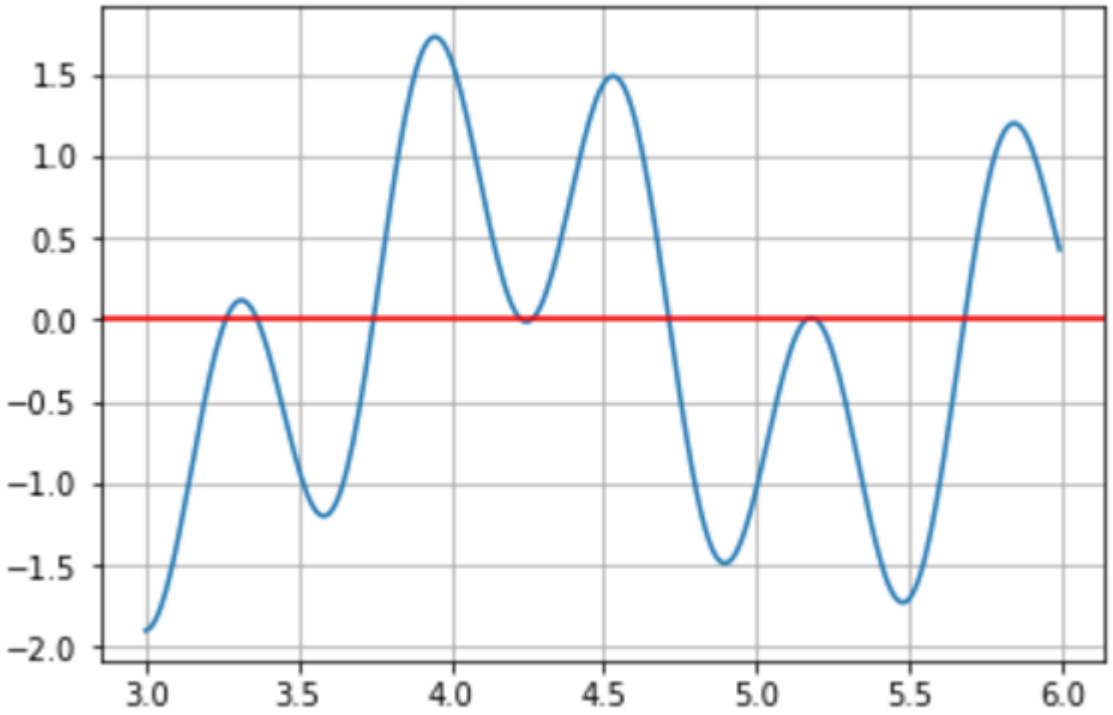
- Diketahui fungsi $f(x) = \sin(10x) + \cos(3x)$
- Nilai x berada pada $[3,6]$
  - Agar gambar halus, buatlah nilai increment adalah 0.01
- Buat program untuk membuat gambar dari fungsi $f(x)$
- Tentukan di titik manakah anda akan mengambil nilai a dan b
- Dengan nilai a dan b tsb, berapakah nilai akar nya?

Jawaban Soal: Program


```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(3.0, 6.0, 0.01)
y = np.sin(10*x) + np.cos(3*x)

plt.plot(x, y, '-')
plt.grid(True); plt.axhline(y=0, color='r')
plt.show()
```



Soal:


```
fx = lambda x: x**3 - 2*(x**2) + 6*x - 4.0
a=float(input("Masukkan a: "))
b=float(input("Masukkan b: "))
if (fx(a)*fx(b)) < 0:
    while abs(fx(a)*fx(b)) > 0.000001:
        c = (a+b)/2
        if (fx(a)*fx(c) < 0.0):
            b = c
        else:
            a = c
    print("Akar persamaan adalah %f" % (c))
else:
    print("penentuan a dan b salah ")
```

- Perhatikan tanda panah disamping kiri tsb.
- Nilai tersebut adalah threshold perulangan untuk komputasi apakah dilanjutkan atau dihentikan

Soal

Metode Numerik

- Ubah program tsb agar evaluasi threshold nya ditentukan dari rumus berikut:

