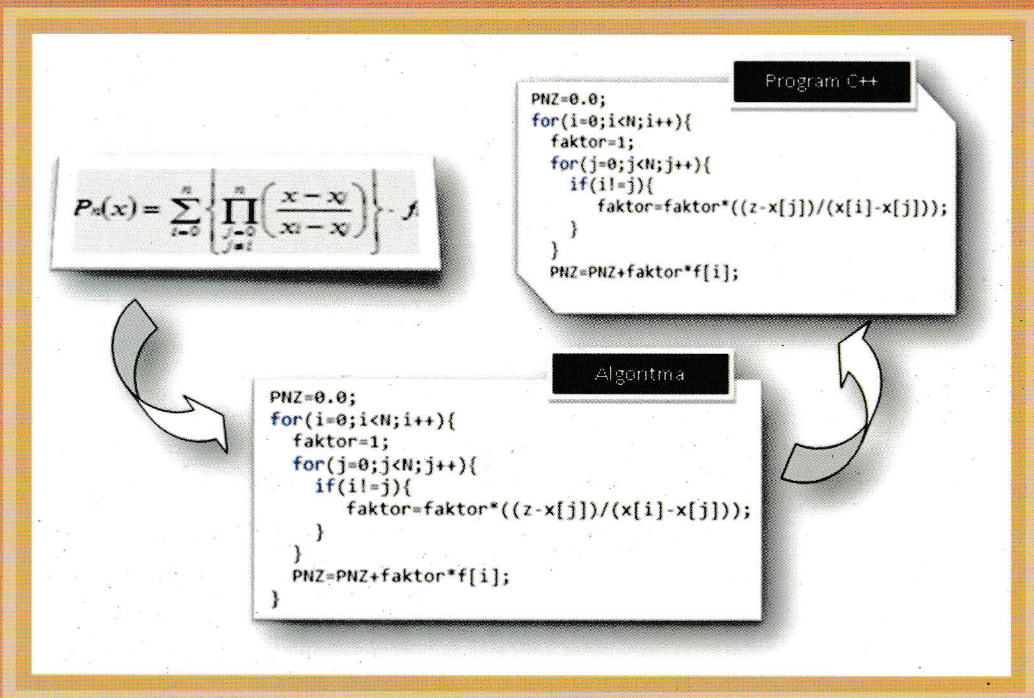
ANDALAS UNIVERSITY Press

METODE NUMERIK ALGORITMA DAN PEMOGRAMAN VISUAL C++



Syafii, ST, MT, PhD

METODE NUMERIK ALGORITMA DAN PEMOGRAMAN VISUAL C++

Penulis :
oleh : Syafii, ST, MT, PhD

Ilustrasi Sampul dan Penata Isi :
Dyans Fahrezionaldo
Safri Y

Andalas University Press
Jl. Situjuh No. 1, Padang 25129, Telp/Faks. : 0751-27066
email : cebitunand@ymail.com
facebook : AU Press (Andalas University Press)

Anggota :
Asosiasi Penerbit Perguruan Tinggi Indonesia (APPTI)

Cetakan :
I. Padang, 2014

ISBN : 978-602-8821-62-9

PRAKATA

Dengan memanjatkan puji dan syukur kehadirat Allah S.W.T yang telah memberikan karunianya, sehingga buku Metode Numerik Algoritma dan Pemograman C++ ini dapat terselesaikan.

Metode numerik merupakan salah satu kelompok mata kuliah dasar keahlian yang banyak diajarkan di Fakultas Teknik. Dengan analisis numerik persoalan yang rumit dibidang rekayasa dapat diselesaikan dengan mudah menggunakan komputer. Permasalahan yang penting dalam metode numerik adalah menemukan algoritma yang tepat dalam memecahkan persoalan dan iterpretasi algoritma tersebut ke dalam bahasa pemograman komputer. Dalam buku ini akan disajikan penyelesaian persamaan matematika untuk memodelkan sistem dan dilengkapi dengan algoritma dan program komputer berbasis Visual C++. Dalam setiap bab dilengkapi dengan contoh soal dan penyelesian, algoritma dan kode program visual C++, serta tampilan hasil running program. Pada akhir setiap bab diberikan soal latihan untuk menguji pemahaman pembaca terhadap materi yang disajikan untuk bab yang bersangkutan. Teknik pemograman berbasis windows form dan pemograman berorientasi objek juga diangkat dalam buku ini.

Pada kesempatan ini penulis menyampaikan terimakasih kepada Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) dan Universitas Andalas Press yang telah memfasilitasi penerbitan buku ajar ini. Ucapan terima kasih juga disampaikan kepada semua pihak yang telah membantu baik langsung maupun tidak langsung penyelesaian penulisan dan penerbitan buku ini. Akhir kata kami berharap semoga buku ini bermanfaat bagi kita sebagai akademisi khususnya dan bagi yang pembaca pada umumnya. Saran dan kritik yang sifatnya membantu kelengkapan dan menuju kesempurnaan untuk edisi selanjutnya dari semua pihak sangat diharapkan.

1 Juni 2014

Syafii, ST, MT, PhD

DAFTAR ISI

BAB I
ALGORITMA DAN PEMOGRAMAN

1.1 Pendahuluan

Dalam berbagai disiplin ilmu pengetahuan dan persoalan rekayasa (*engineering*), perilaku sistem dapat dimodelkan dengan persamaan matematika. Namun, tidak semua persoalan matematika dapat diselesaikan dengan metode analitik khususnya model matematika yang rumit. Metode analitik adalah metode penyelesaian persamaan matematika dengan rumus-rumus aljabar yang sudah baku. Metode analitik hanya unggul untuk sejumlah persoalan yang terbatas, yaitu persoalan yang memiliki tafsiran geometri sederhana. Bila metode analitik tidak dapat diterapkan, maka solusi persoalan masih dapat dicari dengan menggunakan metode numerik.

Metode numerik adalah teknik yang digunakan untuk memformulasikan persoalan matematik sehingga dapat dipecahkan dengan operasi perhitungan dan aritmetika biasa menggunakan operasi tambah, kurang, kali, dan bagi. Metode artinya cara, sedangkan numerik artinya angka. Jadi metode numerik secara harafiah berarti cara berhitung dengan menggunakan angka-angka. Metode numerik selalu dihubungkan dengan perkembangan komputer, karena perkembangan metode numerik itu sendiri berawal dari penemuan komputer. Dengan program komputer persoalan matematika yang rumit dapat diselesaikan secara numerik dengan mudah.

Perbedaan utama antara metode numerik dengan metode analitik terletak pada dua hal. Pertama, solusi dengan menggunakan metode numerik selalu berbentuk angka. Sedangkan metode analitik biasanya menghasilkan penyelesaian dalam bentuk fungsi matematik yang selanjutnya fungsi matematik tersebut dapat dievaluasi untuk menghasilkan nilai dalam bentuk angka. Metode numerik akan diperoleh solusi yang menghampiri atau mendekati solusi sebenarnya sehingga solusi numerik dinamakan juga solusi hampiran (approximation) atau solusi pendekatan, namun solusi hampiran dapat dibuat seteliti yang kita inginkan. Solusi hampiran akan ada selisih antara keduanya yang disebut dengan galat (error). Berbagai metode dan algoritma numerik terus dikembangkan untuk mendapatkan hasil yang mendekati nilai sebenarnya atau galat yang mendekati nol.

1.2 Algoritma

Definisi algoritma yang umum dijumpai di berbagai literatur adalah urutan logis langkah-langkah penyelesaian masalah. Berdasarkan algoritma selanjutnya dapat membantu pembuatan kode program komputer. Program komputer berisi urutan langkah-langkah penyelesaian masalah secara sistematis dapat ditulis menggunakan bahasa pemrograman seperti Bahasa Pascal, basic dan C. Urutan langkah-langkah penyelesaian masalah inilah yang dinamakan algoritma.

Sifat-sifat suatu algoritma adalah:

1. Setiap langkah harus terdefinisi dengan baik
2. Banyaknya operasi berhingga
3. Minimum satu keluaran
4. Bersifat umum

Cara penulisan terdiri dari komponen-komponen berikut:

1. Masukan
2. Langkah-langkah (proses)
3. Keluaran

Contoh soal:

1.1 Buat langkah-langkah perhitungan berulang berikut :

a. $D = a_1 + a_2 + a_3 + \ldots + a_n$
b. $H = b_1 \bullet b_2 \bullet b_3 \bullet \ldots \bullet b_n$

Jawab:

a. Masukan : a_i untuk $i = 1, 2, \ldots, n$

   Proses:

   1. $D = 0$
   2. Untuk $i = 1$ sampai dengan n

      $\hookrightarrow D = D + a_i$

   Keluaran: D

b. Masukan : b_i untuk $i = 1, 2, \ldots, n$

   Proses:

   1. $H = 1$
   2. Untuk $i = 1$ sampai dengan n

      $\hookrightarrow H = H * b_i$

   Keluaran: H

1.3 Jenis-Jenis Hitungan

Dalam penyelesaian persoalan matematika terdapat dua jenis perhitungan seperti berikut:

1. Hitungan langsung
2. Hitungan tidak langsung (iteratif)

Hitungan langsung dapat dilakukan sekiranya ada persamaan analitik untuk penyelesaian permasalah tersebut, seperti:

Tentukan akar persamaan kuadrat (polinom orde 2) berikut $ax^2 + bx + c = 0$, solusinya dapat dihitung langsung menggunakan persamaan berikut:

$$x_{1,2} = \frac{-b \pm \sqrt{D}}{2a}$$

Tetapi, untuk polinom derajat > 2, tidak terdapat rumus aljabar untuk menghitung akar polinom. Yang mungkin kita lakukan adalah dengan memanipulasi polinom, misalnya dengan memfaktorkan (atau menguraikan) polinom tersebut menjadi perkalian beberapa suku. Semakin tinggi derajat polinom, jelas semakin sukar memfaktorkannya. Cara lain yang mungkin dilakukan adalah perhitungan numerik secara tidak langsung.

Dalam perhitungan tidak langsung memiliki ciri-ciri berikut:

1. Ada tebakan awal
2. Skema iterasi
3. Kriteria penghentian

Proses dan algoritma penyelesaian :

Tentukan tebakan awal (x_0)

diperbaiki $x_1 = \ldots$

$x_2 = \ldots$

.

. $\Longrightarrow$ Harus konvergen

.

$x_k = \ldots$

Kriteria penghentian jika $x_{k+1} \approx x_k$ atau

$$\frac{x_{k+1} - x_k}{x_{k+1}} \leq 0.00001$$

Contoh perhitungan tidak langsung:

Tentukan akar persamaan kuadrat berikut:

$x^3 + 4x^2 + 1 = 0$

1.4 Galat

Dalam perhitungan metode numerik akan diperoleh penyelesaian yang menghampiri atau mendekati penyelesaian sebenarnya sehingga penyelesaian numerik dinamakan juga penyelesaian pendekatan. Dengan demikian akan ada selisih atara hasil perhitungan dengan nilai eksaknya yang disebut dengan galat (error). Semakin kecil galatnya, semakin akurat penyelsaian numerik yang didapatkan. Berbagai metode dan algoritma numerik terus dikembangkan untuk mendapatkan hasil yang mendekati nilai sebenarnya atau galat yang mendekati nol.

Timbunya galat disebabkan oleh:

(a) Galat pemotongan

Galat pemotongan muncul untuk nilai fungsi yang didekati dengan uraian Maclaurin seperti berikut:

$$e^x = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \frac{1}{4!}x^4 + \dots + \frac{1}{n!}x^n$$

$\cos x = x - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 - \dots + (-1)^{n/2}\frac{1}{n!}x^n$ (n genap)

Jelas kita tdk dapat memakai semua suku dalam deret tersebut, karena deretnya tak berhingga, sehingga suku yang dipotong menjadi galat pemotongan.

Jika:

$e^x = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3$, maka galat pemotongan adalah: $\frac{1}{4!}x^4 + \dots + \frac{1}{n!}x^n$

$\cos x = x - \frac{1}{2!}x^2$ (n genap), maka galat pemotongan adalah: $\frac{1}{4!}x^4 - \dots + (-1)^{n/2}\frac{1}{n!}x^n$

(b) Galat pembulatan

Keterbatasan komputer dalam menyajikan bilangan riil menghasilkan galat yang disebut galat pembulatan. Sebagai contoh 1/3 = 0.33333… tidak dapat dinyatakan secara tepat dalam hasil komputasi karena digit 3 panjangnya tidak terbatas. Jika diambil 4 digit angka, maka dibulatkan menjadi 0.3333 sehigga galat pembulatan adalah: 1/3 – 0.3333 = 0.00003333....

Galat dapat dihitungan menggunakan rumus berikut:

$$e_x = x - \bar{x}$$

dimana: e_x = galat

x = nilai eksak

$\bar{x}$ = taksiran nilai x

$| e_x | = | x - \bar{x} |$ disebut galat mutlak, sedangakan $e_x = \frac{|x-\bar{x}|}{x}$ disebut galat relatif atau dalam persentase:

$$e_x = \frac{|x-\bar{x}|}{x} \times 100\%$$

1.5 Dasar-dasar pemograman C++

Tahap selanjutnya adalah menerjemahkan algoritma ke dalam program komputer dengan menggunakan salah satu bahasa pemrograman. Saat ini terdapat banyak bahasa pemrograman seperti bahasa pascal, basic dan bahasa C. Pengguna bahasa pemrograman C dan C++ sangatlah banyak karenakan kemampuan bahasa C yang dianggap bisa dipakai dalam banyak bidang termasuk rekayasa dan terapan termasuk diantaranya pembuatan aplikasi. Kelebihan bahasa pemorgraman C++ adalah kemampuan untuk melakukan pemrograman berorientasi objek (Object Oriented Programming atau OOP). Pada OOP, data dan instruksi dibungkus (encapsulation) menjadi satu. Kesatuan ini disebut kelas (class) dan inisiasi kelas pada saat run-time disebut objek (object). Data di dalam objek hanya dapat diakses oleh instruksi yang ada didalam objek itu saja.

Setiap komputer dapat melakukan operasi-operasi dasar dalam pemrograman seperti operasi pembacaan data, operasi perbandingan, operasi aritmetika, dan sebagainya. Perkembangan teknologi komputer hanya merubah kecepatan, biaya, atau tingkat ketelitian tidak mengubah operasi-operasi dasar tersebut. Bahasa pemograman C++ termasuk dalam katagori bahasa pemograman tingkat menengah yang perlu diterjemahkan terlebih dahulu oleh sebuah translator bahasa (yang disebut kompilator atau compiler) ke dalam bahasa mesin sebelum akhirnya dieksekusi oleh CPU. Setiap instruksi dalam bahasa mesin merupakan kombinasi dari operasi dasar yang bersesuaian, dan menghasilkan efek yang sama pada setiap komputer. Oleh karena itu algoritma merupakan hal penting untuk dapat membuat kode pemograman komputer.

Struktur dasar pemograman C++ adalah:

```
# include….
main()
{
Perintah-perintah berdasarkan algoritma pemograman
}
```

1.5.1 Perintah Masukan dan Keluaran

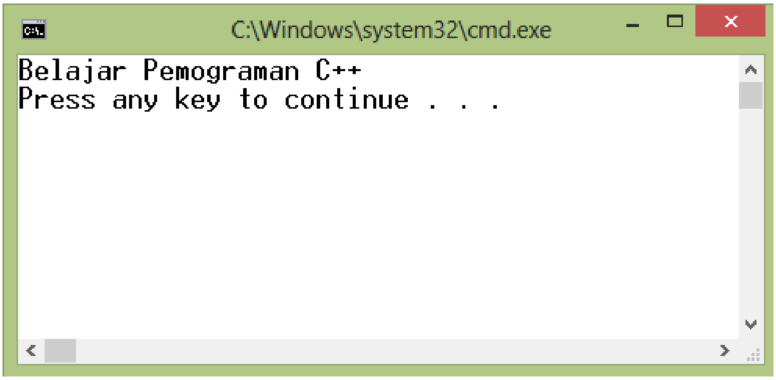
Masukan data dari keyboard menggunakan perintah cin>> , sedangkan keluaran ke tampilan dos (*console*) menggunakan perintah cout<<. Library yang digunakan iostream.h ditambahan dengan pernyataan using namespace std;. iostream adalah salah satu header file yang digunakan untuk fungsi input dan output

yang ada di C++ dan using namespace std adalah perintah yang digunakan untuk mendeklarasikan/ memberitahukan kepada compiler C++ bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std.

Perhatikan contoh berikut untuk menampilkan kalimat Belajar Pemograman C++ ke tampilan dos menggunakan pemograman Ms Visual C++ 2010.

```
#include "stdafx.h"
#include <iostream>
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
	cout<<"Belajar Pemograman C++"<<endl;
	return 0;
}
```

Apabila program ini dijalankan akan diperoleh hasil berikut:



Gambar 1.1 Tampilan hasil running program

Untuk memperbaiki tampilan numerik gunakan library iomanip.h dengan beberapa sintaks penting berikut:

Contoh:

double f =3.14159

cout << setprecision(5) << f ;

Apabila program dijalankan hasilnya adalah:

3.1416

Selanjutnya mengatur jumlah digit numerik yang ditampilakan dapat dilakukan dengan:

```
cout << setw(10);
cout << 77 << endl;
```

Apabila program dijalankan hasilnya adalah:

```
        77
```

Perintah berikut juga dapat digunakan untuk mencetak hasil tetap dengan lebar 5 digit dan rata kanan.

```
cout <<fixed <<setw(5) << right;
```

1.5.2 Operator Aritmatika

Berikut ini merupkan operator penting yang sering digunakan dalam membuat kode program C++:

Operator	Keterangan
+	pertambahan
*	perkalian
%	Sisa pembagian
-	Pengurangan
/	Pembagian

Selain itu pemogramman C++ juga mengenal operator logika berikut:

Operator	Keterangan
&&	Operator Logika AND
||	Operator Logika OR
!	Operator Logika NOT

Operator logika tersebut digunakan untuk menghubungkan dua buah operasi relasi menjadi sebuah ungkapan kondisi. Hasil dari operator logika ini menghasilkan nilai numerik 1 (True) atau 0 (False).

1.5.3 Perulangan

Statement perulangan yang paling sering digunakan adalah for. Statement for memiliki tiga parameter yang diapit dalam tanda kurung ( ), yaitu nilai awal, akhir perulangan dan penambahan/pengurangan. Contoh perulangan dengan statement for adalah:

```
for(i=1;i<=100;i++){
       Perintah yang diulang;
}
```

Dari contoh tersebut perintah yang akan diulang dilakukan sebanyak 100 kali dimulai dari i = 1 sampai dengan i = 100.

Statement perulangan lain adalah while. Statement while memiliki kondisi yang harus dipenuhi yang berada dalam tanda kurung ( ), jika kondisi bernilai benar akan dilakukan perintah yang diulang, jika tidak, maka keluar dari blok perulangan while. Contoh perulangan dengan statement while adalah:

```
Iterasi = 1;
while (Iterasi<=100){
       Perintah yang diulang;
       Iterasi=Iterasi+1;
}
```

Dari contoh di atas perintah yang akan diulang dilakukan sebanyak 100 kali dimulai dari Iterasi = 1 sampai dengan Iterasi = 100.

1.5.4 Aplikasi Windows Form

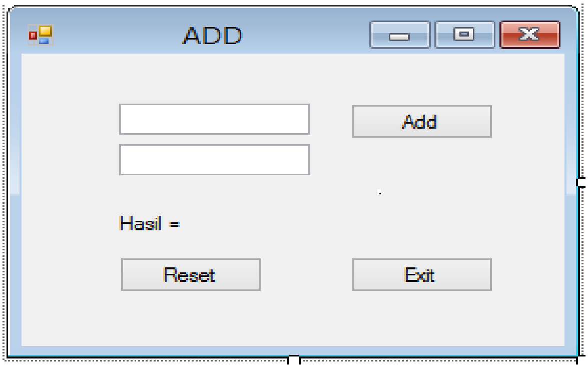
Untuk memperbaiki tampilan supaya lebih menarik dapat menggunakan Visual C++ Windows Form Application. Dalam pemograman bebasis windows form, toolbar-tolbar penting yang tersedia dalam pemograman Visual C++ adalah:

- textbox
- button
- label

Selanjutnya set properties via Properties window

Contoh:

1.2 Buatlah program C++ untuk melakukan perjumlahan sederhana dengan tampilan berikut:



Gambar 1.2 Tampilan Program Perjumlah

Jawaban:

Klik dan drag tolbar-tolbar berikut 2 textbox, 3 button dan 1 Label kedalam windows form dan atur posisinya seperti tampilan gambar 1.2

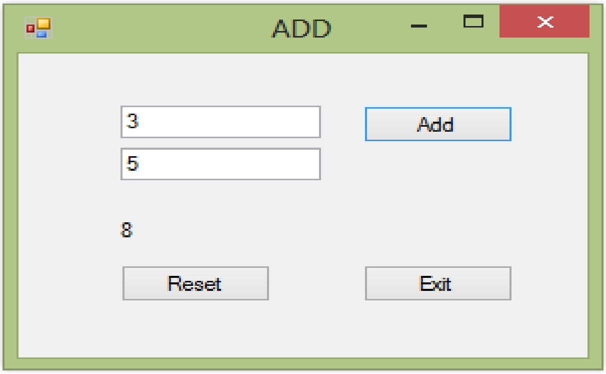
Double click Add, maka proses yang dilakukan adalah:

```
double a,b,c;
 a = System::Convert::ToDouble(textBox1->Text);
 b = System::Convert::ToDouble(textBox2->Text);
```

c = a + b;

label1->Text = System::Convert::ToString(c);

Hasil running



Gambar 1.3 Tampilan hasil running program

Untuk menghapus/ulangi, tambahkan kode program berikut pada editor button Reset:

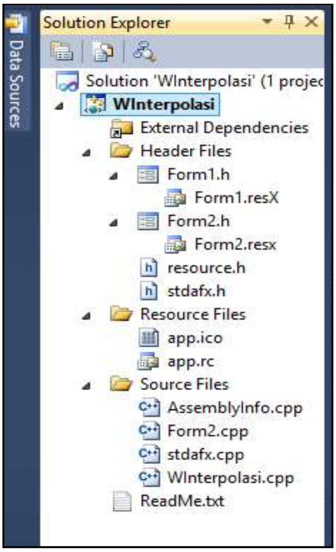
```
textBox1->Text = " ";
textBox2->Text = " ";
```

Selanjutnya desain form satu sebagai form aplikasi interpolasi dengan tampilan seperti gambar 1.5. Double click button Hitung, dan inputkan program interpolasi berikut:

```
int N=3, i,j;
double
x[10],f[10],PNZ,faktor,z;x[0]=2.0;f[0]=4.0;x[1]=3.0;f[1]=8.0;x[2]=5.0
;f[2]=25.0;
z=System::Convert::ToDouble(textBox1->Text);
PNZ=0.0;
for(i=0;i<N;i++){
   faktor=1;
   for(j=0;j<N;j++){
         if(i!=j){
               faktor=faktor*((z-x[j])/(x[i]-x[j]));
         }
   }
```

```
  PNZ=PNZ+faktor*f[i];
}
textBox2->Text = System::Convert::ToString(PNZ);
```

z adalah nilai yang akan dicari interpolasinya diantara x[0] = 2 sampai dengan x[2] = 5, hasilnya disimpan sebagai variabel PNZ.



Gambar 1.4 Solution Explorer untuk aplikasi banyak form

Aplikasi dengan banyak Wimdows Form dapat dilakukan dengan klik kanan pada nama project, selanjutnya pilih add New Item dan Windows Form. Setelah diberi nama Form2, akan muncul File Form2.h dan Form2.cpp pada Solution Explorer seperti gambar 1.5.

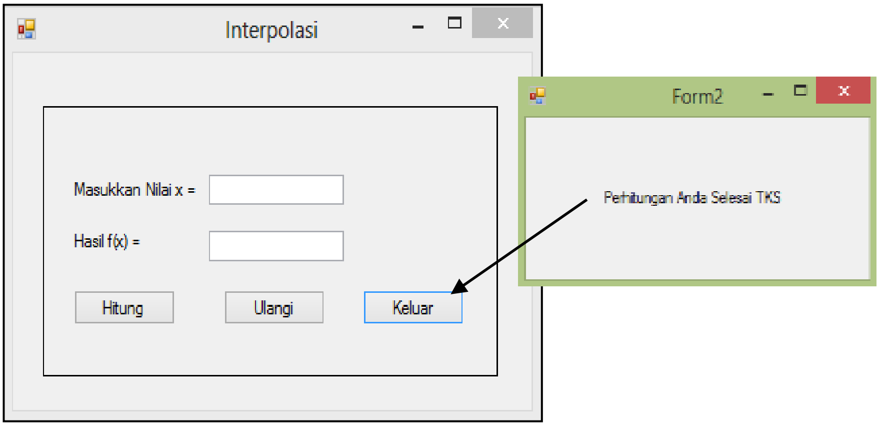
Sebagai contoh form dua akan muncul saat button Keluar ditekan. Kode Program untuk menampilkan form dua adalah:

```
Form2^ form2 = gcnew Form2();
form2->Show();
```

Program ini diketikan dalam `Void button3_Click berikut:`

```
private: System::Void button3_Click(System::Object^  sender,
System::EventArgs^  e) {
       Kode Program untuk menampilkan form dua;
}
```

Sebelumnya tambahkan #include “Form2.h” pada kode editor form1.h. Setelah program dijalankan, hasilnya adalah seperti tampilan berikut:



Gambar 1.5 Tampilan aplikasi banyak form

Untuk keluar dari aplikasi ketika button Keluar ditekan, maka gunakan perintah berikut:

```
Application::Exit();
```

Sehingga program dalam `Void button3_Click menjadi:`

```
private: System::Void button3_Click(System::Object^  sender,
System::EventArgs^  e) {
                   Application::Exit();
             }
```

BAB II
AKAR PERSAMAAN TAK LINIER

Persoalan mencari akar persamaan tak linear dapat dirumuskan secara singkat sebagai berikut:

$$f(x) = 0$$

fungsi x dapat berupa polinom berikut:

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

$$f\,(x) = a + bx + cx^2$$

$$f\,(x) = a_0 + a_1 x + a_2 x^2 + \ldots\ldots + a_n x^n$$

Nilai *x* adalah akar persamaan tak linear sedemikian sehingga *f(akar)* sama dengan nol. Jika tidak terdapat rumus aljabar untuk menghitung akar polinom, maka dilakukan manipulasi polinom, misalnya dengan memfaktorkan (atau menguraikan) polinom tersebut menjadi perkalian beberapa suku. Semakin tinggi derajat polinom, semakin sulit untuk memfaktorkannya, sehinga sulit diselesaikan dengan metode analitik. Bila metode analitik tidak dapat diterapkan, maka solusi persoalan masih dapat dicari dengan menggunakan metode numerik.

Persamaan-persamaan matematika yang akan dibahas adalah persamaan tak linear seperti polinom (derajat $\geq$ 2), trigonometri (sinus, cosinus), eksponensial, logaritma, dan fungsi transenden lainnya. Dalam penyelesai numerik menggunakan penyelesaian iteratif. Secara umum, metode numerik pencarian akar persamaan tak linear tersebut dapat dikelompokkan menjadi dua yaitu:

1. Metode tertutup (bracketing method)

Metode yang termasuk ke dalam golongan ini mencari akar di dalam selang [a, b]. Kelebihan metode ini selalu konvergen (menuju) ke akar, karena selang [a, b] sudah dipastikan berisi minimal satu buah akar. Sedangkan kekurangan metode ini proses iterasinya lebih panjang dan lama.

2. Metode terbuka

Metode terbuka tidak memerlukan selang [a, b] yang mengandung akar. Proses mencari akar dimulai dari sembarang nilai tebakan awal, selanjutnya diperbaiki untuk menghitung hampiran akar yang baru. Jumlah iterasi dan waktu hitung sangat tergantung dari nilai tebakan awal dan ada kemungkinan akar yang baru menjauhi (divergen). Kelebihan metode ini lebih sedikit iterasi dan singkat waktu hitungnya.

2.1 Penentuan Tebakan Awal

Penentuan tebakan awal yang mengandung akar diperlukan dalam penyelesaian metode tertutup. Penentukan dimana letak akar dapat dilakukan dengan dua cara, yaitu:

1. Tabulasi

Cara ini dilakukan dengan membuat tabel nilai x dan f (x) seperti berikut:

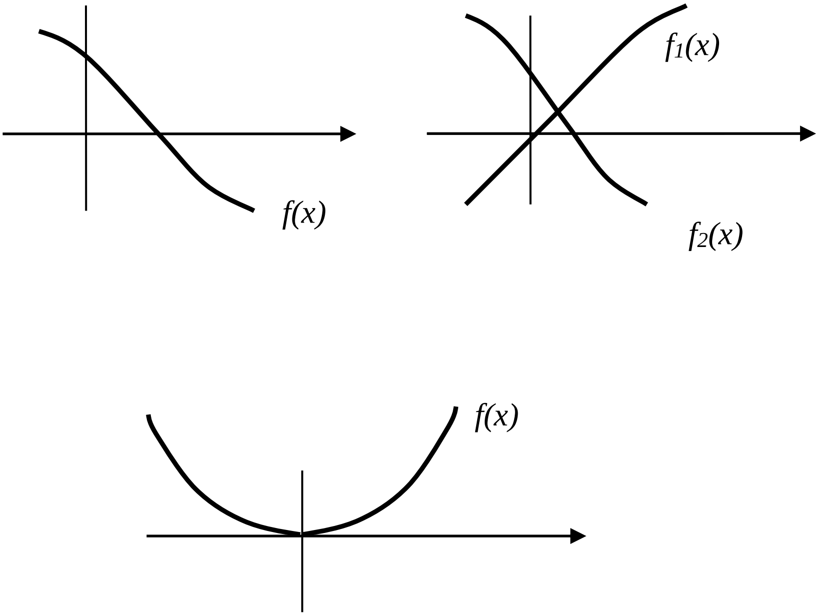
x	f (x)
x_0	$f(x_0)>0$
x_1	$f(x_1)<0$
.	
x_n	$f(x_n)<0$

x_{n+1}	$f(x_{n+1})>0$

Selanjutya diperiksa nilai f(x) apakah ada perbedaan tanda, jika ada maka $f(x_k).f(x_{k+1}) < 0$ artinya diantara x_k dan x_{k+1} terdapat akar. Maka ambil tebakan awal $a = x_k$ dan $b = x_{k+1}$ dimana [a, b] merupkan selang yang mengandung akar. Suatu fungsi tak linear bisa saja tidak memiliki akar, jika fungsi tersebut tidak melewati sumbu x.

2. Grafik

   Cara grafik dilakukan dengan menggambar kurva fungsi f(x) dan memprediksi titik perpotongan dengan sumbu x atau titik perpotongan dua fungsi sebagai selang yang menggandung akar.



Gambar 2.1 Contoh gambar kurva persamaan tak linear

Contoh :

1. Tentukan selang yang megandung akar dari persamaan tak linear berikut:

   $x - \cos x = 0$

   Jawab.

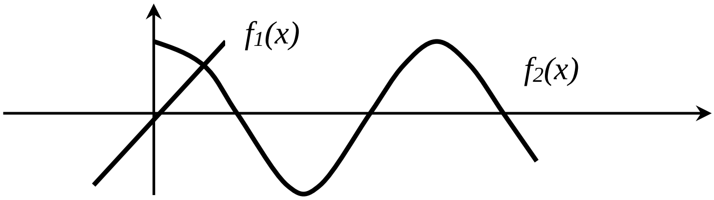
   $x - \cos x = 0$

   $x = \cos x$

   $f_1(x) = x$

   $f_2(x) = \cos x$

   Jika kedua fungsi ini dilukiskan kurvanya pada bidang x,y diperoleh:



Gambar 2.2 Persamaan $f_1(x) = x$ dan $f_2(x) = \cos x$

Selanjut buat tabulasi untuk perkiran titik yang mengandung akar dari grafik diatas, misalnya diambil tittik x = 0 dan x = 1, diperoleh:

x	$f_1(x)$	$f_2(x)$	$f(x)= f_1(x)- f_2(x)$
0	0	1	-1
1	1	0.54	0.44

Dari tabel tersebut terlihat bahwa f(0) dan f(1) terdapat berbendaan tanda sehingga $f(0)\ .\ f(1) < 0$, maka ada akar antara 0 dan 1. Oleh karena itu tebakan awal adalah 0 dan 1.

2. Tentukan selang yang megandung akar dari persamaan tak linear berikut:

$$e^{x} - x^{2} = 0$$

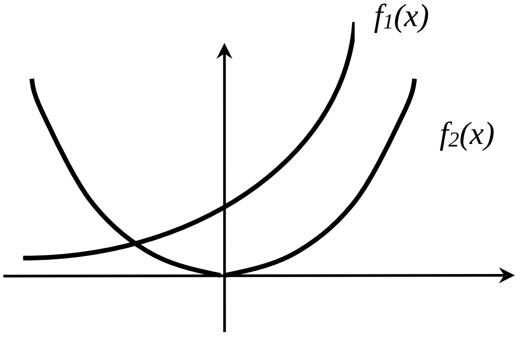
Jawab:

$$e^{x} - x^{2} = 0$$

$$e^{x} = x^{2}$$

$$f_1(x) = e^{x}$$

$$f_2(x) = x^{2}$$



Gambar 2.3 Persamaan $f_1(x) = e^x$ dan $f_2(x) = x^2$

Selanjut buat tabulasi untuk perkiran titik yang mengandung akar dari grafik diatas, misalnya diambil tittik x = 0 dan x = -1, diperoleh:

x	$f_1(x)$	$f_2(x)$	$f(x)= f_1(x)- f_2(x)$
0	1	0	1
-1	$1/e$ =0.368	1	-0.632

Dari tabel tersebut terlihat bahwa f(0) dan f(-1) terdapat berbendaan tanda sehingga f*(0)* . *f(-1)* < 0, maka ada akar antara 0 dan -1. Oleh karena itu tebakan awal adalah 0 dan -1.