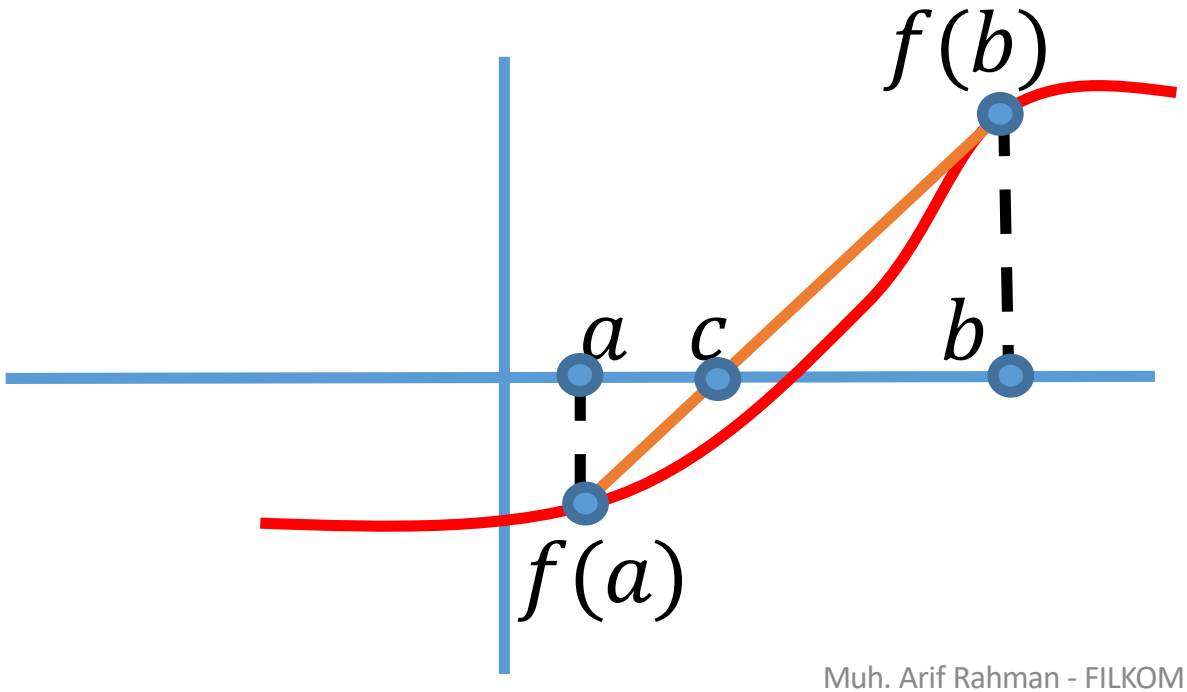
$$|E_r| = |\frac{c_c - c_p}{c_c} \; x \; 100\% \; | < 0.5\%$$

- Dimana:
  - E_r : Galat relative
  - c_c : nilai tengah saat ini (current)
  - c_p : nilai tengah sebelumnya (prev)

Metode False Position (posisi salah)

Metode Numerik

- Metode ini juga sering dinamakan metode linear interpolasi
- Root estimasi ditentukan dengan membuat garis dari titik $f(a)$ dengan $f(b)$



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

Algoritma

- Tahap 1:
  - Pilih a dan b sedemikian hingga $f(a) * f(b) < 0$
- Tahap 2:
  - Hitung c dengan rumus $c = b - \frac{f(b)(a-b)}{f(a)-f(b)}$
- Tahap 3:
  - Jika $f(a) * f(c) < 0$ maka $b = c$, kembali ke tahap 2
  - Jika $f(a) * f(c) > 0$ maka $a = c$, kembali ke tahap 2
  - Jika $f(a) * f(c) = 0$, akar ditemukan. Program selesai

Program

```
fx = lambda x: x**3 - 2*(x**2) + 6*x - 4.0
a=float(input("Masukkan a: "))
b=float(input("Masukkan b: "))
cp = a
Er = 10000.0
if (fx(a)*fx(b)) < 0:
    n = 0
    print("Iterasi\t a\t   b\t     c\t      fx\t Er")
    while abs(Er) > 0.5:
        c = b - (fx(b)*(a-b))/(fx(a)-fx(b))
        cc = c
        Er = ((cc-cp)/cc)*100
        print("%d\t%f  %f  %f  %f  %f" % (n,a,b,c,fx(c),abs(Er)))
        cp = cc; n += 1

        if (fx(a)*fx(c) < 0.0):
            b = c
        else:
            a = c
    print("Akar persamaan adalah %f" % (c))
else:
    print("penentuan a dan b salah ")
```

Output:


```
Masukkan a: -2
Masukkan b: 2
Iterasi  a           b           c           fx          Er
0       -2.000000   2.000000   1.200000   2.048000   266.666667
1       -2.000000   1.200000   1.007519   1.037651   19.104478
2       -2.000000   1.007519   0.913058   0.572193   10.345512
3       -2.000000   0.913058   0.861885   0.325865   5.937388
4       -2.000000   0.861885   0.833035   0.188399   3.463192
5       -2.000000   0.833035   0.816453   0.109773   2.030949
6       -2.000000   0.816453   0.806825   0.064232   1.193387
7       -2.000000   0.806825   0.801202   0.037675   0.701789
8       -2.000000   0.801202   0.797908   0.022128   0.412839
Akar persamaan adalah 0.797908
```

Metode terbuka

Metode Numerik

- Metode terbuka menggunakan satu titik untuk menduga apakah titik tersebut akar dari persamaan $f(x)$
- Kemudian update dilakukan secara iteratif untuk mencari x sedemikian hingga $f(x) = 0$

Metode Iterasi titik tetap

- Permasalahan:

$$f(x) = 0$$

Dapat ditulis sbb:

$$x = g(x)$$

Algoritma iterasi titik tetap mencari x^* sedemikian hingga

$$x^* = g(x^*)$$

Bingung?

Proses

- Metode iterasi titik tetap mencari x^* menggunakan iterasi

$$x_1, x_2, x_3, \ldots x_k, \ldots$$

- Yaitu:

$$x_{k+1} = g(x_k)$$

- Starting from an initial iterate x_0. If such a sequence converges, then the limit must be a fixed point.
- The convergence properties of the fixed point iteration depend on the choice of the function g

Algoritma iterasi titik tetap

Algorithm: Fixed Point Iteration.
Given a scalar continuous function in one variable, $f(x)$, select a function $g(x)$ such that x satisfies $f(x) = 0$ if and only if $g(x) = x$. Then:

1. Start from an *initial guess* x_0.
2. For $k = 0, 1, 2, \ldots,$ set

$$x_{k+1} = g(x_k)$$

until x_{k+1} satisfies termination criteria.