3. Tentukan selang yang megandung akar dari persamaan tak linear berikut:

$$e^{x} + x^{2} = 0$$

Jawab:

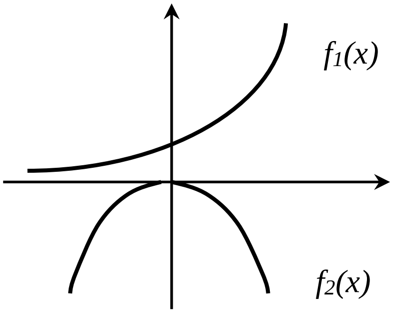
$$e^{x} + x^{2} = 0$$

$$e^{x} = -x^{2}$$

$$f_1(x) = e^{x}$$

$$f_2(x) = -x^{2}$$

Jika kedua fungsi ini dilukiskan kurvanya pada bidang x,y diperoleh:



Gambar 2.4 Persamaan $f_1(x) = e^{x}$ dan $f_2(x) = -x^{2}$

Dari grafik diatas terlihat bahwa kedua fungsi tersebut tidak berpotongan sehigga tidak ada akar karena f(x) tidak pernah melalui sumbu x.

4. Tentukan selang yang megandung akar dari persamaan tak linear berikut:

$$\cos x - \sin x = 0$$

Jawab:

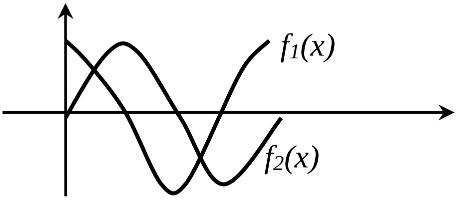
$$\cos x - \sin x = 0$$

$$\cos x = \sin x$$

$$f_1(x) = \cos x$$

$$f_2(x) = \sin x$$

Jika kedua fungsi ini dilukiskan kurvanya pada bidang x,y diperoleh:



Gambar 2.5 Persamaan $f_1(x) = \cos x$ dan $f_2(x) = \sin x$

Dari grafik diatas terlihat bahwa kedua fungsi tersebut saling berpotongan pada banyak titik sehingga akar tidak terhitung banyaknya.

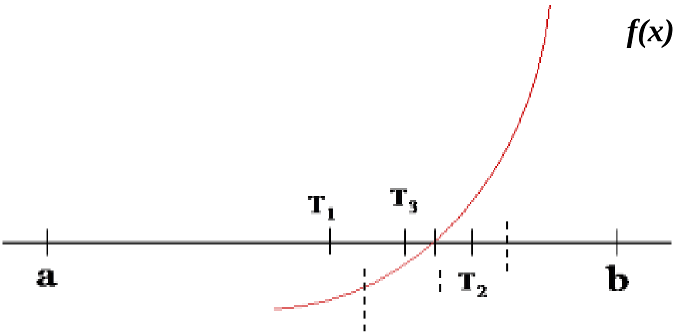
2.2. Langkah-langkah dan Proses Iterasi

Langkah selanjutnya adalah penyelesaian akar persamaan tak linear melalui proses iterasi berikut:

1. Metode tertutup (bracketing method) terdiri dari:
   - Metode bagi dua (bisection)
   - Metode Posisi Palsu (false position)
2. Metode terbuka terdiri dari:
   - Metode Newton Rapson
   - Metode Secant

2.2.1 Metoda Bagi Dua

Setelah diperoleh tebakan awal yang mengandung akar dengan batas bawah a dan batas atas b sehingga $f(a)f(b) < 0$, selanjutnya dilakukan proses iterasi. Pada setiap iterasi, selang [a, b] dibagi dua sehingga diperoleh titik tengah T. Selanjutnya diperiksa terhadap dua buah selang baru yang berukuran sama, yaitu selang [a, T] dan [T, b] yang mengandung akar. Selang yang mengandung akar adalah selang yang menghasikan perkalian fungsinya kecil dari nol atau ada perubahan tanda. Jika $f(a)f(T) < 0$, berarti akar terletak antara [a, T] atau sebaliknya jika $f(T)f(b) < 0$, berarti akar terletak diantara [T, b]. Proses tersebut diilustrasikan seperti pada gambar 2.6.



Gambar 2.6 Penentuan akar dengan metode bagi dua

Proses iterasi dapat diringkas seperti berikut:

1. Tentukan tebakan awal a dan b, dimana f(a) . f(b) <0, kemudian hitung

$$T = \frac{a+b}{2}$$

2. Menentukan selang yang mengandung akar jika :

   f(a) . f(T) < 0 maka akar pada (a,T), selanjutnya geser b $\rightarrow$ T

   f(a) . f(T) = 0 maka akar = T

   f(a) . f(T) > 0 Maka akar pada (T,b), Selanjutnya geser a $\rightarrow$ T

3. Selanjutnya periksa kriteria penghentian apakah f(T) $\cong$ 0 melalui

$$|a - T| \rightarrow 0$$

$$|T - b| \rightarrow 0$$

Jika hasil tersebut sudah mendekati nol maka proses iterasi dihentikan dan akar adalah titik tengah yang terakhir (T).

Contoh Soal:

2.1 Tentukan akar persamaan tak linear berikut:

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

menggunakan metode bagi dua?

Jawab:

Sebelum proses iterasi dilakukan penentuan tebakan awal a dan b seperti berikut:

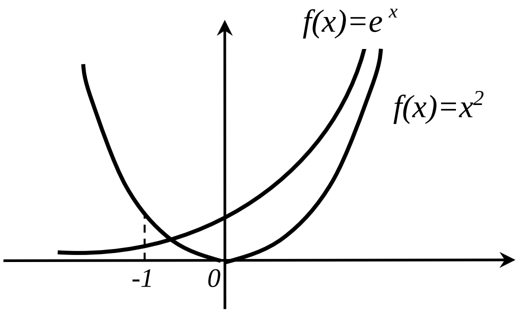
$$f(x) = x^2 - e^x = 0$$

persamaan ini dapat ditulis $x^2 = e^x$, sehingga diperoleh dua fungsi berikut:

$$f_1(x) = e^x$$

$$f_2(x) = x^2$$

Bentuk grafik kedua fungsi tersebut adalah:



Gambar 2.7 Penentuan tebakan awal $f(x) = x^2 - e^x$

Dari grafik tersebut dapat diprediksi tebakan awal berada pada selang [-1, 0], sehingga sebagai tebakan awal diambil a = -1 dan b =0.

Iterasi 1:

Tebakan awal

$a = -1 \quad \rightarrow \quad f(-1) = 1 - e^{-1} \quad > 0$

$b = 0 \quad \rightarrow \quad f(0) = 0 - 1 \quad < 0$

Karena f(-1) . f(0) < 0, maka diantara -1 dan 0 ada nilai akar. Selanjutnya dihitung titik tengah menggunakan persamaan berikut:

$$T = \frac{a+b}{2} = \frac{-1+0}{2} = -0.5$$

Nilai fungsi untuk titik tengah adalah

$$f(-0.5) = (-0.5)^2 - e^{-0.5} = 0.25 - \frac{1}{\sqrt{e}} = < 0$$

Selanjutnya periksa selang baru yang mengandung akar melalui perkalian fungsi berikut:

f(-0.5) . f(0) > 0, berarti diantara -0.5 dan 0 tidak mengandung akar.

f(-1) . f(-0.5) < 0 sehingga akar berada diantara (-1 dan –0.5)

Oleh karena itu geser b menjadi -0.5, lanjutkan ke iterasi berikutnya

Iterasi 2 :

Nilai batas yang baru adalah:

a = -1

b = -0.5

Selanjutnya dihitung kembali titik tengah menggunakan persamaan berikut:

$$T = \frac{-1-0.5}{2} = \frac{-1.5}{2} = -0.75$$

Nilai fungsi untuk titik tengah adalah

$$f(-0.75) = (-0.75)^2 - e^{-0.75} > 0$$

Selanjutnya periksa selang baru yang mengandung akar melalui perkalian fungsi berikut:

f(-1) . f(-0.75) > 0, berarti diantara -1 dan -0.75 tidak mengandung akar.

f(-0.75) . f(-0.5) < 0, maka akar berada pada (-0.75 dan –0.5)

Oleh karena itu geser a menjadi -0.75, lanjutkan ke iterasi berikutnya

Iterasi 3 :

Nilai batas yang baru adalah:

a = -0.75

b = -0.5

Selanjutnya dihitung kembali titik tengah menggunakan persamaan berikut:

$$T = \frac{-0.75 + (-0.5)}{2} = \frac{-1.25}{2} = -0.625$$

dan seterusnya sampai iterasi ke 16, pada iterasi 16 diperoleh :

T = -0.703468

Proses berhenti sampai selisih (Epsilon) = 10^{-4}, maka akar adalah titik tengah terahir yaitu T = -0.703468

Dari langkah-langkah diatas dapat disusun Algoritma metoda bagi dua sebagai berikut:

Var : a = ujung kiri
b = ujung kanan
T = titik tengah

Masukan : f(x)
Tebakan awal a dan b (syarat f(a).f(b) < 0)
Eps = 0.00001

Keluaran : akar

Langkah-langkah:

1. $T = \frac{a+b}{2}$
2. jika f(a).f(T) < 0 maka b = T
   jika tidak a = T
3. jika |b – a| < Eps maka akar = T, selesai
4. Kembali ke langkah 1

Selanjutnya adalah menerjemahkan algoritma ke dalam program komputer dengan menggunakan pemrograman Visual C++ dapat ditulis seperti berikut:

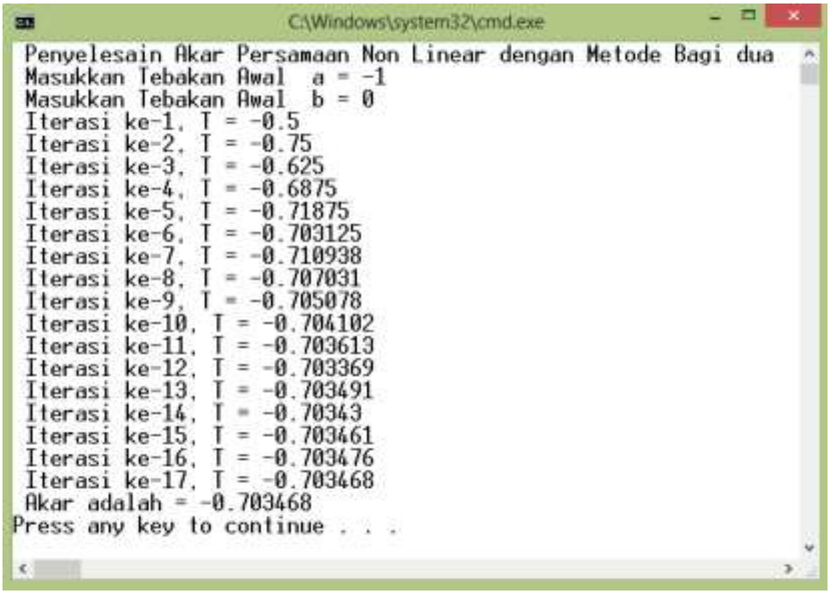
Program C++

```
float a, b, T, eps=0.00001, Mak=20;
int Iter;
float f(float x);
cout<<" Penyelesain Akar Persamaan Non Linear dengan Metode Bagi
dua"<<endl;
cout<<" Masukkan Tebakan Awal  a = ";
cin>>a;
cout<<" Masukkan Tebakan Awal  b = ";
cin>>b;
Iter=0;
while (Iter<Mak){
       T=(a+b)/2;
       if(f(T)==0)break;
       if ((f(a)*f(T))<0)
              b=T;
       else
              a=T;
       if(abs(b-a)<eps) break;
       Iter=Iter+1;
       cout<<" Iterasi ke-"<<Iter<<", T = "<<T<<endl;
}
cout<<" Iterasi ke-"<<Iter+1<<", T = "<<T<<endl;
cout<<" Akar adalah = "<<T<<endl;
```

Jika fungsi yang diselesaikan adalah $f(x) = x^2 - e^x = 0$, maka tuliskan fungsi tersebut pada akhir program:

```
float f(float x)
{
       float f=2*x*x-exp(2*x);;
       return f;
}
```

Selanjutnya compile dan jalankan program, diperoleh hasil seperti berikut:



Gambar 2.8 Hasil running program metode bagi dua f(x) = $x^2 - e^x$

2.2.2 Metoda Posisi Palsu

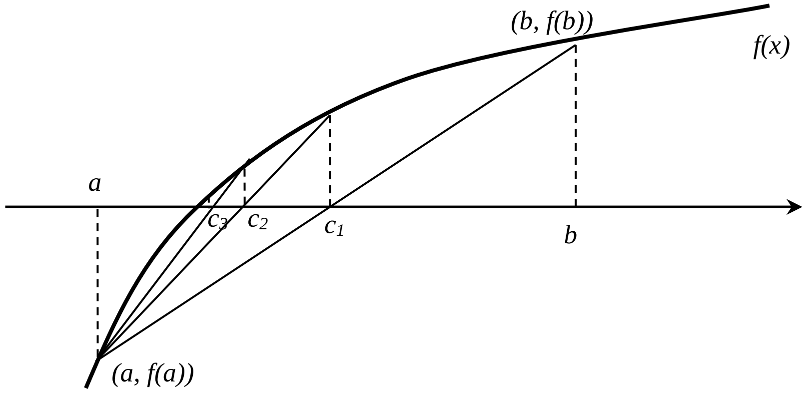
Metode posisi palsu menggunakan titik perpotongan garis lurus dengan sumbu x untuk menentukan wilayah akar. Garis lurus yang menghubungkan titik (a, f(a)) dan (b, f(b)) tadi seolah-olah berlaku menggantikan kurva f(x) dan memberikan posisi palsu dari akar. Perpotongan garis lurus dengan sumbu-x merupakan taksiran akar yang diperbaiki. Perhatikan gambar 2.9 proses penentuan akar dengan metode posisi palsu.

Persamaan garis lurus yang melalui titik (a,f(a)) dan (b, (fb)) adalah:

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

Perpotongan garis lurus dengan sumbu x dimisalkan pada titik c. Untuk mencari titik c, maka substitusikan nilai y = 0 dan x = c:

$$\frac{0-f(b)}{f(a)-f(b)} = \frac{c-b}{a-b}$$



Gambar 2.9 Penentuan akar dengan metode posisi palsu

Sehingga diperoleh titik c berikut:

$$c = b - f(b)\,\frac{a-b}{f(a)-f(b)} \quad \text{atau}$$

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

Nilai c untuk iterasi pertama ditulis c_1 selanjutnya didiperiksa selang baru yang mengandung akar apakah antara a dan c atau c dan b.

Jika f(a) . f(c) < 0 maka akar pada (a, c), selanjutnya geser b $\rightarrow$ c

Jika f(a) . f(c) = 0 maka akar = c

Jika f(a) . f(c) > 0 Maka akar pada (c, b), Selanjutnya geser a $\rightarrow$ c

Selanjunya proses yang sama dilakukan untuk memperoleh $c_{2,}$ c_3, ...,c_n. Kriteria penghentian didapat menggunakan selisih lebar selang, karena salah satu batas selalu bernilai konstan. Sebagai ganti bisa dilakukan dengan menguji nilai fungsi akar apakah sudah mendekati nol. Jika nilai fungsi akar sudah mendekati nol maka proses iterasi dihentikan dan akar adalah titik perpotongan garis lurus dengan sumbu x yang terakhir (c_n).

Contoh soal:

2.2 Tentukan akar persamaan tak linear berikut:

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

menggunakan metode posisi palsu?

Jawab:

Sebelum proses iterasi dilakukan penentuan tebakan awal a dan b seperti seperti contoh soal 2.1. Karena persamaannya sama maka tebakan awal adalah a = -1 dan b = 0.

Iterasi 1.

Tebakan awal

$a = -1 \rightarrow f(-1) = 1 - e^{-1} = 0.632$ atau > 0

$b = 0 \rightarrow f(0) = 0 - 1 = -1$ atau < 0

Selanjutnya hitung c:

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

$$c = 0 - (-1) \frac{-1-0}{(-1) - 0.632} = -0.613$$

Setelah nilai c diperoleh, selanjutnya menggitung f(c)

$$f(-0.613) = 0.613^2 - e^{-0.613}$$
$$= -0.166$$

Selanjutnya periksa selang baru yang mengandung akar:

Karena f(a) f(c) < 0 maka akar pada antara a dan c yaitu (-1,-0.163). Oleh karena itu geser b menjadi -0.163, lanjutkan ke iterasi berikutnya

Iterasi 2

Nilai batas yang baru adalah:

a = -1 f(a) = 0.632

b = -0.613 f(b) = -0.166

Selanjutnya dihitung kembali titik titik potong dengan sumbu x menggunakan persamaan berikut:

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

$$c = -0.613 - (-0.166)\ \frac{(-0.613)-(-1)}{(-0.1665)-0.6321} = -0.693$$

Nilai fungsi untuk titik c yang baru adalah adalah

$$f(-0.693) = 0.693^2 - e^{-0.693}$$

$$= -0.02$$

Karena f(a) f(c) < 0 maka akar pada (-1,-0. 693). Oleh karena itu geser b menjadi -0. 693, lanjutkan ke iterasi berikutnya

Iterasi 3

Nilai batas yang baru adalah:

a = -1 f(a) = 0.632

b = -0. 693 f(b) = -0.02

Selanjutnya dihitung kembali titik titik potong dengan sumbu x menggunakan persamaan berikut:

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

$$c = -0.693 - (-0.02)\ \frac{(-1)-(-0.693)}{(-0.02)-0.632} = -0.702$$

Nilai fungsi untuk titik c yang baru adalah adalah

$$f(-0.702) = 0.702^2 - e^{-0.702}$$

$$= -0.003$$

Karena f(a) f(c) < 0 maka akar pada (-1,-0.702). Oleh karena itu geser b menjadi -0.702, lanjutkan ke iterasi berikutnya.

Sampai iterasi ke 6, diperoleh :

c = -0.703466

Jika proses berhenti sampai nilai fungsi akar (Epsilon) = 10^{-4}, maka akar = c = -0.703466

Dari langkah-langkah diatas dapat disusun Algoritma metoda bagi dua sebagai Langkah-langkah:

1. $c = b - f(b)\ \dfrac{b-a}{f(b)-f(a)}$
2. Jika f(a).f(c)<0, maka b=c

   Jika tidak a=c
3. |f(c)| < Eps, maka akar = c, selesai
4. kembali ke 1

Terjemahkan algoritma posisi palsu dalam program komputer dengan menggunakan pemrograman Visual C++ dapat ditulis seperti berikut:

<u>Program C++</u>

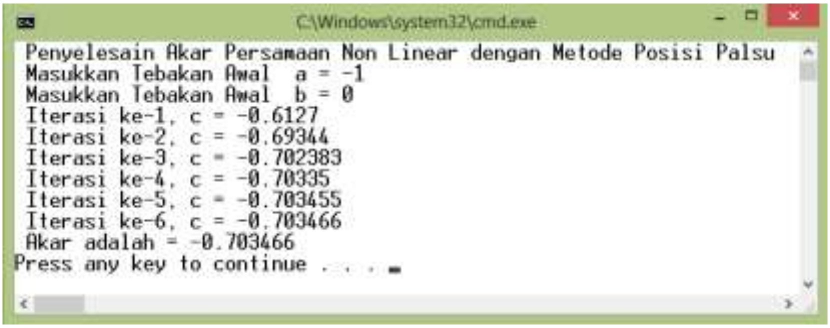
```
float a, b, c, eps=0.00001, Mak=20;
int Iter;
float f(float x);
cout<<" Penyelesain Akar Persamaan Non Linear dengan Metode Posisi
Palsu"<<endl;
cout<<" Masukkan Tebakan Awal  a = ";
cin>>a;
```

```
cout<<" Masukkan Tebakan Awal  b = ";
cin>>b;
Iter=0;
while (Iter<Mak){
       c=b-f(b)*(b-a)/(f(b)-f(a));
       if ((f(a)*f(c))<0)
              b=c;
       else
              a=c;
       if(abs(f(c))<eps) break;
       Iter=Iter+1;
       cout<<" Iterasi ke-"<<Iter<<", c = "<<c<<endl;
}
cout<<" Iterasi ke-"<<Iter+1<<", c = "<<c<<endl;
cout<<" Akar adalah = "<<c<<endl;
```

Jika fungsi yang diselesaikan adalah $f(x) = x^2 - e^x = 0$, sama dengan prosedur penyelesaian metode bagi dua maka tuliskan fungsi tersebut pada akhir program seperti berikut:

```
float f(float x)
{
       float f=x*x-exp(x);
       return f;
}
```

Selanjutnya compile dan jalankan program, diperoleh hasil seperti berikut:



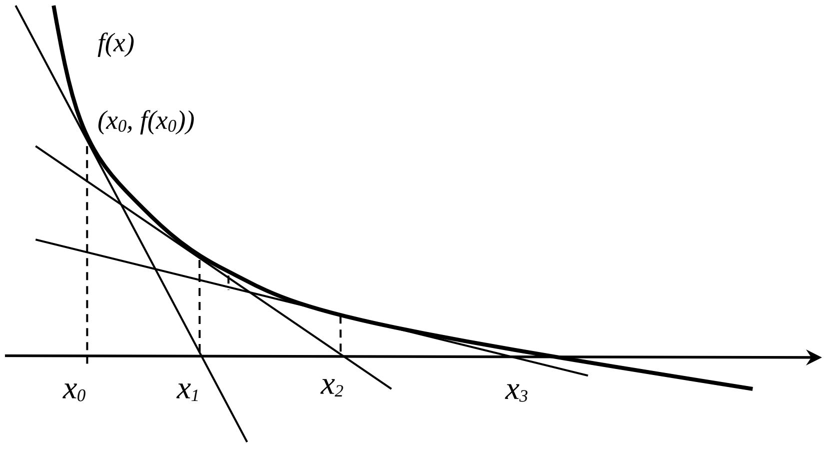
Gambar 2.10 Hasil running program metode posisi palsu untuk $f(x) = x^2 - e^x$

Soal Latihan:

1. Tentukan akar persamaan $f(x) = x^2 - e^x$ dengan tebakan awal a = -2 dan b = 1 menggunakan metode bagi dua (lakukan sampai 3 iterasi)?
   (Jawab: -0.5; -1.25; -0.875)
2. Tentukan akar persamaan $f(x) = x^2 - e^x$ dengan tebakan awal a = -2 dan b = 1 menggunakan metode posisi palsu (lakukan sampai 3 iterasi)?
   (Jawab: 0.0766802; -0.37487;-0.5763)
3. Bandingkan hasil dan jumlah iterasi perhitungan komputer metode bagi dua dengan metode posisi palsu soal no 1 dan 2 untuk Eps yang sama yaitu 0,00001?
   (Jawab: -0.703465 Bagi dua 18 iterasi; -0.703462 Posisi Palsu 13 iterasi)

2.2.3 Metode Newton Raphson

Penyelesaian akar persamaan tak linear dengan menggunakan Metode Newton Raphson berdasarkan pada f(x) dihampiri oleh perpotongan garis singgung melalui titik (x, f(x)) dengan sumbu x, seperti diperlihat pada gambar berikut ini:



Gambar 2.11 Penentuan akar dengan Metode Newton Raphson

Metode Newton Raphson termasuk dalam kelompok metode terbuka sehigga dalam proses iterasinya tidak memerlukan selang [a, b] yang mengandung akar. Proses mencari akar dimulai dari sembarang nilai tebakan awal (x_0), selanjutnya diperbaiki untuk menghitung hampiran akar yang baru. Jumlah iterasi dan waktu hitung sangat tergantung dari nilai tebakan awal.

Untuk menentukan nilai x berikutnya diperoleh dari perpotongan garis singgung melalui (x_0, $f(x_0)$) dengan sumbu x. Persamaan garis singgung melalui (x_0, $f(x_0)$) adalah:

$$y-f(x_0)= m (x - x_0)$$

dimana: $m = f'(x_0)$

$$y-f(x_0)= f'(x_0) (x - x_0)$$

Perpotongan dengan sumbu x pada titik y = 0 dan $x = x_1$

$$-f(x_0)= f'(x_0) (x_1 - x_0)$$

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

Dengan cara yang sama diperoleh :

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

Dan seterusnya sampai:

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

Proses ini dilanjutkan sampai $|x_{k+1} - x_k| < Eps$

Kendala kedivergenan :

- $f'(x_k) = 0$
- Nilai hampiran berulang

Dari langkah-langkah diatas dapat disusun algoritma Metoda Newton Raphson sebagai berikut:

Masukan : f(x), f'(x)

Tebakan awal x_0

Eps = 0.00001

Keluaran : akar

Langkah-langkah:

1. Hitung

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

2. Jika $|x_1 - x_0| < Eps$ maka akar = x_1, selesai
3. $x_0 = x_1$
4. Kembali ke langkah 1

Selanjutnya adalah menerjemahkan algoritma ke dalam program komputer dengan menggunakan pemrograman Visual C++ dapat ditulis seperti berikut:

Kode program NR:

Program C++

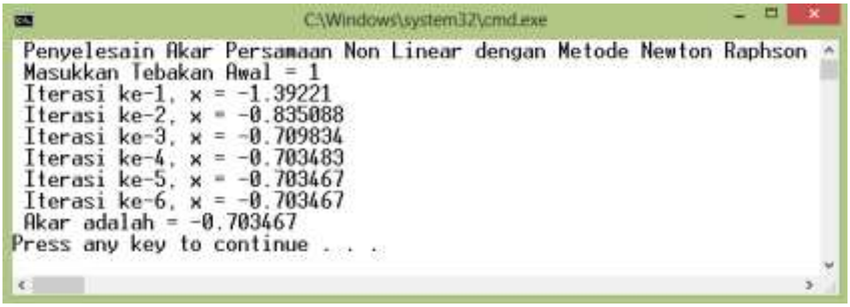
```
float x0, x1, eps=0.00001, Mak=100;
int Iter;
float f(float x);
float g(float x); // fungsi turunan
cout<<" Penyelesain Akar Persamaan Non Linear dengan Metode Newton
Raphson"<<endl;
cout<<" Masukkan Tebakan Awal = ";
cin>>x0;
Iter=0;
while (Iter<Mak){
       x1=x0-f(x0)/g(x0);
       if(abs(x1-x0)<eps) break;
       x0=x1;
       Iter=Iter+1;
```

```
        cout<<" Iterasi ke-"<<Iter<<", x = "<<x1<<endl;
}
cout<<" Iterasi ke-"<<Iter+1<<", x = "<<x1<<endl;
cout<<" Akar adalah = "<<x1<<endl;
```

Jika fungsi yang diselesaikan adalah f(x) = x^2 - e^x dengan fungsi turunan f'(x) = 2x-e^x maka tambahkan fungsi f(x) dan f'(x) tersebut pada akhir program seperti berikut:

```
float f(float x)
{
        float f=x*x-exp(x);;
        return f;
}

float g(float x)
{
        float g=2*x-exp(x);
        return g;
}
```

Selanjutnya compile dan jalankan program, diperoleh hasil seperti berikut:



Gambar 2.12 Hasil running program Metode Newton Raphson untuk f(x) = x^2 - e^x

2.2.4 Metode Secant

Metode Secant merupakan modifikasi dari metode Newton Raphson dalam hal $f'(x_k)$ sulit diperoleh, sehingga hampiran untuk $f'(x_k)$ dapat didekati dengan fungsi berikut:

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

Sehingga :

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

Dengan demikian algoritma metoda secant adalah sebagai berikut:

Masukan : f(x)

Tebakan awal x_0 dan x_1

Eps = 0.00001

Keluaran : akar

Langkah-langkah:

1. Hitung

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

2. Jika $|x_2 - x_1| < Eps$ maka akar = x_2, selesai
3. $x_0 = x_1$

   $x_1 = x_2$
4. Kembali ke langkah 1

Contoh soal:

2.3 Tentukan akar persamaan tak linear $f(x) = x^2 - e^x$ menggunakan Metode Secant dengan tebakan awal x_0 = -1, dan dan x_1=2, lakukan sampai dengan 4 iterasi ?

Jawab:

Iterasi 1:

$f(x_0) = f(-1) = (-1)^2 - e^{(-1)} = 0.632$

$f(x_1) = f(2) = (2)^2 - e^{(2)} = -3.389$

maka :

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

$$x_2 = 2 - (-3.389)\frac{2-(-1)}{-3.389-0.632}$$

x_2 = -0.582

Kemudian nilai x_0 dan x_1 diperbaharui untuk iterasi selanjutnya.

x_0 = 2, dan x_1 = -0.582

Iterasi 2:

$f(x_0) = f(2) = (2)^2 - e^{(2)} = -3.389$

$f(x_1) = f(-0.582) = (-0.582)^2 - e^{(-0.582)} = -0.22$

Hitung kembali nilai x_2 untuk iterasi ke 2:

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

$$x_2 = -0.582 - (-0.22)\frac{-0.582-(2)}{-3.389-(-0.22)}$$

x_2 = -0.783

Kemudian nilai x_0 dan x_1 diperbaharui lagi untuk iterasi selanjutnya.

x_0 = -0.582, dan x_1 = -0.783

Iterasi 3:

$f(x_0) = f(-0.582) = (-0.582)^2 - e^{(-0.582)} = -0.22$

$f(x_1) = f(-0.783) = (-0.783)^2 - e^{(-0.783)} = 0.156$

Hitung kembali nilai x_2 untuk iterasi ke 3:

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

$$x_2 = -0.783 - (0.156)\frac{-0.783-(-0.582)}{0.156-(-0.22)}$$

x_2 = -0.698

Kemudian nilai x_0 dan x_1 diperbaharui lagi untuk iterasi selanjutnya.

x_0 = -0.783, dan x_1 = -0.698

Iterasi 4:

$f(x_0) = f(-0.783) = (-0.783)^2 – e^{(-0.783)} = 0.156$

$f(x_1) = f(-0.698) = (-0.698)^2 – e^{(-0.689)} = -0.027$

Hitung kembali nilai x_2 untuk iterasi ke 4:

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

$$x_2 = -0.698 - (0.027)\frac{-0.698 - (-0.783)}{-0.027 - (0.156)}$$

x_2 = -0.701

Sampai iterasi ke-4 nilai akar adalah x_2 = -0.701

Berdasarkan algoritma metode secant dapat ditulis kode program Visual C++ seperti berikut:

Program C++

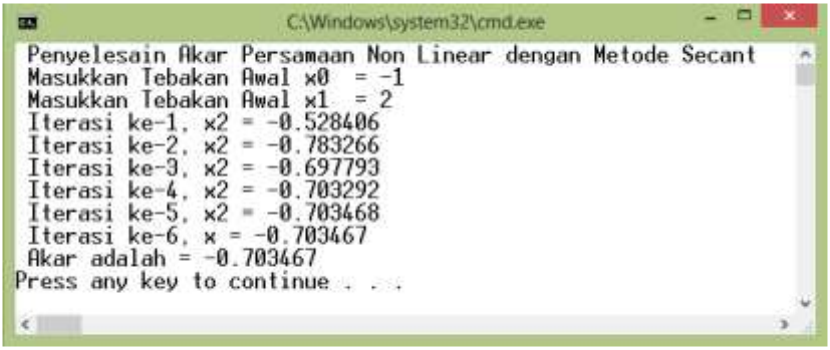
```
float x0,x1,x2,eps=0.00001, Mak=100;
int Iter;
float f(float x);
cout<<" Penyelesain Akar Persamaan Non Linear dengan Metode 
Secant"<<endl;
cout<<" Masukkan Tebakan Awal x0  = ";
cin>>x0;
cout<<" Masukkan Tebakan Awal x1  = ";
cin>>x1;
Iter=0;
while (Iter<Mak){
	x2=x1-f(x1)*(x1-x0)/(f(x1)-f(x0));
	if(abs(x2-x1)<eps) break;
	x0=x1;
```

```
        x1=x2;
        Iter=Iter+1;
        cout<<" Iterasi ke-"<<Iter<<", x2 = "<<x2<<endl;
}
cout<<" Iterasi ke-"<<Iter+1<<", x = "<<x2<<endl;
cout<<" Akar adalah = "<<x2<<endl;
```

Jika fungsi yang diselesaikan adalah f(x) = $x^2 - e^x = 0$, sama dengan prosedur penyelesaian metode sebelumnya maka tuliskan fungsi tersebut pada akhir program seperti berikut:

```
float f(float x)
{
        float f=x*x-exp(x);;
        return f;
}
```

Selanjutnya compile dan jalankan program, diperoleh hasil seperti berikut:



Gambar 2.10 Hasil running program metode secant untuk f(x) = $x^2 - e^x$

Soal Latihan:

1. Tentukan akar persamaan tak linear f(x) = $e^{2x} - 2x^2$ menggunakan Metode Newton Raphson dengan tebakan awal $x_0 = 0$, lakukan sampai dengan 3 iterasi ?

   Jawab: -0.5, -0.4517, -0.4506

2. Tentukan akar persamaan tak linear $f(x) = e^{2x} - 2x^2$ menggunakan Metode Secant dengan tebakan awal $x_0 = 0$, dan dan $x_1=2$, lakukan sampai dengan 3 iterasi ?

   Jawab: -0.0438614; -0.0846692; -0.496253

3. Bandingkan hasil dan jumlah iterasi perhitungan komputer Metode Newton Raphson dengan metode Secant soal no 1 dan 2 untuk Eps yang sama yaitu 0,00001?

   (Jawab: -0.450601 Newton Raphson 7 iterasi; -0.450601 Secant 7 iterasi)

BAB III
INTERPOLASI

Teori iterpolasi digunakan untuk mencari nilai tertentu diantara sejumlah data himpunan titik-titik yang diketahui. Untuk dapat memperkirakan nilai tersebut, pertama kali dibuat suatu fungsi atau persamaan yang melalui titik-titik yang diketahui, setelah persamaan garis atau kurva terbentuk, kemudian dihitung nilai fungsi yang berada di antara titik-titik yang diketahui tersebut. Himpunan titik-titik yang diketahui tersebut didekati dengan fungsi Polinom berikut:

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + - - - + a_n x^n$$

Nilai koefisien a_0, a_1, a_2, . . ., a_n dapat diperoleh dengan pola Lagrange dan pola Newton.

3.1 Interpolasi dengan Pola Lagrange

Untuk membentuk polinom ini, diambil beberapa titik diskrit dari suatu fungsi f. Titik-titik tersebut secara alami direpresentasikan dalam bentuk tabel atau susunan angka-angka. Selanjutnya titik-titik data ini dicocokkan untuk menentukan polinom $P_n(x)$ yang menghampiri fungsi aslinya.