Pertanyaan

- Misal, kita tentukan fungsi $g \in C[a, b]$

1. Is there a fixed point x^* in $[a, b]$?
2. If yes, is it unique?
3. Does the sequence of iterates converge to a root x^*?
4. If yes, how fast?
5. If not, does this mean that no root exists?

Is there a fixed point x^* in [a,b]?

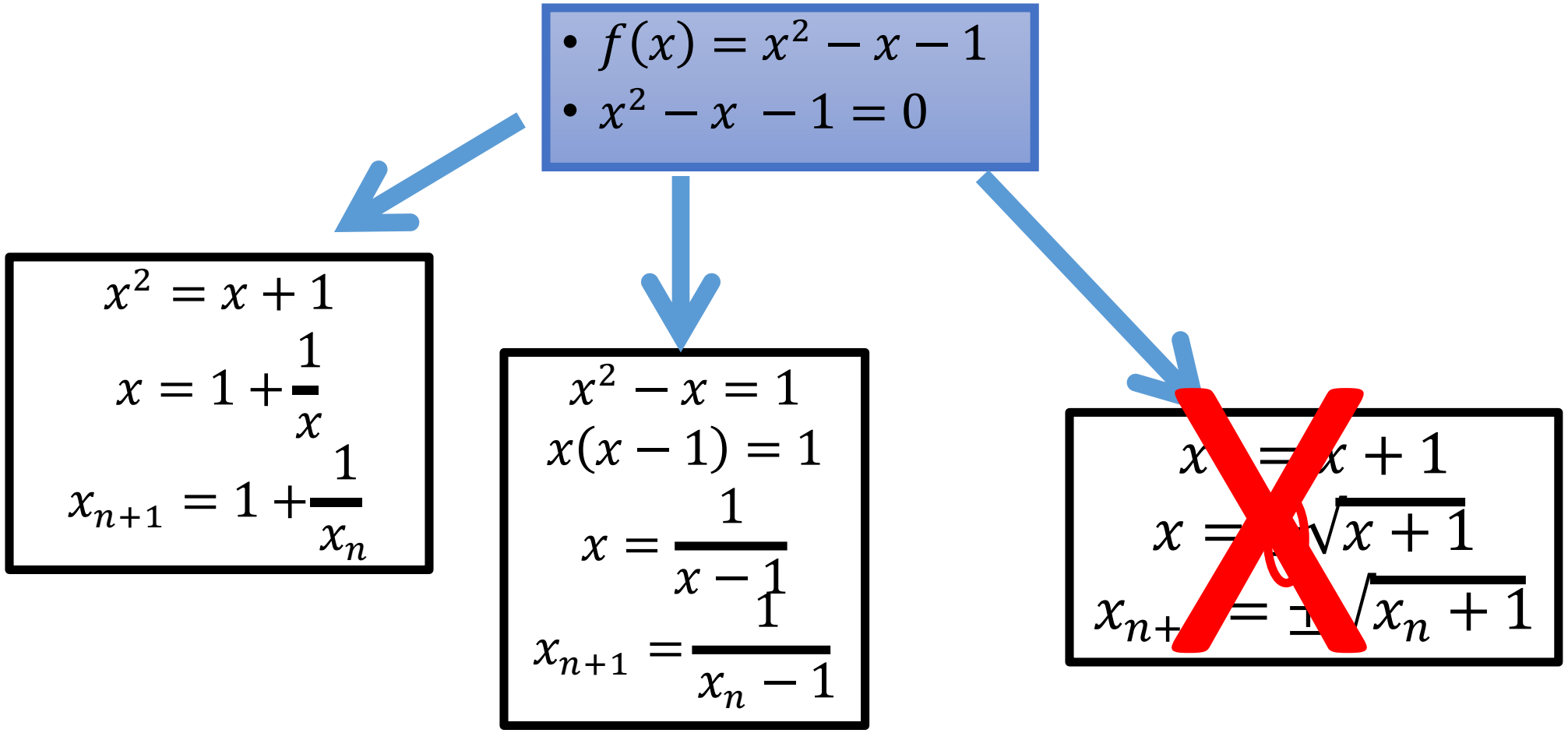
- Pada selang a dan b ([a,b]) yaitu $a < b$ sehingga

$$g(a) \geq a \text{ dan } g(b) \leq b$$

- Jika
  - $g(a) = a$ atau $g(b) = b$

- Maka iterasi titik tetap telah ditemukan.