Jika diketahui 2 titik : (x_0, $f(x_0)$) dan (x_1, $f(x_1)$), maka fungsi yang diperoleh adalah poilinom orde satu atau fungsi linear

$$P_1\,(x) = \frac{x - x_1}{x_0 - x_1}\, f_0 + \frac{x - x_0}{x_1 - x_0}\, f_1$$

Jika x = x_0, maka $P_1(x_0) = \dfrac{x_0 - x_1}{x_0 - x_1} f_0 + \dfrac{x_0 - x_0}{x_1 - x_0} f_1$

$$= 1.f_0 + \frac{0}{x_1 - x_0} f_1$$

$$= f_0$$

Jika x = x_1, maka $P_1(x_1) = \frac{x_1 - x_1}{x_0 - x_1} f_0 + \frac{x_1 - x_0}{x_1 - x_0} f_1$

$$= \frac{0}{x_0 - x_1} f_0 + 1 f_1$$

$$= f_1$$

Jika diketahui 3 titik : (x_0, $f(x_0)$), (x_1, $f(x_1)$) dan (x_2, $f(x_2)$), maka fungsi yang diperoleh adalah *kuadratik.*

atau

$$P_2(x) = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2} f_0 + \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2} f_1 + \frac{x - x_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1} \cdot f_2$$

$$P_2(x) = \sum_{i=0}^{2} li(x) \cdot fi$$

$$P_2(x) = \prod_{j=1,2} \frac{x - x_j}{x_0 - x_j} \cdot f_0 + \prod_{j=0,2} \frac{x - x_j}{x_1 - x_j} \cdot f_1 + \prod_{j=0,1} \frac{x - x_j}{x_2 - x_j} \cdot f_2$$

Sehingga dalam bentuk umum dapat ditulis:

$$P_2(x) = \sum_{i=0}^{2} \prod_{\substack{j=0 \\ j \neq i}}^{2} \frac{x - x_j}{x_i - x_j} \cdot f_i$$

Jika dikethui 4 titik (x_0, $f(x_0)$), (x_1, $f(x_1)$), (x_2, $f(x_2)$) dan (x_3, $f(x_3)$), maka fungsi yang diperoleh adalah *kubik*

$$P_3(x) = \frac{x-x_1}{x_0-x_1}\cdot\frac{x-x_2}{x_0-x_2}\cdot\frac{x-x_3}{x_0-x_3}\cdot f_0 + \frac{x-x_0}{x_1-x_0}\cdot\frac{x-x_2}{x_1-x_2}\cdot\frac{x-x_3}{x_1-x_3}\cdot f_1 +$$

$$\frac{x-x_0}{x_2-x_0}\cdot\frac{x-x_1}{x_2-x_1}\cdot\frac{x-x_3}{x_2-x_3}\cdot f_2 + \frac{x-x_0}{x_3-x_0}\cdot\frac{x-x_1}{x_3-x_1}\cdot\frac{x-x_2}{x_3-x_2}\cdot f_3$$

$$= l_0(X)\cdot f_0 + l_1(X)\cdot f_1 + l_2(X)\cdot f_2 + l_3(X)\cdot f_3$$

Atau

$$= \sum_{i=0}^{3}\prod_{\substack{j=0\\ j\neq i}}^{3}\frac{x-x_j}{x_i-x_j}\cdot f_i$$

Sehingga dalam bentuk umum dapat ditulis:

$$P_3(X) = \sum_{l=0}^{3} li(X).f_i$$

Pola umum Polinom Lagrange :

Untuk $n+1$ titik yang diketahui adalah:

$$P_n(X) = \sum_{i=0}^{n}\left\{\prod_{\substack{j=0\\ j\neq i}}^{n}\left(\frac{x-x_j}{x_i-x_j}\right)\right\}\cdot f_i$$

Contoh soal:

3.1 Diketahui himpunan titik berikut:

x	**f(x)**

1	-1
2	2
3	7

a. Tentukan hampiran fungsi f(x) untuk pasangan titik-titik tersebut?

b. Tentukan nilai fungsi untuk x = 2.5?

Jawab:

a. Karena diketahui 3 titik, maka fungsi didekati oleh polinom orde 2

$$P_2(x) = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2} f_0 + \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2} f_1 + \frac{x - x_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1} \cdot f_2$$

$$= \frac{x-2}{1-2} \cdot \frac{x-3}{1-3}(-1) + \frac{x-1}{2-1} \cdot \frac{x-3}{2-3} 2 + \frac{x-1}{3-1} \cdot \frac{x-2}{3-2} \cdot 7$$

$$= x^2 - 2$$

Jadi f(x) = $P_2(x) = x^2 - 2$, Selanjutnya titik lain diantara 1 sampai dengan 3 dapat ditentukan dengan memasukkan nilai x ke dalam fungsi tersebut.

b. Untuk menntukan nilai fungsi untuk x = 2.5 dapat langsung dengan mensubstutusikan nilai x kedalam $P_2(x)$ atau hampiran fungsi x.

$f(x) = P_2(x) = x^2 - 2$

x= 2.5 , maka:

$$\begin{aligned} f(2.5) &= P_2(2.5) \\ &= 2.5^2 - 2 \\ &= 4.25 \end{aligned}$$

Berdasarkan langkah-langkah di atas dapat disusun algoritma interpolasi pola Lagrange sebagai berikut:

PNZ = 0

u/i = 0,1,2, … n ;

faktor = 1

u/j = 0,1, … n

jika $i \neq j$,maka faktor=faktor* (z-x_j)/(x_i-x_j)

PNZ = PNZ + faktor * fi

Contoh soal:

3.2 Diketahui himpunan titik sebagai berikut:

X	**f**
1	2
5	7

3 →

Tentukan nilai fungsi untuk x = 3 menggunakan algoritma interpolasi pola Lagrange?

Jawab:

PNZ = 0 ;

i = 0;

faktor = 1 ;

j = 0 ;

j = 1

$$\text{faktor} = 1 \cdot \frac{3-5}{1-5} = 0{,}5$$

$\text{PNZ} = 0 + 0{,}5 \cdot 2 = 1$

i = 1

faktor = 1

j = 0

$$\text{faktor} = 1 \cdot \frac{3-1}{5-1} = 0{,}5$$

j = 1;

$\text{PNZ} = 1 + 0{,}5 \cdot 7 = 4{,}5$

Berdasarkan algoritma algoritma interpolasi pola Lagrange dapat ditulis kode program Visual C++ seperti berikut:

Program C++

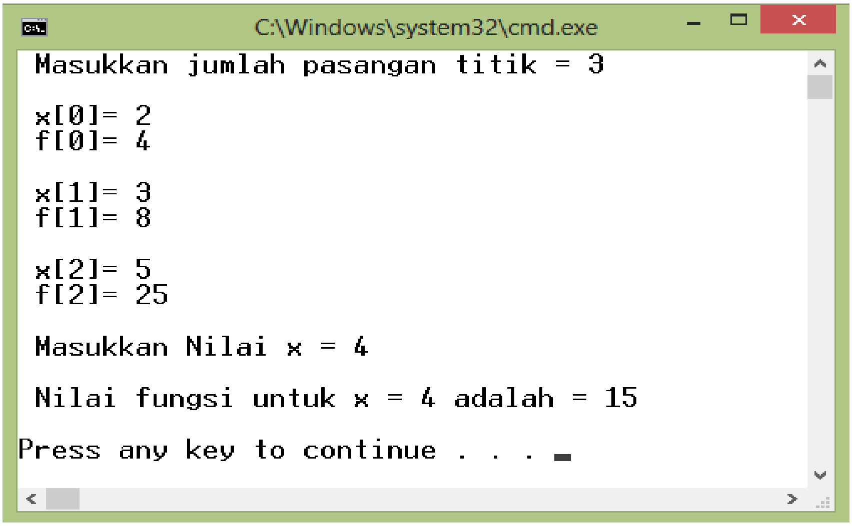
```
float PNZ,faktor,f[10],z;
int i,j,N,x[10];
N=3;
x[0]=2.0;
f[0]=4.0;
x[1]=3.0;
f[1]=8.0;
x[2]=5.0;
f[2]=25.0;
cout<<" Masukkan Nilai x = ";
cin>>z;
PNZ=0.0;
for(i=0;i<N;i++){
   faktor=1;
   for(j=0;j<N;j++){
      if(i!=j){
        faktor=faktor*((z-x[j])/(x[i]-x[j]));
      }
   }
  PNZ=PNZ+faktor*f[i];
}
cout<<endl<<" Nilai f(x) adalah :"<<PNZ<<endl<<endl;
```

atau jika titik dimasukkan melalui input keyboard:

```
float PNZ,faktor,f[10],z;
```

```
int i,j,N,x[10];

cout<<" Masukkan jumlah pasangan titik = ";
cin>>N;
for(i=0;i<N;i++){
	cout<<endl<<" x["<<i<<"]= ";
	cin>>x[i];
	cout<<" f["<<i<<"]= ";
	cin>>f[i];
}

cout<<endl<<" Masukkan Nilai x = ";
cin>>z;

PNZ=0.0;
for(i=0;i<N;i++){
	faktor=1;
	for(j=0;j<N;j++){
		if(i!=j){
			faktor=faktor*((z-x[j])/(x[i]-x[j]));
		}
	}
PNZ=PNZ+faktor*f[i];
}

cout<<endl<<" Nilai fungsi untuk x = "<<z<<" adalah = "<<PNZ<<endl;
```

Selanjutnya compile dan jalankan program, diperoleh hasil seperti berikut:



Gambar 3.1 Hasil running program interpolasi pola Lagrange

Soal Latihan:

1. Diketahui himpunak titik berikut ($f(x) = 2^x$):

x	F(x)
2	4
3	8
5	32

   Tentukan hampiran di titik x = 4 menggunakan interpolasi dengan Pola Lagrange ? (Jawab: 52/3)

2. Diketahui himpunak titik berikut : ($f(x) = 2\,x^2$)

x	F(x)
1	2

2	8
4	32

Tentukan hampiran di titik $x = 3$ menggunakan interpolasi dengan Pola Lagrange? (Jawab: 18)

3.2 Interpolasi dengan Pola Newton

Penyelesaian interpolasi dengan Pola Newton didekati dengan polinom berikut:

$P_1(x) = b_0 + b_1 (x - x_0)$

$P_2(x) = b_0 + b_1 (x - x_0) + b_2 (x - x_0)(x - x_1)$

$= P_1(x) + b_2 (x - x_0)(x - x_1)$

$P_3(x) = P_2(x) + b_3 (x - x_0)(x - x_1)(x - x_2)$

•

•

•

$P_n(x) = P_{n-1}(x) + b_n (x - x_0)(x - x_1). \dots (x - x_{n-1})$

$P_1(x)$ melalui (x_0, f_0) dan (x_1, f_1)

$P_1(x_0) = b_0 + b_1 (x_0 - x_0)$

$= b_0 = f_0$

$P_1(x_1) = b_0 + b_1 (x_1 - x_0)$

$= f_1$

$P_1(x_1) = f_0 + b_1 (x_1 - x_0) = f_1$ karena $f_0 = b_0$

Dari persamaan tersebut diperoleh:

$$b_1 = \frac{f_1 - f_0}{x_1 - x_0} = f[x_0, x_1]$$

Maka dapat ditulis

$P_1(x) \quad = f_0 + f\,[x_0, x_1]\,(x_1 - x_0)$

Dengan cara yang sama diperoleh :

$$b_2 = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = f[x_0, x_1, x_2]$$

$$b_3 = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} = f[x_0, x_1, x_2, x_3]$$

Dan seterusnya diperoleh:

$$b_n = \frac{f[x_1, x_2, \ldots x_n] - f[x_0, x_1, \ldots x_{n-1}]}{x_n - x_0} = f[x_0, x_1, \ldots, x_n]$$

Sehingga:

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

$$= f_0 + \sum_{i=1}^{n} f[x_0, x_1, \ldots, x_i] \prod_{j=0}^{(i-1)} (x - x_j)$$

Untuk mepermudah perhitungan buat tabel beda terbagi :

x	*f*	$f_1[\ ,\ ]$	$f_2[\ ,\ ,\ ]$		$f_{n-1}[\ ,\ , \ldots,\ ,\ ]$	$f_n[\ ,\ , \ldots,\ ,\ ]$
x_0	$f[x_0]$					
		$f[x_1, x_0]$				
x_1	$f[x_1]$		$f[x_2, x_1, x_0]$			
		$f[x_2, x_1]$				

x_2	$f[x_2]$		$f[x_3, x_2, x_1]$			
		$f[x_3, x_2]$				
x_3	$f[x_3]$		$f[x_4, x_3, x_2]$			
		$f[x_4, x_3]$		.		
.	.	.	.	.	$f[x_{n-1}, x_{n-2},..., x_1,x_0]$	
				.		$f[x_n, x_{n-1},..., x_1,x_0]$
				.	$f[x_n, x_{n-1},..., x_2,x_1]$	

Hasil intepolasi Newton adalah:

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

Contoh soal:

3.3 Diketahui himpunan titik berikut:

x	**f**
1	-1
2	2
3	7

a. Tentukan hampiran fungsi f(x) untuk pasangan titik-titik tersebut menggunakan pola Newton?

b. Tentukan nilai fungsi untuk x = 2.5?

Jawab:

Persamaan polinom Pola Newton untuk 3 titik:

$P_2(x) \quad = b_0 + b_1 (x - x_0) + b_2 (x - x_0) (x - x_1)$

$= f0 + f[x_1, x_0] (x\text{-}x_0) + f[x_2, x_1, x_0] (x\text{-}x_0) (x - x_1)$

$$= -1 + \frac{f_1 - f_0}{x_1 - x_0}(x\text{-}x_0) + \frac{f[x_2, x_1] - f[x_1 - x_0]}{x_2 - x_0}(x\text{-}x_0)(x - x_1)$$

$$= -1 + \frac{2-(-1)}{2-1}(x\ \text{-}1) + \frac{5-3}{3-1}(x\text{-}1)(x\text{-}2)$$

$= -1 + 3 (x\ \text{-}1) + 1(x\text{-}1)(x\text{-}2)$

$= -1 + 3x\ \text{-}3 + x^2\ \text{-}3x\ +2$

$= x^2 - 2$

Jadi hasilnya sama dengan interpolasi Lagrange $f(x) = P_n(x) = x^2\text{-}2$. Selanjutnya titik lain diantara 1 sampai dengan 3 dapat ditentukan dengan memasukkan nilai x ke dalam fungsi tersebut.

b. Untuk menentukan nilai fungsi untuk x = 2.5 dapat langsung dengan mensubstitusikan nilai x kedalam $P_2(x)$ atau hampiran fungsi x.

$f(x) = P_2(x) = x^2 - 2$

$x = 2.5 \rightarrow f(2.5) = P_2(2.5)$

$= 2.5^2 - 2$

$= 4.25$

Berdasarkan algoritma algoritma interpolasi pola Newton dapat ditulis kode program Visual C++ seperti berikut:

```
int n,i,j;
float x[10],f[10],z,PNZ,faktor;

   cout<<" Masukkan jumlah pasangan titik = ";
   cin>>n;
   for(i=0;i<n;i++){
          cout<<endl<<" x["<<i<<"]= ";
          cin>>x[i];
          cout<<" f["<<i<<"]= ";
```

```
            cin>>f[i];
        }
    cout<<endl<<" Masukkan Nilai x = ";
    cin>>z;

    // Interpolasi pola Newton
    PNZ=0;
    for(j=0;j<n-1;j++)
    {
        for(i=n-1;i>j;i--)
            f[i]=(f[i]-f[i-1])/(x[i]-x[i-j-1]);
    }
    for(i=n-1;i>=0;i--)
    {
        faktor=1;
        for(j=0;j<i;j++)
            faktor=faktor*(z-x[j]);

        faktor=faktor*f[j];
        PNZ=PNZ+faktor;
    }
    cout<<endl<<" Nilai fungsi untuk x = "<<z<<" adalah = "<<PNZ<<endl;
```

Soal Latihan:

1. Susun algoritma Interpolasi Pola Newton ?
2. Diketahui himpunak titik berikut ($f(x) = 2^x$):

x	F(x)
2	4
3	8
5	32

   Tentukan hampiran di titik x = 4 menggunakan interpolasi dengan Pola Newton? (Jawab: 52/3)

3. Diketahui himpunak titik berikut : ($f(x) = 2\ x^2$)

x	F(x)
1	2
2	8
4	32

Tentukan hampiran di titik x = 3 menggunakan interpolasi dengan Pola Newton? (Jawab: 18)

BAB IV

INTEGRAL NUMERIK

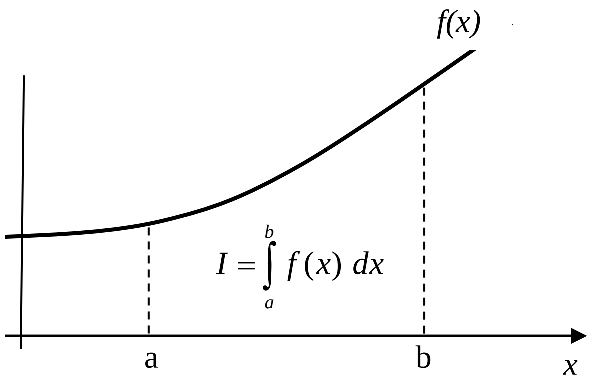
4.1 Pendahuluan

Integral banyak diterapan dalam penyelesaian bidang sains dan rekayasa. Integral digunakan untuk menghitung persamaan kecepatan, arus rms, dan penyelesaian sistem kendali.

Penyelesaian integral secara numerik adalah menentukan luas kurva terhadap sumbu x dengan batas bawah a dan batas atas b atau ditulis sebagai berikut:

$$I = \int_a^b f(x)\, dx$$

Persoalan integral batas dapat digambarkan sebagai berikut:



Gambar 4.1 Kurva integral batas

Penyelesaian persoalan Integral dengan metoda numerik diturunkan dari dua metode berikut:

1. Metoda Newton – Cotes
2. Metoda Pias

Metoda Newton - Cotes berdasarkan polinom interpolasi, dimana fungsi integrand f(x) dihampiri dengan polinom interpolasi $P_n(x)$. Selanjutnya, integrasi dilakukan terhadap $P_n(x)$ karena polinom lebih mudah diintegralkan daripada mengintegralkan f(x). Sedangkan Metoda Pias berdasarkan tafsiran geometri integral tentu yang dibagi atas sejumlah pias (strip) yang berbentuk segiempat. Luas daerah integrasi dihampiri dengan luas seluruh pias.

Berdasarkan metode Newton - Cotes tersebut fungsi f(x) dapat didekati dengan polinom berikut:

a. $f(x) \cong P_1(x)$

$$\int_a^b f(x)dx = \int_a^b P_1(x)dx$$

$$= \int_a^b \left(\frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b) \right) dx$$

$$= \frac{f(a)}{(a-b)} \int_a^b (x-b)dx + \frac{fb}{(b-a)} \int_a^b (x-a)dx$$

$$= \frac{b-a}{2} \left(f(a) + f(b) \right)$$

$$= \frac{h}{2} \left(f(a) + f(b) \right)$$

Kaidah ini disebut dengan kaidah Trapesium.

a. $f(x) \cong P_2(x)$

$$\int_a^b f(x)dx = \int_a^b P_2(x)dx$$

$$\int_a^b \left[\frac{\left(x-{(a+b)}/{2}\right)}{a-{(a+b)}/{2}} \frac{(x-b)}{(a-b)} f(a) + \frac{(x-a)(x-b)}{({(a+b)}/{2}-a)({(a+b)}/{2}-b)} f\left(\frac{a+b}{2}\right) + \frac{(x-a)(x-{(a+b)}/{2})}{(b-a)(b-{(a+b)}/{2})} f(b) \right] dx$$

$$= \frac{b-a}{3}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

$$= \frac{h}{3}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

Kaidah ini disebut dengan Kaidah Simpson

b. $f(x) \cong P_3(x)$

$$h = \frac{b-a}{3}$$

$$\int_a^b f(x)dx \cong \int_a^b P_3(x)dx$$

$$\cong \frac{3h}{8}\left(f(a) + 3f(a+h) + 3f(a+2h) + f(b)\right)$$

$$(Kaidah\, Simpson\;\; 3/8)$$

Kaidah ini disebut dengan Kaidah Simpson 3/8

c. $f(x) \cong P_4(x)$

$$h = \frac{b-a}{4}$$

$$\int_a^b f(x)dx \cong \int_a^b P_4(x)dx$$

$$\cong \frac{2h}{45}\left(7f0 + 32f1 + 12f2 + 32f3 + 7f4\right)$$

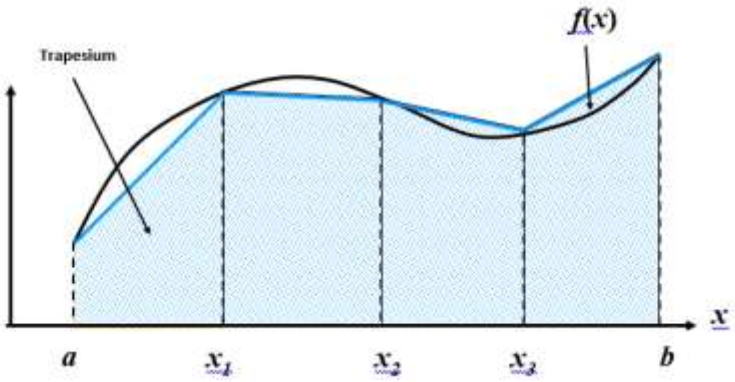
$$(Kaidah\, Bode)$$

Kaidah ini disebut dengan Kaidah Boole

Untuk selanjunya yang akan dibahas adalah integral numerik dengan menggunakan kaidah Trapesium dan Kaidah Simpson.

4.2 Aturan Trapesium Komposit

Untuk memperbaiki hasil perhitungan integral numerik, maka luas dibawah kurva dibagi atas trapesium-trapesium kecil sehingga metode ini disebut dengan trapesium komposit.



Gambar 4.2 Kaidah Trapesium Komposit

Bidang (a,b) dibagi atas m selang. Semakin banyak m hasil yang didapat semakin akurat.

Nilai h didapatkan dari:

$$h = \frac{b-a}{m}$$

dimana: a = nilai batas bawah

b = nilai batas atas

m = banyak pembagian dari (a,b)

Rumus Trapesium Komposit dapat ditulis sebagai berikut:

$$\int_a^b f(x)dx = \int_{x0}^{x1} P_1(x)dx + \int_{X1}^{X2} P_2(x)dx + \ldots$$

$$\int_a^b f(x)dx = \int_{X0}^{X1} f(x)dx + \int_{X1}^{X2} f(x)dx + \ldots + \int_{Xm-1}^{Xm} f(x)dx$$

$$= \frac{h}{2}\left(f(a) + f1\right) + \frac{h}{2}\left(f1 + f2\right) + \ldots + \frac{h}{2}\left(f_{m-1} + f_m\right)$$

$$\int_a^b f(x)dx = \frac{h}{2}\left[f(a) + 2.\sum_{i=1}^{m-1} fi + f(b) \right]$$

Algoritma:

Kaidah Trapesium Komposit

[a,b] : dibagi atas m selang

$$h = \frac{b-a}{m}$$

Algoritma :

Masukan : f(x),a,b,m

Langkah – langkah :

1. $h = \dfrac{b-a}{m}$
2. D = 0
3. Untuk i = 1,2,...,m-1

   x = a + i.h

   → D = D +f(x)
4. Trap = [f(a) + 2D + f(b)] * h/2

Keluaran : Trap

Contoh soal:

4.1 Selesaikan integral berikut menggunakan Kaidah Trapesium Komposit dengan m = 4:

$$\int_1^5 5 + 0.5x + 0.5x^2 dx$$

Jawab:

Fungsi f(x) = 5 + 0.5 x +0.5 x^2

h = $\frac{5-1}{4} = 1$

Menggunakan kaidah Trapesium Komposit dapat dihitung:

Trap = $\frac{h}{2}\left(f(1) + 2 * \sum_{i=1}^{3} f(i) + f(5) \right)$

= 0.5 (6 + 2(8 + 11 + 15) + 20)

= 0.5 (6 + 68 + 20)

= 0.5 (94)

Trap = 47

Berdasarkan kaidah Trapesium Komposit dapat disusun kode program menggunakan pemograman Visual C++ berikut:

Program C++ :

```
/* Menghitung nilai integral fungsi dengan menggunakan metode
trapesium Fungsi adalah f(x)= x / ( 1 + x * x ) Nilai batas atasnya
adalah 3.0 Nilai batas bawahnya adalah 1.0, Jumlah selangnya adalah
10 */

float fx,fa,fb,x,trap,h,D;
float a = 1.0 ,b = 3.0;
int i, m = 10;

fa = a / ( 1 + a * a ) ;
fb = b / ( 1 + b * b ) ;
h  = ( b - a ) / m;
D  = 0;
for ( i = 1 ; i <= m - 1 ; i++){
    x  = a + i * h;
    fx = x / ( 1 + x * x );
    D = D + fx;
```

```
}
trap = ( fa + 2 * D + fb ) * h / 2;
cout<<" Nilai integral adalah: "<<trap;
```

atau

```
        float x,trap,h,D;
        float a,b;
        int i, m;
        cout<<" Masukkan batas bawah, atas dan m : ";
        cin>>a>>b>>m;
        float f(float x);

        h  = ( b - a ) / m;
        D  = 0;
        for ( i = 1 ; i <= m - 1 ; i++){
        x  = a + i * h;
        D = D + f(x);
        }
        trap = ( f(a) + 2 * D + f(b) ) * h / 2;
        cout<<" Nilai integral adalah: "<<trap<<endl;
```

dimana fungsi f(x) dinyatakan sebagai fungsi berikut:

```
float f(float x)
{
        float f=(5+0.5*x+0.5*x*x);
        return f;
}
```

4.3 Aturan Simpson Komposit

Untuk memperbaiki hasil perhitungan integral numerik, maka luas dibawah kurva dibagi atas bagian-bagian kecil sehingga metode ini disebut dengan Simpson komposit.

Kaidah Simpson Komposit

$$\int_a^b f(x)dx = \int_{X_0}^{X_1} P_1(x)dx + \int_{X_1}^{X_2} P_2(x)dx + \ldots$$

atau

$$\int_a^b f(x)dx = \int_{X_0}^{X_2} f(x)dx + \int_{X_2}^{X_4} f(x)dx + \ldots + \int_{X_{m-2}}^{X_m} f(x)dx$$

$$\cong \frac{h}{3}\left(f(a) + 4f_1 + f_2\right) + \frac{h}{3}\left(f_2 + 4f_3 + f_4\right) + \ldots + \frac{h}{3}\left(f_{m-2} + f_m\right)$$

$$\cong \frac{h}{3}\left(f(a) + 4\sum_{k=1}^{m/2} f(2k-1) + 2\sum_{k=1}^{m/2-1} f(2k) + f(b)\right)$$

Berdasarkan langkah-langkah diatas dapat disusun Algoritma integral numerik kaidah Simpson komposit sebagai berikut:

Masukan : f(x),a,b,m (harus genap)

Langkah – langkah :

1. $h = {(b-a)}/{m}$
2. $D_1 = 0$
3. U/ k = 1,2,...,m/2
   - x = a + (2k – 1) h
   - $D_1 = D_1 + f(x)$
4. $D_2 = 0$
5. U/ k = 1,2,...,m/2 – 1;
   - x = a + 2k . h
   - $D_2 = D_2 + f(x)$
6. Simp = (f(a) + 4 *D_1 + 2*D_2 + f(b)) * h/3

Keluaran : Simp.

<u>Contoh Soal</u> :

4.2 Gunakan Kaidah Simpson Komposit dengan m = 4, pada integral :

$$\int_{1}^{5} 5 + 0.5x + 0.5x^2 dx$$

Jawab :

h = $\frac{5-1}{4} = 1$	x	f(x)
	1	6
	2	8
	3	11
	4	15
	5	20

Menggunakan kaidah Simpson Komposit dapat dihitung:

Simp $= \frac{1}{3}(6 + 4(8) + 11) + \frac{1}{3}(11 + 4(15) + 20)$

= 1/3 (6 + 4 (8 + 15) + 2 (11) + 20)

= 1/3 (6 + 4 (23) + 22 + 20)

= 1/3 (48 + 92)

= 1/3 (140)

= 46.67

Contoh Soal:

4.3. Hitunglah dengan menggunakan aturan trapesium komposit dan simpson komposit dengan m = 4?

$$\int_1^3 \frac{x}{1+x^2}dx$$

Jawab:

Pertama hitung lebar selang h berikut:

$$h = \frac{b-a}{m} = \frac{3-1}{4} = ½$$

Selanjutnya hitung f(x) = $\frac{x}{1+x^2}$ untuk setiap segmen sehingga diperoleh tabel berikut:

x	*f(x)*
1	0,5
1,5	0,46
2	0,4
2,5	0,34
3	0,3

Penyelesaian dengan aturan Trapesium Komposit, diperoleh:

$$I = \frac{0.5}{2}(0.5+0.46) + \frac{0.5}{2}(0.46+0.4) + \frac{0.5}{2}(0.4+0.34) + \frac{0.5}{2}(0.34+0.3)$$

$$= 0.803$$

Penyelesaian dengan aturan Simpson Komposit, diperoleh

$$I = \frac{0.5}{3}(0.5+4(0.46)+0.4) + \frac{0.5}{3}(0.4+4(0.34)+0.3)$$

$$= \frac{0.5}{3}(0.5+4(0.46+0.34)+2(0.4)+0.3)$$

$$= 0.804$$

Soal Latihan:

1. Hitung integral :

$$I = \int_{0}^{2} (x^2 + x - 6)\ dx$$

   a. Menggunakan aturan trapesium komposit dengan 2 interval ?

   b. Menggunakan aturan Simpson ?

   c. Bandingkan dengan cara biasa, bagaimana hasilnya ?

2. Hitung integral berikut menggunakan metode Simpson komposit :

   I= $\int_{1}^{7} (x-3)^2$; (m = 6)

3. Buat Kode program menggunakan bahasa pemograman C++ untuk menghitung integral numerik menggunakan aturan Simpson Komposit?

4. Hitung integral :

$$I = \int_{-2}^{2} (5 - 2x - x^3)\ dx$$

   a. Menggunakan aturan Trapesium komposit dengan m=4 ?

   b. Menggunakan aturan Simpson komposit dengan m=4?

   c. Bandingkan hasilnya dengan cara biasa ?

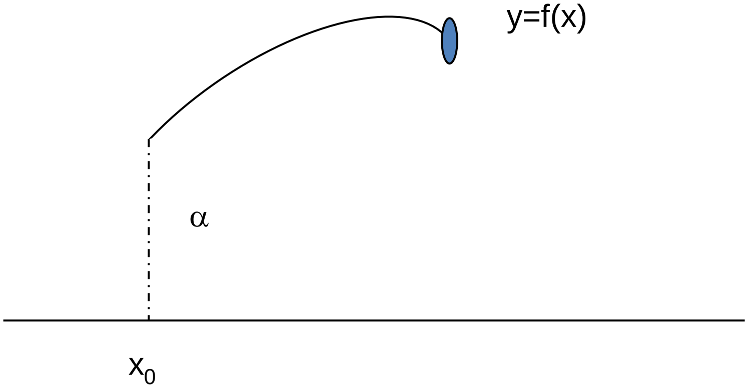
BAB V

PERSAMAAN DIFFERENSIAL NUMERIK

Persamaan differensial terdiri dari dua bentuk yaitu: Persamaan differensial biasa (PDB) dan persamaan differensial parsial (PDP). PDB persamaan diferensial yang hanya mempunyai satu peubah bebas. Peubah bebas biasanya disimbolkan dengan x.

Hasil penyelesaian secara numerik merupakan himpunan titik (x_k, y_k) yang menyatakan fungsi tersebut

$y' = f(x,y)$, $y(x_0) = \alpha$



Gambar 5.1 Fungsi turunan

5.1 Algoritma Euler

Dasar uraian taylor

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)(x-a)^2}{2!} + \cdots + \frac{f^{(n)}(a)(x-a)^n}{n!} + R_n$$

Jika diambil sampai dengan suku ke 2

$$y_{n+1} = y_n + h\, f(x_n, y_n)$$

Contoh soal:

5.1 Gunakan metode Euler untuk menyelesaikan persamaan diferensial berikut

$y' = y - x^2 + 1$, $y(0) = 0.5$

Hitung sampai dengan x = 2, dengan n = 10 .

Jawab:

h = (2-0)/10 = 0.2

y(0.2) = y(0) + h . f(0, 0.5)

= 0.5 + 0.2 $(0.5-(0)^2+1)$

=0.5 +0.2 *1.5

= 0.8

y(0.4) = y(0.2) + h . f(0.2, 0.8)

= 1.152

Dst

Sehingga jika ditabelkan menjadi:

x	y
0	0.5
0.2	0.8
0.4	1.152
0.6	1.5504
0.8	1.9885
1	2.4582
1.2	2.9498
1.4	3.4518
1.6	3.9501
1.8	4.4282
2	4.8658

Berdasarkan langkah-langkah di atas dapat disusun program komputer perhitunga turunan numerik menggunakan metode Euler sperti berikut:

Program C++:

```
for (k=0;k<=n-1;k++){
    x=a+k*h;
       cout<<x<<"  "<<y<<endl;
    y=y+(h*f(x,y));
}
```

Dengan f(x,y) adalah:

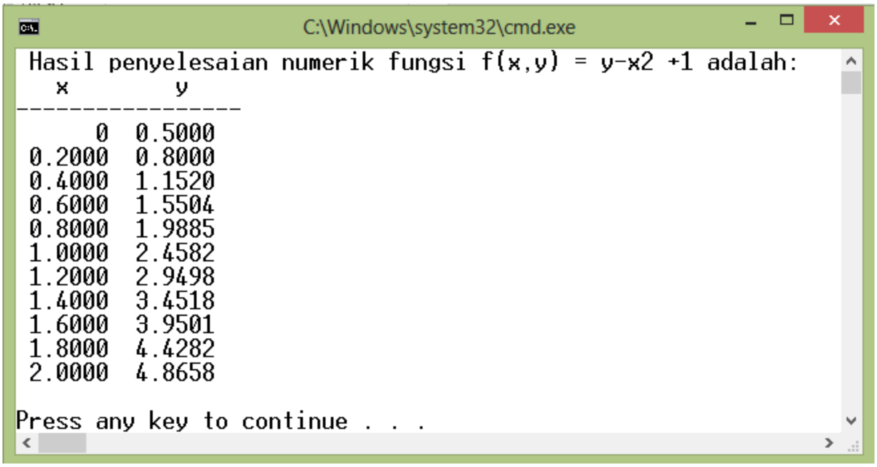
```
float f(float x, float y)
{
       float f=y-x*x+1;
       return f;
}
```

Dan tuliskan:

```
float f(float x, float y);
```

pada bagian deklarasi.