Contoh: $f(x) = 0$


• $f(x) = x^2 - x - 1$
• $x^2 - x - 1 = 0$
$x^2 = x + 1$
$x = 1 + \frac{1}{x}$
$x_{n+1} = 1 + \frac{1}{x_n}$
$x^2 - x = 1$
$x(x-1) = 1$
$x = \frac{1}{x-1}$
$x_{n+1} = \frac{1}{x_n - 1}$
$x = x + 1$
$x = \sqrt{x+1}$
$x_{n+1} = \sqrt{x_n + 1}$

Contoh: $f(x) = 0$

- $f(x) = x^2 - x - 1$
- $x^2 - x - 1 = 0$

$$x^2 = x + 1$$

$$x = 1 + \frac{1}{x}$$

$$x_{n+1} = 1 + \frac{1}{x_n}$$

$$x - x = 1$$

$$x(x - 1) = 1$$

$$x = \frac{1}{x - 1}$$

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

Manakah yang dipilih?

Komputasi $x_{n+1} = 1 + \frac{1}{x_n}$

Perhitungan menggunakan excel

n	Xn	Xn+1
0	2	1,5
1	1,5	1,666667
2	1,666667	1,6
3	1,6	1,625
4	1,625	1,615385
5	1,615385	1,619048
6	1,619048	1,617647
7	1,617647	1,618182
8	1,618182	1,617978
9	1,617978	1,618056
10	1,618056	1,618026
11	1,618026	1,618037
12	1,618037	1,618033

Berapakah nilai x^*?

Komputasi $x_{n+1} = \frac{1}{x_n - 1}$

Perhitungan menggunakan exel

n	Xn	Xn+1
0	2	1
1	1	#DIV/0!
2	#DIV/0!	#DIV/0!
3	#DIV/0!	#DIV/0!

Berapakah nilai x^*?

Apa kesimpulan saudara?

Mengingatkan

- Starting from an initial iterate x_0.
- If such a sequence converges, then the limit must be a fixed point.
- The convergence properties of the fixed point iteration depend on the choice of the function g

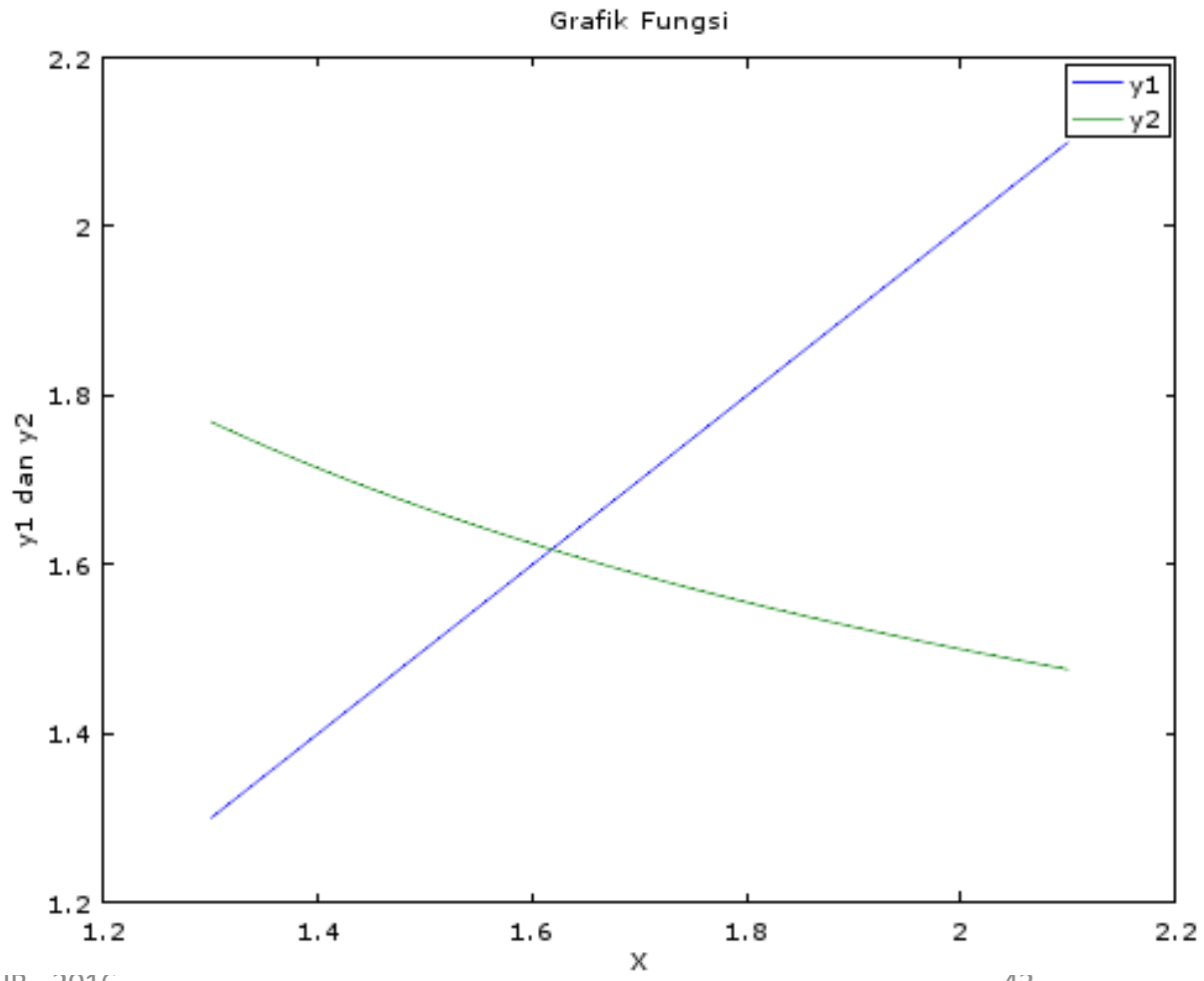
Ilustrasi $x_{n+1} = 1 + \frac{1}{x_n}$

Ada 2 fungsi yaitu:

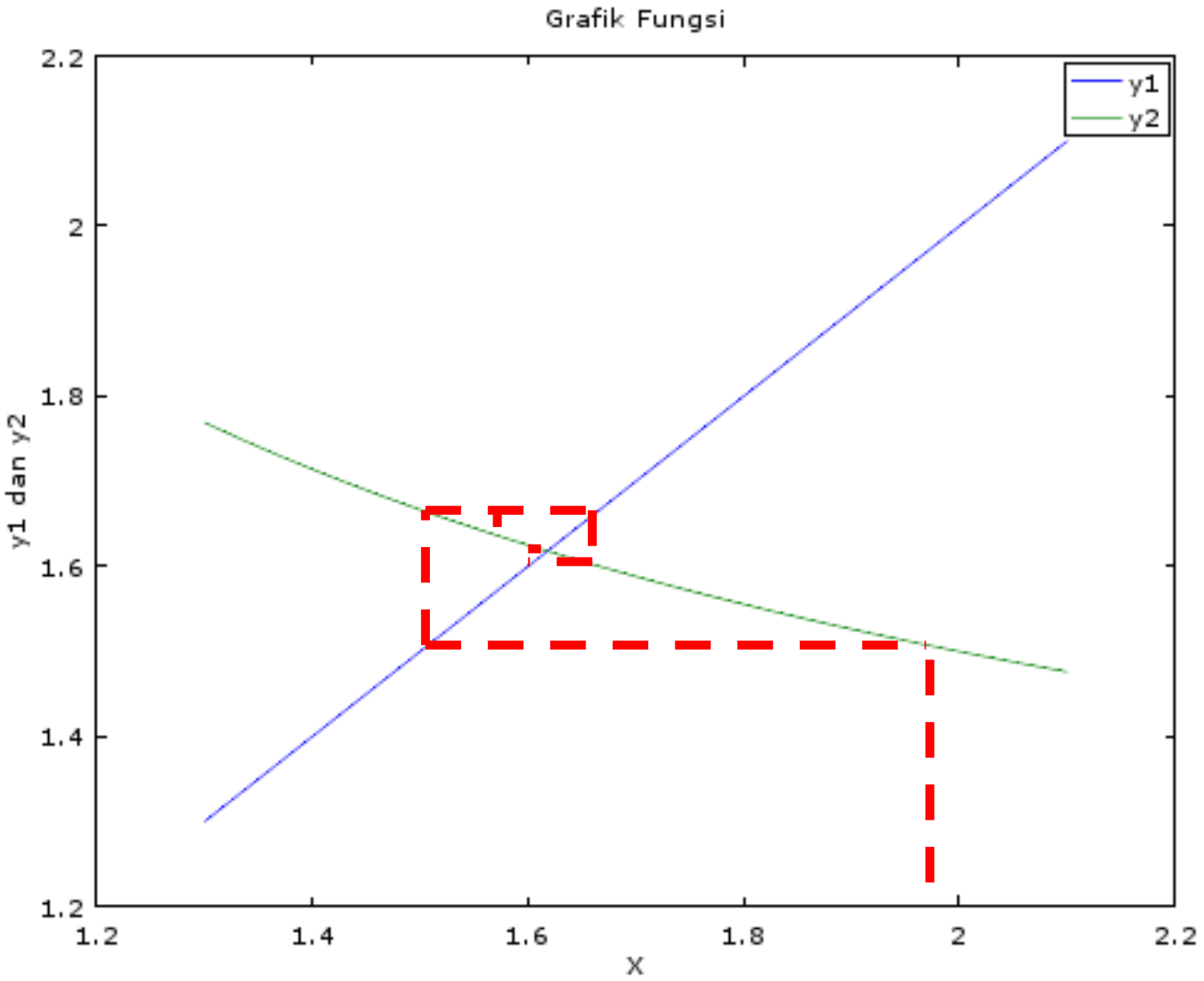
- $y1 = x$ dan
- $y2 = 1 + \frac{1}{x}$

Code

```
>> x=linspace(1.3,2.1,100);
>> y1=x;
>> y2=1+x.^(-1);
>> plot(x,y1,x,y2);
>> xlabel('X');
>> ylabel('y1 dan y2');
>> title('Grafik Fungsi');
>> legend('y1','y2');
>>
```