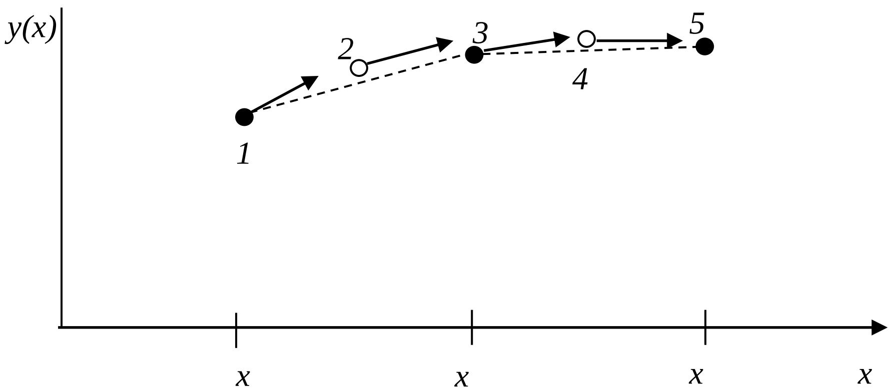
Jika contoh soal 5.1 dijalankan menggunakan kode program komputer diatas diperoleh hasil perhitungan seperti tampilan berikut:



Gambar 5.2 Hasil running Metode Euler

5.2 Metode Runge-Kutta

Pengembangan metode langkah tunggal berikutnya untuk memperbaiki hasil perhitungan metode Euler adalah Metode Rungge Kutta.



Gambar 5.2 Langkah-langkah Metode Rungge Kutta Orde 4

Hampiran nilai y dengan Metode Rungge Kutta adalah:

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + Oh^5$$

Dimana:

$$k_1 = h \ f(x_n, y_n)$$

$$k_2 = h \ f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$$

$$k_3 = h \ f(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})$$

$$k_4 = h \ f(x_n + h, y_n + k_3)$$

Contoh soal:

5.2 Gunakan metode Rungge kutta orde 4 untuk menyelesaikan persamaan diferensial berikut

y’= y-x^2 +1, y(0) = 0.5

Hitung sampai dengan x = 2, atau n = 10 dengan h =0.2.

Jawaban:

h = (2-0)/10

$$y_1 = y_0 + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}$$

$$k_1 = h \ f(x_n, y_n) = 0.3$$

$$k_2 = h \ f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) = 0.328$$

$$k_3 = h \ f(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) = 0.3308$$

$$k_4 = h \ f(x_n + h, y_n + k_3) = 0.358$$

Maka

$$y_1 = 0.5 + \frac{0.3}{6} + \frac{0.328}{3} + \frac{0.3308}{3} + \frac{0.358}{6} = 0.829$$

Dst

Sehingga jika ditabelkan menjadi:

x	y
0	0.5
0.2	0.8293
0.4	1.2141
0.6	1.6489
0.8	2.1272
1	2.6408
1.2	3.1799
1.4	3.7323
1.6	4.2834
1.8	4.8151
2	5.3054

Berdasarkan langkah-langkah di atas dapat disusun program komputer perhitungan turunan numerik menggunakan metode Rungge –Kutta Orde 4 seperti berikut:

Program C++:

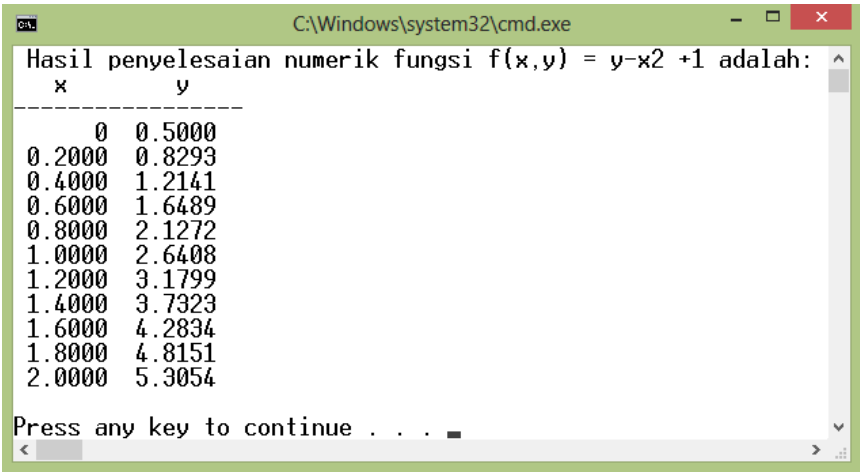
```
for (k=0;k<=n-1;k++){
    x=a+k*h;
    K1    = (h * f(x,y));
    K2    = (h * f((x + 0.5*h), (y + 0.5*K1)));
    K3    = (h * f((x + 0.5*h), (y + 0.5*K2)));
    K4    = (h * f((x + h), (y + K3)));
    y = (y+(K1 + 2 * K2 + 2 * K3 + K4)/6);
}
```

Dengan f(x,y) adalah:

```
float f(float x, float y)
```

```
{
        float f=y-x*x+1;
        return f;
}
```

Jika contoh soal 5.2 dijalankan menggunakan kode program Rungge Kutta Orde 4 diperoleh hasil perhitungan seperti tampilan berikut:



Gambar 5.2 Hasil running Rungge Kutta Orde 4

5.3 Persamaan Differensial Orde ke-2

Bentuk rumusannya:

$$y'' = \frac{d^2y}{dx^2} = f(x, y, y')$$

diketahui: $y(x_0) = y_0,\ y'(x_0) = y'_0 \dashrightarrow y(x) = ?$

Masalah persamaan differensial Orde 2 diubah menjadi Orde 1:

$u = y'$ maka $u_0 = y'_0$

$y' = u(x,y)$

$u' = f(x,y,u)$

Contoh:

5.3 Selesaikan persamaan differensial berikut:

$$\frac{d^2y}{dx^2} + 3\frac{dy}{dx} + 2xy = 1$$

Jika $x_0 = 0$, $y_0 = 0$ dan $z_0 = 0$, dengan h = 0,2, hitung nilai y yang lain sampai dengan x=2 menggunakan metode Euler?

Jawab:

Dari persamaan:

$$\frac{d^2y}{dx^2} + 3\frac{dy}{dx} + 2xy = 1$$

Dapat diperoleh dua fungsi PD orde 1:

Misalkan: z= dy/dx, maka:

f(x,y,z) = *z* dan

g(x,y,z) = 1-2xy-3z = turunan kedua

y_1 = $y_0 + h * f(x_0, y_0, z_0)$

= $0 + 0.2 * z_0$

= 0 + 0.2 (0) = 0

z_1 = $z_0 + h * g(x_0, y_0, z_0)$

= 0 + 0.2 * (1-2 (0) (0) – 3 (0))

= 0.2 * 1 = 0.2

y_2 = $y_1 + h * f(x_1, y_1, z_1)$

= $0 + 0.2 * z_1$

= 0 + 0.2 (0.2) = 0.04

z_2 = $z_1 + h * g(x_1, y_1, z_1)$

= 0.2+ 0.2 * (1-2 (0.2) (0) – 3 (0.2))

= 0.2+0.08 = 0.28

dan seterusnya sampai n = 10 diperoleh:

x = 2

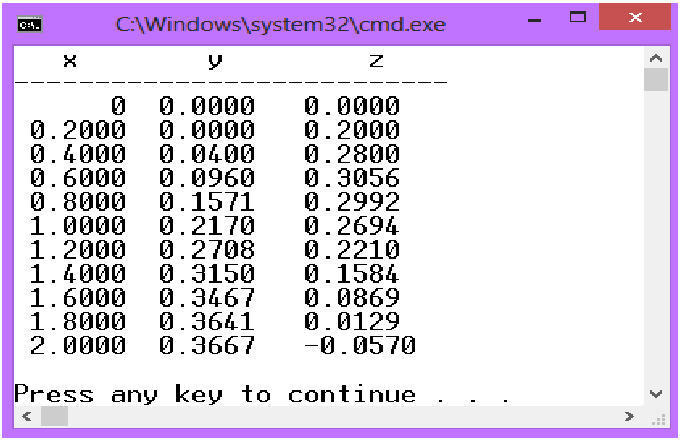
y_{10} = 0.36668

z_{10} = -0.057

Program komputer penyelesaian persamaan differensial orde 2 adalah:

```
float h, x,y,y1,z1,z,a;
int n,k;
float f(float x, float y, float z);
float g(float x, float y, float z);
n=10;
h=0.2;
y=0;
z=0;
a=0.0;

cout<<"   x         y      z "<<endl;
cout<<"------------------"<<endl;
for (k=0;k<=n;k++){
    x=a+k*h;
       cout << setprecision(4)<< setw(7);
       cout<<x<<"  "<<fixed <<setw(4) << right <<y<<"   "<<fixed
<<setw(4) << right <<z<<endl;
       y1=y+(h*f(x,y,z));
       z1=z+(h*g(x,y,z));
       y=y1;
       z=z1;
}

       cout<<endl;
       return 0;
}

Dimana fungsi f dan g adalah:

float f(float x, float y, float z)
{
       float f=z;
       return f;
}
float g(float x, float y, float z)
{
       float g=1-2*x*y-3*z;
       return g;
}
```

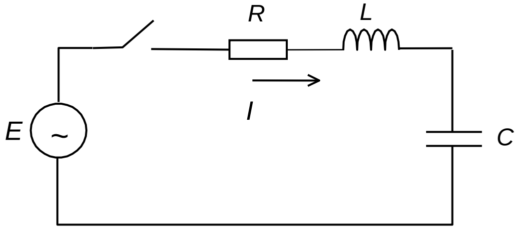
Jika contoh soal 5.3 dijalankan menggunakan program di atas diperoleh hasil perhitungan seperti tampilan berikut:



Turunan numerik banyak diaplikasikan pada penyelesaian persoaalan rekayasa, seperti rangkaian listrik AC untuk melihat repon tegangan saat saklar di tutup. Perhatikan contoh berikut:

Contoh soal:

5.4



Gambar 5.3 Rangkaian R, L dan C seri

Diketahui rangkaian R, L dan C dengan C = 10^{-5}, R= 10K dan L = 10^{-4} dan E = 1

Tentukan respons tegangan saat saklar ditutup ?

Jawab:

Persamaan matematis rangkaian diatas adalah:

$$C\frac{dV}{dt}+\frac{1}{R}V+L\int_{0}^{t}V(u)du=E$$

Jika kedua ruas diturunkan dan nilai C = 10^{-5}, R= 10K dan L = 10^{-4} dan E = 1 disubstitusikan diperoleh persamaan berikut:

$$\frac{d^2V}{dt^2}+10\frac{dV}{dt}+10V=1$$

Persamaan ini dapat dipisahkan menjadi dua fungsi PD orde 1:

Misalkan: z = dV/dt, maka:

f(t,V,z) = *z* dan

g(t,V,z) = 1-10z-10V

Jika saklar ditutup pada saat t = 0, V_0 = 0 dan V_0' = 1 dan h = 0.2, dengan metode Euler diperoleh:

Untuk t_1 = 0.2, maka:

V_1 = V_0 + h * f(t_0 ,V_0 ,z_0)

= 0 + 0.2 * 1

= 0.2

z_1 = z_0 + h * g(t_0 ,V_0 ,z_0)

= 1 + 0.2 * (1-10 (1) – 10 (0))

= 1-0.2 * 9

= 1-1.8 = -0.8

Untuk t_2 = 0.4, maka:

V_2 = V_1 + h * f(t_1,V_1,z_1)

$= 0.2 + 0.2 * z_1$

$= 0.2 + 0.2 (-0.8)$

$= 0.04$

$z_2 \quad = z_1 + h * g(t_1,V_1,z_1)$

$= -0.8 + 0.2 * (1-10 (-0.8) – 10 (0.2))$

$= -0.8 + 0.2 * (1+8-2)$

$= 0.6$

Untuk $t_3 = 0.6$, maka:

$V_3 \quad = V_2 + h * f(t_2,V_2,z_2)$

$= 0.04 + 0.2 * z_2$

$= 0.04 + 0.2 (0.6)$

$= 0.16$

$z_3 \quad = z_2 + h * g(t_2,V_2,z_2)$

$= 0.6 + 0.2 * (1-10 (0.6) – 10 (0.16))$

$= 0.8 + 0.2 * (1-6-1.6)$

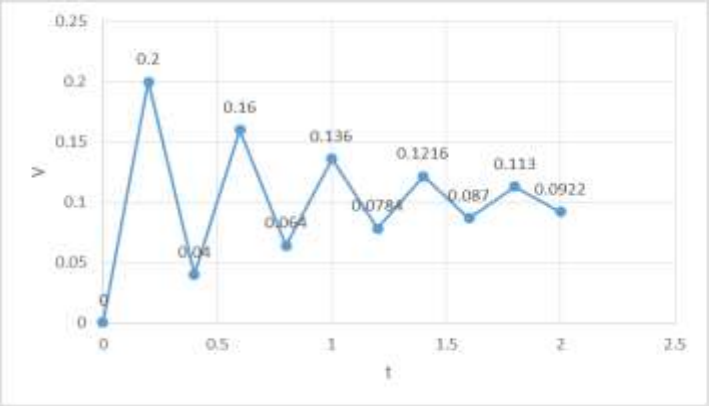
$= -0.52$

Dengan cara yangs sama, diperoleh hasil yang lain seperti tabel berikut:

n	**t**	**V**	**dV/dt**
0	0	0	1
1	0.2	0.2	-0.8
2	0.4	0.04	0.6
3	0.6	0.16	-0.52
4	0.8	0.064	0.36
5	1	0.136	-0.288
6	1.2	0.0784	0.216
7	1.4	0.1216	-0.1728
8	1.6	0.087	0.1296
9	1.8	0.113	-0.1037

10	2	0.0922	0.0778

Respons tegangan saat saklar ditutup adalah seperti gambar berikut:



Gambar 5.3 Respon tegangan dalam Rangkaian R, L dan C seri

5.4 Metode Banyak-Langkah

Sampai sejauh ini kita telah mengenal metode Euler, metode deret Taylor, dan metode Runge-Kutta. Semua metode tersebut dikelompokkan ke dalam metode satu langkah (one-step), sebab untuk menaksir nilai $y(x_{r+1})$ dibutuhkan satu buah taksiran nilai sebelumnya, $y(x_r)$.

Kelompok metode penyelesaian numerik PDB yang lain ialah metode banyak – langkah (multi step). Pada metode banyak langkah, perkiraan nilai $y(x_{r+1})$ membutuhkan beberapa taksiran nilai sebelumnya, $y(x_r)$, $y(x_{r-1})$, $y(x_{r-2})$, ... yang termasuk ke dalam metode banyak langkah adalah metode penduga pengoreksi (*predictor-corrector*).

Tujuan utama dari metode banyak langkah ini adalah menggunakan informasi dari beberapa titik sebelumnya y_k , y_{k-1} , y_{k-2} , … , untuk menghitung nilai y_{k+1} yang lebih baik.

Pada metode penduga pengoreksi, kita menaksir nilai $y(x_{k+1})$ dari $y(x_k)$, $y(x_{k-1}), y(x_{k-2}),\ldots$ dengan persamaan penduga, dan menggunakan persamaan pengoreksi untuk menghitung nilai y_{k+1} yang lebih baik.

Penduga : Menaksir y_{k+1} dari y_k , y_{k-1} , y_{k-2} , …

Pengoreksi : Memperbaiki nilai y_{k+1} dari penduga.