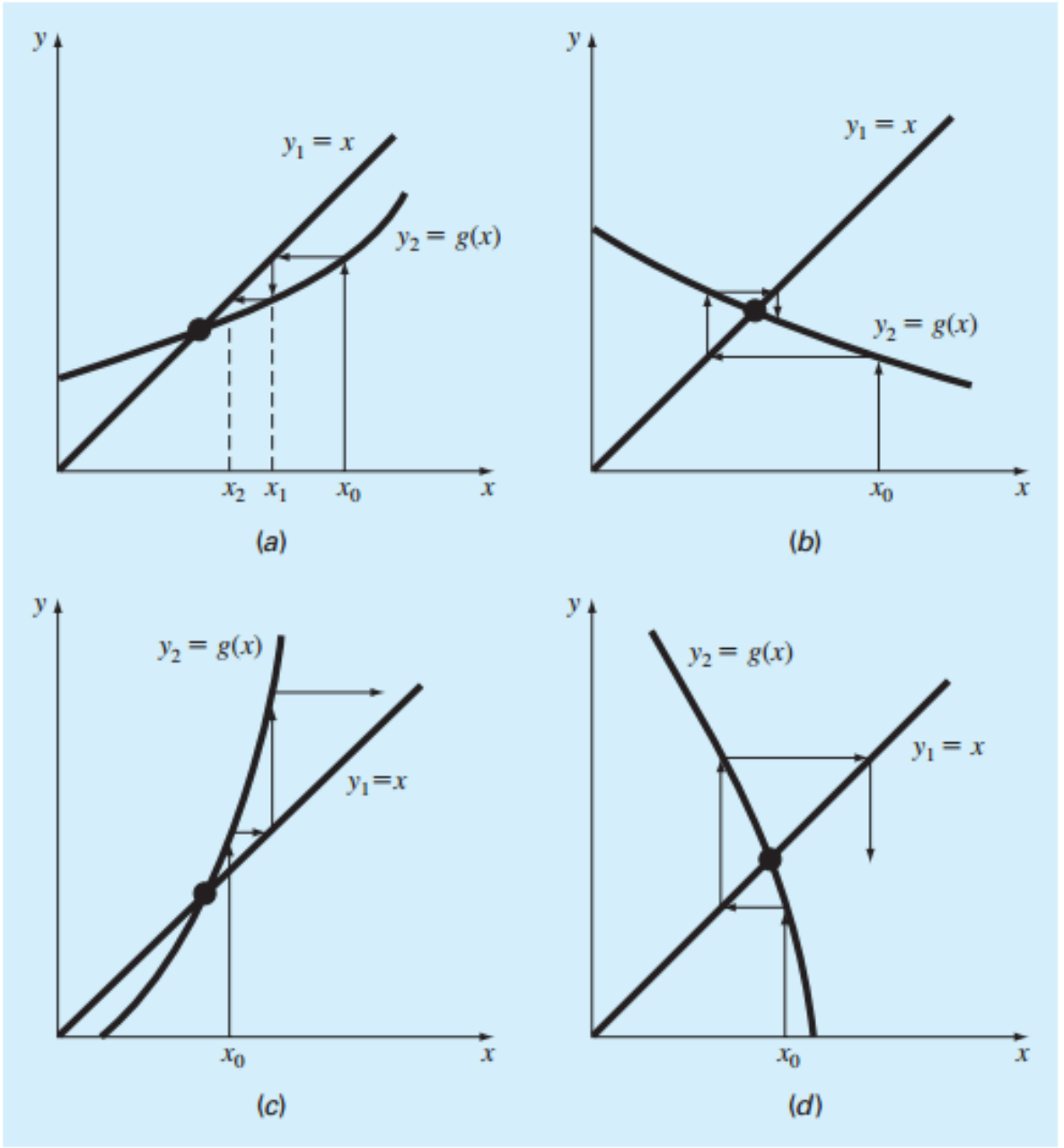


Ilustrasi $x_{n+1} = 1 + \frac{1}{x_n}$ (2)



Perhatikan Gambar berikut

Pada gambar manakah x^* ditemukan?



Metode Newton

- $f(x)$ adalah fungsi kontinyu pada $[a, b]$ dan mempunyai turunan $f'(x)$
- Berdasarkan ekspansi taylor maka

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

$\xi(x_k)$ adalah nilai diantara x dan x_k

Metode Newton

Jika x^* maka $f(x^*) = 0$. Jika f linear yaitu $f''(x) = 0$ maka akar-akar dapat dicari dengan cara sbb:

$$f(x^*) = f(x_k) + f'(x_k)(x^* - x_k) + f''(\xi(x_k))\frac{(x^* - x_k)^2}{2!}$$

$$0 = f(x_k) + f'(x_k)(x^* - x_k)$$

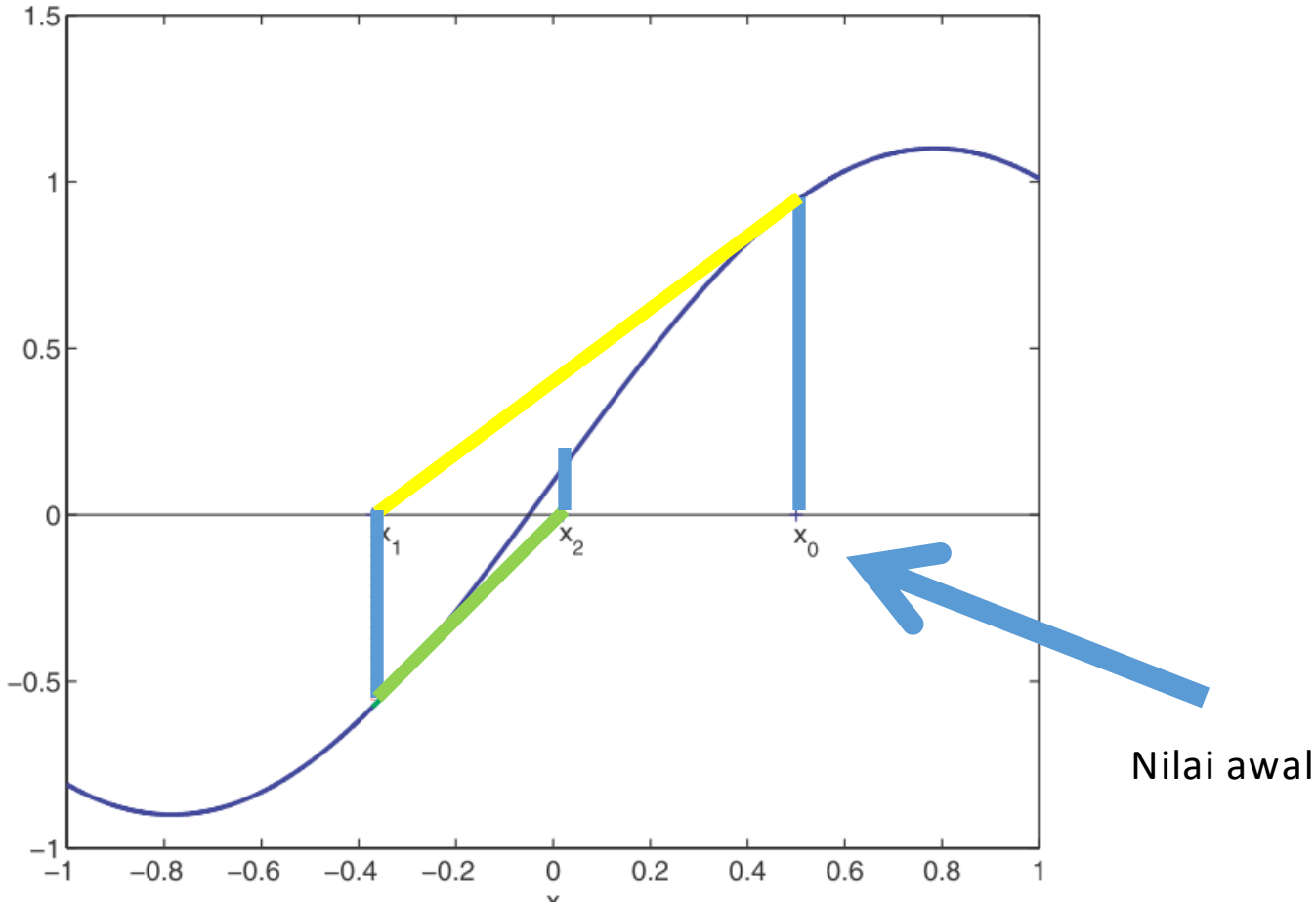
$$x^* = x_k - \frac{f(x_k)}{f'(x_k)}$$

Metode Newton

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

$$k = 1,2,3, \ldots$$

Interpretasi Geometri metode Newton



Nilai awal

Algoritma Metode Newton

Algorithm: Newton's Method.
Given a scalar differentiable function in one variable, $f(x)$:

1. Start from an *initial guess* x_0.
2. For $k = 0, 1, 2, \ldots,$ set

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},$$

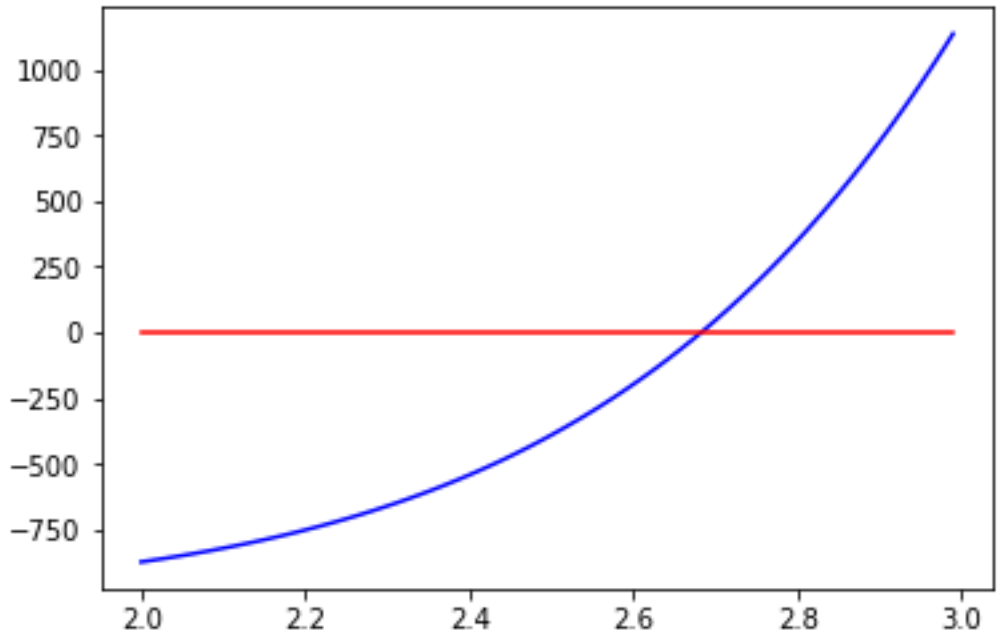
until x_{k+1} satisfies termination criteria.

Contoh: $f(x) = x^7 - 1000$

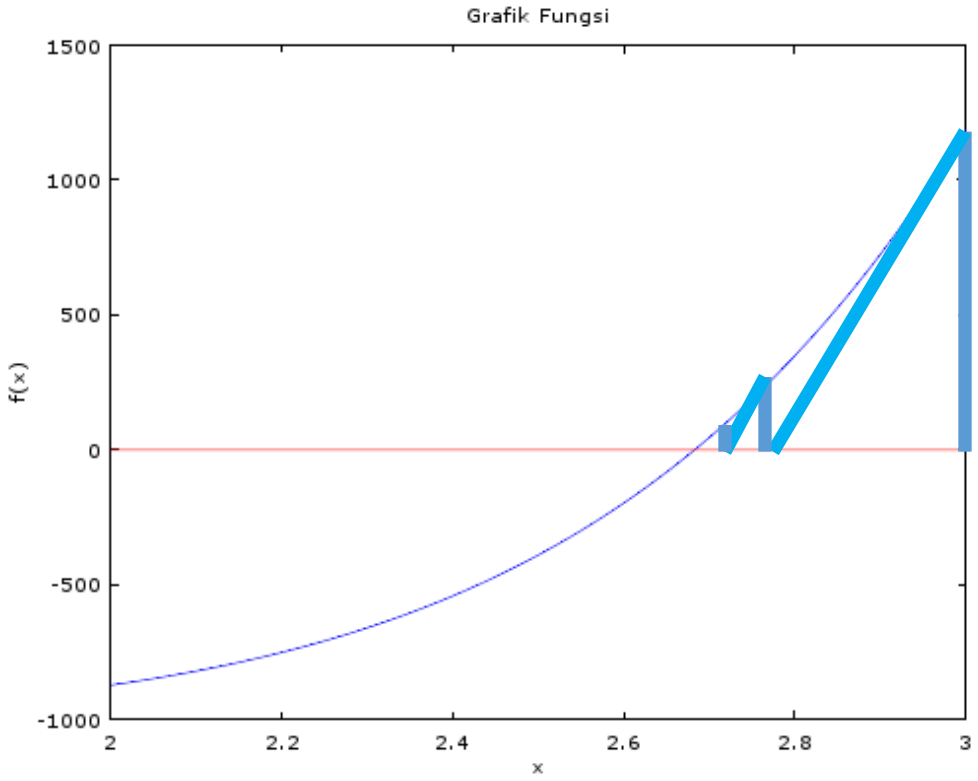
Metode Numerik

Code

```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(2,3,0.01)
y= x**7 -1000
plt.plot(x,y,'b-')
plt.plot(x,np.zeros(100),'r-')
plt.show()
```



Contoh: $f(x) = x^7 - 1000$



k	x(k)	fx(k)	f'x(k)	x(k+1)
0	3	1187	5103	2,767392
1	2,767392	243,0668	3144,285	2,690087
2	2,690087	19,4472	2652,75	2,682756
3	2,682756	0,158271	2609,67	2,682696
4	2,682696	1,07E-05	2609,316	2,682696
5	2,682696	0	2609,316	2,682696
6	2,682696	0	2609,316	2,682696
7	2,682696	0	2609,316	2,682696
8	2,682696	0	2609,316	2,682696

Latihan

1. Buat Programnya
2. Tentukan kriteria hentinya
3. Lakukan perubahan pada nilai awal x_0. Apa akibat perubahan nilai $x_0$tsb?

Metode Secan

Metode Numerik

- Metode newton memerlukan informasi turunan/derivatif (f'(x))
- Nilai turunan suatu fungsi ada kalanya susah dihitung
- Nilai turunan suatu fungsi dapat didekati dengan metode pendekatan beda hingga

$$f'(x) \cong \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

Metode Secan

- Metode newton dapat dimodifikasi, dengan menggantikan nilai f'(x) menggunakan metode beda hingga

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

$$f'(x) \cong \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

- Sehingga :

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$$

Algortima metode secan

Algorithm: Secant Method.
Given a scalar differentiable function in one variable, $f(x)$:

1. Start from two *initial guesses* x_0 and x_1.
2. For $k = 1, 2, \ldots$, set

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$$

until x_{k+1} satisfies termination criteria.

Catatan : Metode secan memerlukan 2 buah titik bantu x_0 dan x_1

Contoh metode secan

- $f(x) = x^7 - 1000$

k	x(k-1)	x(k)	fx(x-1)	fx(k)	x(k+1)
0	2	3	-872	1187	2,423507
1	3	2,423507	1187	-508,968	2,596515
2	2,423507	2,596515	-508,968	-204,325	2,712552
3	2,596515	2,712552	-204,325	80,55368	2,679741
4	2,712552	2,679741	80,55368	-7,68538	2,682598
5	2,679741	2,682598	-7,68538	-0,25398	2,682696
6	2,682598	2,682696	-0,25398	0,000841	2,682696
7	2,682696	2,682696	0,000841	-9,2E-08	2,682696
8	2,682696	2,682696	-9,2E-08	0	2,682696