Beberapa Metode penduga pengoreksi antara lain adalah:

1. Metode Adams- Bashforth – Moulton

   Tinjau Persamaan Differensial Biasa orde satu

   $$y'(x) = f(x, y(x))$$

   integralkan kedua ruas persamaan dari x_k sampai x_{k+1} :

$$\int_{x_k}^{x_{k+1}} f(x, y(x))dx = \int_{x_k}^{x_{k+1}} y'(x)dx$$

$$= y(x)\Big|_{xk}^{xk+1}$$

$$= y(x_{k+1}) - y(x_k)$$

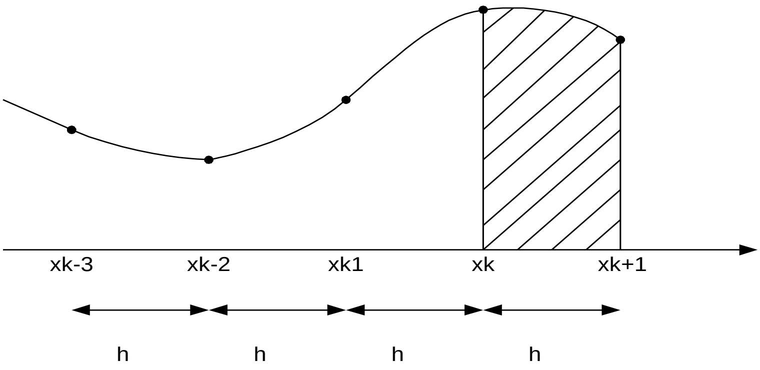
$$= y_{k+1} - y_k$$

$$y_{k+1} = y_k + \int_{x_k}^{x_{k+1}} f(x, y(x))dx$$

Persamaan di atas adalah merupakan dasar penurunan Persamaan Penduga dan Persamaan Pengoreksi

Untuk rumus penduga:

Persamaan penduga dihampiri dengan menghampiri fungsi menggunakan lagrage di 4 titik yang berjarak sama, yaitu: $\left(x_{kr-3}, f_{k-3}\right), \left(x_{kr-2}, f_{k-2}\right), \left(x_{kr-1}, f_{k-1}\right), \left(x_k, f_k\right)$



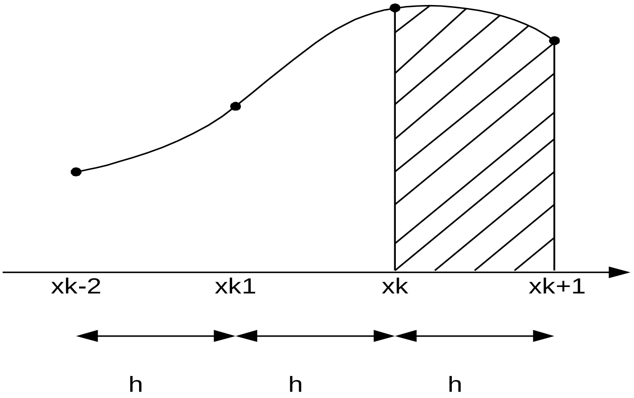
Gambar. Pembentukan persamaan penduga

Hasilnya :

$$p_{k+1} = y_k + \frac{h}{24}\left(-9 f_{k-3} + 37 f_{k-2} - 59 f_{k-1} + 55 f_k\right)$$

Persamaan pengoreksi dihampiri dengan menghampiri fungsi menggunakan lagrage di 4 titik yang berjarak sama,yaitu:

$\left(x_{kr-2}, f_{k-2}\right), \left(x_{kr-1}, f_{k-1}\right), \left(x_{kr}, f_k\right), \left(x_{k+1}, f_{k+1}\right)$



gambar. Pembentukan persamaan pengoreksi

Hasilnya :

$$y_{k+1} = \quad y_k + \frac{h}{24}\left(f_{k-2} - 5f_{k-1} + 19f_k + 9f_{k+1}\right)$$

$$f_{k+1} = f(x_{k+1}, p_{k+1})$$

Contoh soal:

5.5 Diketahui fungsi turunan berikut:

$$y' = \frac{x-y}{2} = f(x,y)$$

$$y_o = 1 \text{ pada } (0,1) \text{ dengan } h = \frac{1}{8}$$

Dengan menggunakan Metode Adams- Bashforth – Moulton, hitung y_4 = $y(0,5)$ dan y_5

Jawab:

Tiga nilai sebelumnya yaitu y_1, y_2 dan y_3 dihitung menggunakan metode RK4 sebagai berikut:

$$y_1 = \quad y_0 + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

$$k_1 = h\,f(x_0, y_0) = \frac{1}{8}\,f(0,1) = \frac{1}{8}(-0.5) = 0.625$$

$$k_2 = h\,f\left(x_0 + \tfrac{1}{2}h,\, y_0 + \tfrac{1}{2}k_1\right) = \left(\frac{-29}{512}\right)$$

$$k3 = h\,f\left(x_0 + \tfrac{1}{2}h,\, y_0 + \tfrac{1}{2}k_2\right) = -0.0568$$

$$k4 = h\,f\left(\frac{1}{8}, 0.9432\right) = -0.0511$$

diperoleh:

$y_1 = 0.94325$; selanjutnya hitung ; $f_1 = f\left(\frac{1}{8}, 0.94325\right)$

dengan cara yang sama diperoleh:

$y_2 = 0.8975$; selanjutnya hitung ; $f_2 = f\left(\frac{1}{4}, 0.8975\right)$

$y_3 = 0.8621$; selanjutnya hitung ; $f_3 = f\left(\frac{3}{8}, 0.8621\right)$

Kemudian hitung $p_4 = y_3 + \frac{h}{24}\left(-9f_0 + 37f_1 - 59f_2 + 55f_3\right)$

dengan $f_0 = f(x_0, y_0) = \frac{(0-1)}{2} = -\frac{1}{2}$

$$f_1 = f(x_1, y_1) = \frac{\tfrac{1}{8} - 0.94325}{2} = -0.409125$$

$$f_2 = f(x_2, y_2) = \frac{\tfrac{1}{4} - 0.8975}{2} = -0.5283125$$

$$f_3 = f(x_3, y_3) = \frac{\tfrac{3}{8} - 0.8621}{2} = -0.24355$$

diperoleh $p_4 = 0.03708$

$$f_4 = f(x_4, p_4) = \frac{1/2 - 0.03708}{2} =$$

$$y_4 = y_3 + \frac{h}{24}\left(f_1 - 5f_2 + 19f_3 + 9f_4\right)$$

$$= \underline{0.8575}$$

$$f_4 = f\left(x_4, y_4\right) = \frac{1/2 - 0.8575}{2} = -0.178$$

$$p_5 = 0.7593$$

$$f_5 = f(x_5, p_5) = \frac{5/8 - 0.7593}{2} =$$

$$y_5 = y_4 + \frac{h}{24}\left(f_2 - 5f_3 + 19f_4 + 9f_5\right)$$

$$= \underline{0.849}$$

Algoritma: Adams- Bashforth – Moulton

Masukan : $f(x,y)$

a

b

y_o

N {banyak selang}

langkah -langkah :

I. R-k orde 4 untuk menghitung y_1 , y_2, y_3

$$h = \frac{(b-a)}{N}$$

Untuk k = 0, 1, 2

$k_1 = \; h\, f\left(x_0, y_0\right)$

$k_2 = \; h\, f\left(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}k_1\right)$

$k3 \; = \; h\, f\left(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}k_2\right)$

$k_4 = h \; f\,(x_0 + h, y_0 + k_3)$

$$y_1 = \quad y_0 + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

II. Untuk k = 3, 4, 5….., N-1

x_k = a + k×h

$$p_{k+1} = \quad y_k + \frac{h}{24}\left(-9 f_{k-3} + 37 f_{k-2} - 59 f_{k-1} + 55 f_k\right)$$

$$f_{k+1} = f(\, x_{k+1}, p_{k+1})$$

$$y_{k+1} = \quad y_k + \frac{h}{24}\left(f_{k-2} - 5 f_{k-1} + 19 f_k + 9 f_{k+1}\right)$$

Keluaran : (x_k, y_k)

k = 0, 1, ……..,N

2. Metode Milne – Simpson

Metode Milne – Simpson didasrkan pada integrasi *f*(*x*, *y*(*x*)) pada selang (x_{k-3}, x_{k+1})

$$y\left(x_{k+1}\right) = \quad y\left(x_{k-3}\right) + \int_{x_{k-3}}^{x_{k+1}} f(x, y(x))dx$$

dengan polinom langrage diperoleh

Persamaan Penduga:

$$p_{k+1} = y_{k-3} + \frac{4h}{3}\left(2f_{k-2} + f_{k-1} + 2f_k\right)$$

Persamaan Pengoreksi :

$$y_{k+1} = y_{k-1} + \frac{h}{3}\left(f_{k-1} + 4f_k + f_{k+1}\right)$$

$$f_{k+1} = f(x_{k+1}, p_{k+1})$$

3. Metode Hamming

Persamaan Penduga:

$$p_{k+1} = y_{k-3} + \frac{4h}{3}\left(2f_{k-2} - f_{k-1} + 2f_k\right)$$

Persamaan Pengoreksi :

$$y_{k+1} = \frac{-y_{k-2} + 9y_k}{8} + \frac{3h}{8}\left(-f_{k-1} + 2f_k + f_{k+1}\right)$$

$$f_{k+1} = f(x_{k+1}, p_{k+1})$$

Untuk mendapatkan beberapa nilai awal yang lain, kita harus melakukan prosedur pendahuluan (starting procedure) dengan metode PDB yang bebas. Metode PDB yang sering dijadikan sebagai prosedur pendahuluan adalah:

- metode Euler
- metode Runge-Kutta
- metode deret Taylor

Contoh soal:

5.6 Suatu persamaan differensial berikut:

$$\frac{dy}{dx} - \frac{1}{2} y = 0$$

Diketahui $y(0) = 1$

Hitung $y(1)$, menggunakan metode Adam Bashfort Multon

Gunakan h = 0,25

Jawab:

$$\frac{dy}{dx} = f(x, y) = y' = 0.5y$$

Tiga titik pertama dihitung menggunakan metode Euler.

$$y_1 = y_0 + h.f(x_0, y_0)$$

$$y_1 = 1 + 0.25.0.5.1 = 1.125$$

$$y_2 = y_1 + h.f(x_1, y_1)$$

$$y_2 = 1.125 + 0.25.0.5.1.125 = 1.2656$$

$$y_3 = y_2 + h.f(x_2, y_2)$$

$$y_3 = 1.2656 + 0.25.0.5.1.2656 = 1.4238$$

Dan seterusnya, sehingga diperoleh seperti tabel berikut:

i	xi	yi	f(xi,yi)
k-3	0	1	0.5
k-2	0.25	1.125	0.5625
k-1	0.5	1.2656	0.6328
k	0.75	1.4238	0.7119
k+1	1	1.6018	0.8009

Selanjutnya menggunakan Metode Adams- Bashforth – Moulton diperoleh:

$$p_{k+1} = y_k + \frac{h}{24}\left(-9f_{k-3} + 37f_{k-2} - 59f_{k-1} + 55f_k\right)$$

$$= y_k + \frac{0.25}{24}\left(-9(0.5) + 37(0.5625) - 59(0.6328) + 55(0.7119)\right)$$

= 1.4238 + 0.18887

= 1.6127

Selanjutnya dihitung pengoreksi atau y(1) sebagai berikut:

$$f_{k+1} = f(x_{k+1}, p_{k+1})$$

= ½ y_{k+1} atau ½ P_{k+1}

= ½ 1.6127

= 0.8063

$$y_{k+1} = y_k + \frac{h}{24}\left(f_{k-2} - 5f_{k-1} + 19f_k + 9f_{k+1}\right)$$

$$= y_k + \frac{0.25}{24}\left(0.5625 - 5(0.6328) + 19(0.7119) + 9(0.8063)\right)$$

= 1.4238 + 0.1894

= 1.6132

Dengan demikian diperoleh y(1) = 1.6132

Soal Latihan

1. Gunakan metode Rungge kutta ordo 4 untuk menyelesaikan persamaan diferensial berikut

   $y' = y - x^2 + 1$, $y(0) = 0.5$

   Hitung sampai dengan x = 2, dengan n = 10 .
2. Jelaskan secara lengkap metode Adams Bashfort Multon untuk menentukan hampiran titik (x_k, y_k) yang menyatakan fungsi suatu turunan ?

BAB VI
MATRIK DAN OPERASI MATRIK

Matrik adalah susunan bilangan atau elemen matrik dalam bentuk persegi dan diapit oleh tanda kurung siku “[ ]”. Suatu matrik dinotasikan dengan huruf kapital tebal. Sebuah matrik mempunyai ukuran yang disebut dengan ordo sebagai indeks pada notasi matrik. Matrik berordo n x m berarti matrik tersebut terdiri dari n baris dan m kolom. Selain itu matrik memiliki elemen matrik yang memiliki indeks sebagai tanda posisi elemen didalam matrik. Elemen matrik berindek i,j berarti bilangan tersebut terletak pada barik ke-i dan kolom ke-j suatu matrik.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & a_{n-1n-1} & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

A adalah notasi matriks sedang a_{nm} adalah elemen matriks.

6.1 JENIS-JENIS MATRIK

Jenis-jenis matriks dapat dibagi berdasarkan ordo dan elemen / unsur dari matriks tersebut.

1. Matriks Bujursangkar

Matriks Bujursangkar adalah matriks yang memiliki ordo n x n atau banyaknya baris sama dengan banyaknya kolom yang terdapat dalam mtriks tersebut. Matriks ini disebut juga dengan matriks persegi berordo n.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & a_{n-1n-1} & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

Matriks Bujursangkar banyak digunakan pada penyelesaian sistem persamaan linier, dalam sistem ini jumlah persamaan (baris) dan jumlah bilangan tak diketahui (kolom) sehingga dapat diperoleh penyelesaian tunggal.

2. Matriks Diagonal

Matrik diagonal adalah matrik bujursangkar dengan semua diagonal bernilai 0.

$$\begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ a_{21} & 0 & \dots & a_{2n} \\ \dots & \dots & 0 & \dots \\ a_{n1} & a_{n2} & \dots & 0 \end{bmatrix}$$

$a_{ij} = 0$; untuk $j = i$

3. Matriks Satuan

Sama dengan matriks diagonal tetapi semua elemen diagonalnya bernilai 1

$$\begin{bmatrix} 1 & a_{12} & \dots & a_{1n} \\ a_{21} & 1 & \dots & a_{2n} \\ \dots & \dots & 1 & \dots \\ a_{n1} & a_{n2} & \dots & 1 \end{bmatrix}$$

$a_{ij} = 1$; untuk $j = i$

4. Matriks Segitiga

a. Segitiga bawah

$$\begin{bmatrix} a_{11} & 0 & .... & 0 \\ a_{21} & a_{22} & ... & 0 \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{bmatrix} \quad a_{ij} = 0 \;\; ; \;\; i < j$$

Semua elemen matrik yang berada diatas diagonal bernilai 0.

b. Segitiga atas

$$\begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ 0 & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ 0 & 0 & ... & a_{nn} \end{bmatrix} \quad a_{ij} = 0 \;\; ; \;\; \text{untuk } i>j$$

Semua elemen matrik yang berada dibawah diagonal bernilai 0.

5. Matrik Pita

Matriks pita, adalah matriks yang mempunyai elemen sama dengan 0, kecuali pada satu jalur yang berpusat pada diagonal utama, bentuknya sebagai berikut:

$$\begin{bmatrix} a_{11} & a_{12} & ... & a_{1n-1} & 0 \\ a_{21} & a_{22} & a_{23} & ... & a_{2n} \\ .a_{31}.. & .a_{32}.. & ... & ... & ... \\ ... & ... & ... & ... & ... \\ 0 & a_{n2} & ... & ... & a_{nn} \end{bmatrix}$$

Sifat matriks simetris, $\alpha_{ij} = \alpha_{ji}$; $A^T = A$.

6. Matrik Tridiagonal

Matrik Tridiagonal adalah matrik pita dengan lebar terkecil yaitu tiga seperti berikut:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \ldots & 0 \\ a_{21} & a_{22} & a_{23} & \ldots & \ldots \\ .0.. & a_{32} & \ldots & \ldots & .0 \\ \ldots & \ldots & \ldots & \ldots & a_{n-1,n} \\ ..0. & \ldots & .0.. & a_{n,n-1} & a_{nn} \end{bmatrix}$$

6.2 OPERASI MATRIK

A. Penjumlahan dan pengurangan matriks

Suatu matrik $A = [a_{ij}]$ dan matrik $B = [b_{ij}]$ dengan dimensi $m \times n$, maka untuk operasi penjumlahan atau pengurangan ($A \pm B$) dari kedua matriks tersebut, menghasilakan suatu matriks $C = [c_{ij}]$ dengan dimensi $m \times n$, dimana setiap elemen matriks C adalah jumlah atau selisih dari elemen-elemen yang berkaitan dari A dan B.

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1m} \\ b_{21} & b_{22} & \ldots & b_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ b_{n1} & b_{n2} & \ldots & b_{nm} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \ldots & c_{1m} \\ c_{21} & c_{22} & \ldots & c_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ c_{n1} & c_{n2} & \ldots & c_{nm} \end{bmatrix}$$

$c_{11} = a_{11} + b_{11}$

$c_{12} = a_{12} + b_{12}$

dan seterusnya.

Untuk operasi pengurangan matrik:

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1m} \\ b_{21} & b_{22} & \ldots & b_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ b_{n1} & b_{n2} & \ldots & b_{nm} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \ldots & c_{1m} \\ c_{21} & c_{22} & \ldots & c_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ c_{n1} & c_{n2} & \ldots & c_{nm} \end{bmatrix}$$

$c_{11} \quad = a_{11} - b_{11}$

$c_{12} \quad = a_{12} - b_{12}$

dan seterusnya.

B. Perkalian Matrik

A . x = b

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

Cara pengoperasian perkalian matriks dengan vektor :

$$b_1 = a_{11}\, x_1 + a_{12}\, x_2 + \dots + a_{1n}\, x_n$$

$$= \sum_{j=1}^{n} a_{1j}.x_j$$

$$b_i = \sum_{j=1}^{n} a_{ij}.x_j$$

untuk : i = 1, 2, 3, ..., n;

$b_i = 0$

j = 1, 2, 3, ..., n;

$b_i = b_i + a_{ij} \,.\, x_j$

b. Substitusi Maju (Forward Substitution)

Substitusi maju digunakan untuk menentukan vektor x jika matrik A adalah matrik segitiga bawah seperti berikut: