

APLIKASI PERHITUNGAN INTERPOLASI NEWTON DENGAN BORLAND DELPHI 5.0.

Ni Wayan Parwati Septiani

Program Studi Teknik Informatika – Fakultas Teknik, Matematika dan IPA
Universitas Indraprasta PGRI

***Abstract.** This Application program is a form of visualization calculation application of Newton Interpolation using Borland Delphi 5.0. This visual design from Borland Delphi 5.0. make easier for programmers to create an attractive user interface without typing any codes. Besides an attractive user interface, this applications perform Newton Interpolation calculations more quickly and efficiently.*

Key Words: interpolation, borland, application

PENDAHULUAN

Latar Belakang

Suatu harga atau nilai dengan interval tertentu dapat diperkirakan atau dilakukan penaksiran, yaitu dengan teknik interpolasi dan ekstrapolasi. Dimana keduanya merupakan dasar-dasar dari metode numerik. Metode numerik merupakan cara atau teknik dimana masalah-masalah matematika diformulasikan sedemikian rupa sehingga dapat diselesaikan oleh pengoperasian aritmatika. Perbedaan antara interpolasi dan ekstrapolasi adalah interpolasi memperkirakan harga atau nilai yang ada di dalam tabel yaitu di antara data awal dan data akhir, sedangkan ekstrapolasi memperkirakan harga atau nilai yang ada di luar tabel yaitu sebelum data awal atau sesudah data akhir. Interpolasi terbagi menjadi empat, yaitu interpolasi Linier, interpolasi Kuadrat, interpolasi Lagrange, dan interpolasi Newton. Dalam penulisan ini, penulis akan mengangkat masalah interpolasi, khususnya interpolasi Newton.

Untuk melakukan perhitungan dari interpolasi Newton ini biasanya masih dilakukan secara manual, dimana penggunaan seperti ini sangat tidak efektif dari segi waktu karena tata cara perhitungan interpolasi Newton

ini harus melakukan operasi pengurangan harga fungsi di titik n terhadap harga fungsi di titik $n-1$, perkalian, dan juga melakukan operasi penjumlahan, yang bisa saja nantinya terjadi ketidakakuratan informasi bagi para pengguna.

Hal-hal tersebut di ataslah yang melatarbelakangi penulis untuk membuat program komputerisasi perhitungan interpolasi Newton. Sehingga dengan demikian dapat memudahkan para pengguna dalam memperoleh output secara cepat dan juga tidak kalah pentingnya adalah dalam hal keakuratannya.

Tujuan Penulisan

- Mengimplementasikan Interpolasi Newton dengan menggunakan bahasa pemrograman Borland Delphi versi 5.0.
- Mempermudah pengguna dalam memperoleh output dengan cepat dan akurat.

TINJAUAN PUSTAKA

Pengertian Interpolasi

Sebuah fungsi seringkali disajikan dalam bentuk harga pada sebuah tabel. Sebagai contoh adalah tabel di bawah ini:

Tabel 1. Harga Fungsi

X	x ₀	x ₁	x ₂	x ₃	...	x _n
f(x)	f(x ₀)	f(x ₁)	f(x ₂)	f(x ₃)	...	F(x _n)

X	x ₀	x ₁	x ₂	x ₃	...	x _n
Y	y ₀	y ₁	y ₂	y ₃	...	y _n

x adalah letak titik dari suatu fungsi, sedangkan f(x) atau y adalah harga dari fungsi tersebut. Dari penjelasan di atas maka interpolasi merupakan suatu pendekatan numerik yang perlu dilakukan, bila diperlukan nilai suatu fungsi y = f(x) yang tidak diketahui perumusannya secara tepat, pada nilai argumen x tertentu, bila nilainya pada argumen lain disekitar argumen yang diinginkan diketahui. Lebih sederhananya, interpolasi adalah suatu teknik untuk mencari harga fungsi di suatu titik di antara dua titik yang harga fungsinya sudah diketahui atau cara penaksiran harga y pada saat x berada di luar tabel tetapi masih berada dalam interval-interval data yang ada yaitu y[x₀,x_n]. Untuk memperlihatkan hal ini, perhatikan contoh berikut. Misalkan dilakukan percobaan atau pengamatan, dan diperoleh sekumpulan data (x,y), seperti pada Tabel berikut hubungan y = f(x) tidak diketahui secara jelas (eksplisit).

Tabel 2. Data x dan y

X	y
1.0	1.0
1.1	1.21
1.2	1.44
1.3	1.69
1.4	1.96

Misalkan suatu waktu diperlukan nilai y = f(1.70), yang tidak tercantum pada Tabel diatas. Dalam keadaan demikian, perlu diperkirakan nilai y(1.70) dengan melakukan interpolasi pada data yang tersedia.

Polinomial Interpolasi

Polinomial biasa digunakan sebagai fungsi pendekatan pada kebanyakan

masalah-masalah analisa numerik karena strukturnya yang sederhana, sehingga menyebabkan polinomial dapat digunakan secara efektif.

Harga-harga fungsi di titik-titik yang diketahui sesuai dengan yang ada di tabel membentuk polinomial berderajat lebih kecil atau sama dengan n, polinomial ini disebut polinomial interpolasi. Titik x* yang ingin diketahui harga fungsinya dimasukkan ke dalam polinomial tersebut.

Berdasarkan derajat polinomial, interpolasi dibagi menjadi empat macam, yaitu:

1. Interpolasi Linier: untuk polinomial berderajat satu.
Persamaan umum polinomial derajat satu:

$$P(x) = a_0 + a_1x$$

2. Interpolasi Kuadrat : untuk polinomial berderajat dua.
Persamaan umum polinomial derajat dua:

$$P(x) = a_0 + a_1x + a_2x^2$$

3. Interpolasi Lagrange: untuk polinomial berderajat n-1 jika diberikan n titik, yaitu x₁, x₂, ..., x_n.
Persamaan umum polinomial derajat n-1:

$$P(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-2}x^{n-2} + a_{n-1}x^{n-1}$$

4. Interpolasi Newton : untuk polinomial berderajat n jika diberikan n+1 titik, yaitu x₀, x₁, x₂, ..., x_n
Persamaan umum polinomial Newton:

$$P(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n$$

a₀, a₁, a₂, ..., a_n disebut sebagai koefisien-koefisien polinomial interpolasi. Untuk menentukan koefisien-koefisien tersebut

dapat digunakan tabel selisih hingga, yaitu tabel selisih dari tiap selang hingga sampai selisih yang kita inginkan. Tabel selisih hingga ini terdiri atas 3 macam, yaitu: tabel selisih muka, dengan operator yang disebut “DELTA (Δ)”; tabel selisih belakang, dengan operator yang disebut “NABLA”; tabel selisih tengah, dengan operator yang disebut “DO”. Ketiganya memiliki hubungan DELTA = NABLA = DO. Ketiganya dicari dengan cara mengurangkan nilai data di titik n dengan nilai data di titik n-1.

Contoh:

Perhatikan daftar berikut yang menunjukkan jarak yang ditempuh (s, dalam meter) berturut-turut pada setiap selang waktu 10 detik.

t	s = f(t)
0	0
10	214
20	736
30	1446

Tabel selisih hingganya adalah:

Tabel 3. Selisih Fungsi Jarak

t	f(t)	$\Delta 1$	$\Delta 2$	$\Delta 3$
0	0	214		
10	214	522	308	
20	736	710	188	-120
30	1446			

Berdasarkan Tabel di atas, didefinisikan:

$$\Delta y_0 = y_1 - y_0$$

$$\begin{aligned} \Delta^2 y_0 &= \Delta (\Delta y_0) \\ &= \Delta (y_1 - y_0) \\ &= \Delta y_1 - \Delta y_0 \\ &= (y_2 - y_1) - (y_1 - y_0) \\ &= y_2 - 2y_1 + y_0 \\ &= y_2 - (2 / 1!) y_1 + ((2 * 1) / 2!) y_0 \end{aligned}$$

$$\begin{aligned} \Delta^3 y_0 &= \Delta (\Delta^2 y_0) \\ &= \Delta (y_2 - 2y_1 + y_0) \end{aligned}$$

Rumus Perhitungan Interpolasi Newton

Diberikan himpunan (n+1) buah nilai x dan y, yaitu $(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ dimisalkan dengan $(x_i, y_i), i = 0, 1, \dots, n$ dan interval antara x_i adalah sama. Dari nilai tersebut akan dicari $y_n(x)$, yakni suatu polinomial berderajat n, sedemikian sehingga y dan $y_n(x)$ memenuhi daftar titik tersebut. Dalam bentuk tabel disajikan dengan:

Tabel 4. Selisih Hingga Berderajat n

x_i	y_i	Δy_i	$\Delta^2 y_i$...	$\Delta^n y_i$
x_0	y_0				
		$y_1 - y_0$			
x_1	y_1		$(y_2 - y_1) - (y_1 - y_0)$		
		$y_2 - y_1$			
x_2	y_2	:	:		$(y_n - \dots) - (\dots - y_0)$
:	:	:	$(y_n - y_{n-1}) - (y_{n-1} - y_{n-2})$		
x_{n-1}	y_{n-1}	$y_n - y_{n-1}$			
x_n	y_n				

Keterangan:

x_i adalah letak titik ke-i dari suatu fungsi, dengan $i = 0, 1, \dots, n$.

y_i adalah harga atau nilai dari titik ke-i.

Banyak data adalah n dengan n+1 titik

$$\begin{aligned}
 &= \Delta y_2 - 2 \Delta y_1 + \Delta y_0 \\
 &= (y_3 - y_2) - 2(y_2 - y_1) + (y_1 - y_0) \\
 &= y_3 - 3y_2 + 3y_1 - y_0 \\
 &= y_3 - (3 / 1!) y_2 + ((3 * 2) / 2!) y_1 - \\
 &\quad ((3 * 2 * 1) / 3!) y_0
 \end{aligned}$$

$$\begin{aligned}
 \Delta^4 y_0 &= \Delta (\Delta^3 y_0) \\
 &= \Delta (y_3 - 3y_2 + 3y_1 - y_0) \\
 &= \Delta y_3 - 3 \Delta y_2 + 3 \Delta y_1 - \Delta y_0 \\
 &= (y_4 - y_3) - 3(y_3 - y_2) + 3(y_2 - y_1) - (y_1 - y_0) \\
 &= y_4 - 4y_3 + 6y_2 - 4y_1 + y_0 \\
 &= y_4 - (4 / 1!) y_3 + ((4 * 3) / 2!) y_2 - \\
 &\quad ((4 * 3 * 2) / 3!) y_1 + ((4 * 3 * 2 * 1) / 4!) y_0
 \end{aligned}$$

Maka:

$$\begin{aligned}
 \Delta^n y_0 &= y_n - (n / 1!) y_{n-1} + ((n(n-1)) / 2!) y_{n-2} - ((n(n-1)(n-2)) / 3!) y_{n-3} + \\
 &\quad ((n(n-1)(n-2)(n-3)) / 4!) y_{n-4} + \dots (+ \text{atau } -) \\
 &\quad ((n(n-1)(n-2)\dots(n-(n-1)(1))) / n!) y_{n-n}
 \end{aligned}$$

Keterangan:

Δ : menyatakan selisih antara $n+1$ dengan n .

$$n! : n \times (n-1) \times (n-2) \times \dots \times (n-(n-2)) \times 1$$

Operator suku genap adalah penjumlahan, sedangkan operator suku ganjil adalah pengurangan, dengan y_n sebagai suku genap.

Polinomial interpolasi secara umum adalah:

$$\begin{aligned}
 P_n(x) &= a_0 + a_1x + a_2x^2 + \dots + a_nx^n \\
 \text{Dengan } i &= 0, 1, 2, \dots, n \quad \text{dan} \quad h = \text{interval data} \\
 0! &= 1
 \end{aligned}$$

Maka:

$$a_i = (\Delta^i y_0) / (i! h^i)$$

Sehingga Polinomial Interpolasi Newton adalah:

$$\begin{aligned}
 P_n(x) &= y_0 + (\Delta y_0 / h)(x-x_0) + (\Delta^2 y_0 / 2! h^2)(x-x_0)(x-x_1) + \\
 &\quad (\Delta^3 y_0 / 3! h^3)(x-x_0)(x-x_1)(x-x_2) + \dots + (\Delta^n y_0 / n! h^n) \\
 &\quad (x-x_0)(x-x_1)\dots(x-x_{n-1})
 \end{aligned}$$

$$= \sum_{i=0}^n (\Delta^i y_0 / i! h^i) \prod_{j=0}^{i-1} (x-x_j) \quad ; \quad \Delta^0 y_0 = y_0$$

Keterangan: y_0 adalah harga awal dari fungsi.

Sekilas Tentang Borland Delphi 5.0

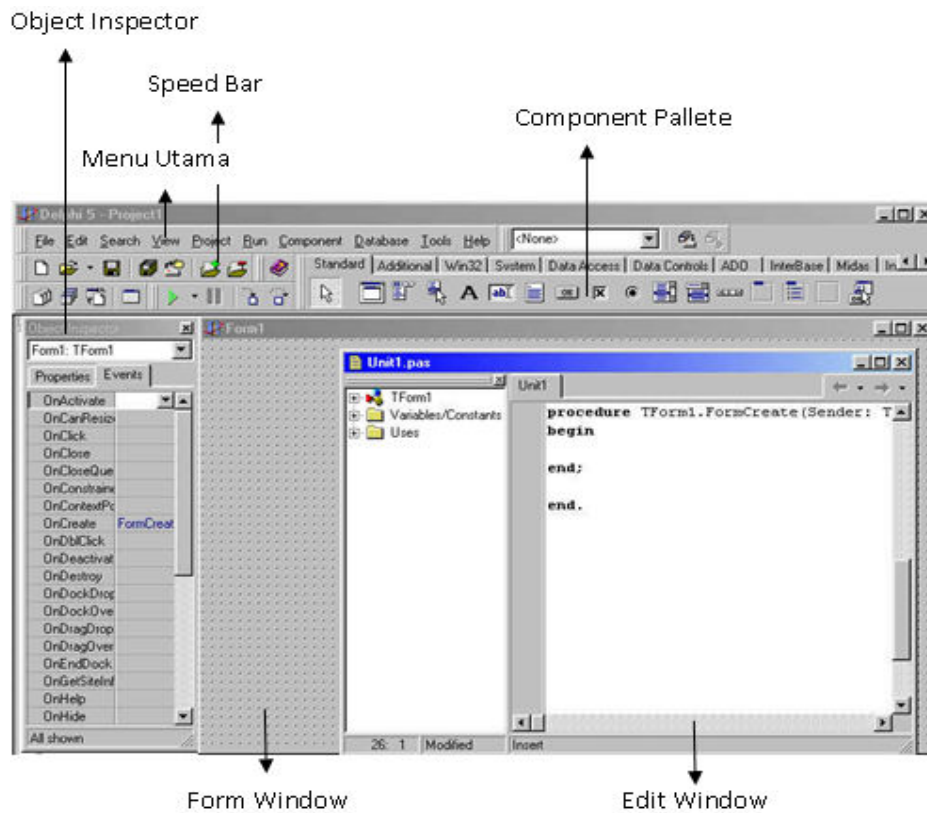
Pembuatan program ini menggunakan bahasa pemrograman Delphi, yang ditujukan untuk pemrograman berbasis pada sistem operasi Windows. Program ini sendiri menggunakan Borland Delphi versi 5.0 yang beroperasi pada Windows 9.x atau versi yang lebih tinggi.

Delphi sendiri memiliki bahasa dasar yaitu bahasa pemrograman Pascal dengan pengembangan Object Oriented Programming (OOP). Dengan digunakannya bahasa Pascal dalam Delphi maka para pengguna bahasa Pascal dapat dengan mudah untuk beralih ke bahasa pemrograman Delphi.

Integrated Development Environment (IDE)

IDE adalah sebuah lingkungan dimana semua tools yang diperlukan untuk disain, menjalankan, dan mengetes sebuah aplikasi disajikan dan terhubung dengan baik sehingga memudahkan pengembangan program. Melalui IDE inilah pemrogram secara visual

merancang tampilan untuk pemakai dan menuliskan kode. Pada Delphi, IDE terdiri dari Menu Utama, Speedbar, jendela Form, Object Inspector, Component Palette, dan jendela Edit Window (Object Window). Komponen-komponen IDE tersebut akan dijelaskan selanjutnya.



Gambar 1. Tampilan IDE Borland Delphi 5.0

Main Menu (Menu Utama)

Main Menu menampilkan menu yang berisikan perintah-perintah pada Borland Delphi 5.0.

Speed Bar (Palang Cepat)

Speed Bar disediakan oleh Borland Delphi 5.0 untuk mengakses berbagai fungsi yang ada dalam menu secara lebih cepat dan lebih mudah. Pada aplikasi Windows yang lain (biasanya

keluaran Microsoft), fasilitas ini lebih dikenal dengan istilah Toolbar.

Form (Lembar Kerja)

Form adalah lembar disain user interface dari aplikasi yang dibuat. Form ini menjadi pondasi tempat diletakkannya komponen visual yang dimiliki oleh Borland Delphi, sesuai dengan keinginan programmer. Bentuk dan penggunaan standard Form ini sesuai

dengan standard window (jendela) pada Windows.

Object Inspector (Inspeksi Obyek)

Object Inspector adalah kotak yang berisikan property dan event sebuah object terpilih. Object Inspector sering diakses saat bekerja dengan form maupun komponen yang terdapat di dalam form. Object Inspector ini terbagi dua, yaitu bagian properti (properties) dan bagian kejadian (events). Properti adalah sejumlah karakteristik atau sifat seperti tinggi, lebar dan judul yang dimiliki oleh form dan semua komponen. Pada bagian properties dapat digunakan untuk melihat dan memodifikasi properti sebuah object sedangkan bagian events merupakan tempat menuliskan procedure apakah yang harus dijalankan bila sebuah object menerima respon.

Component Palette (Palet Komponen)

Component Palette berisi kumpulan komponen yang menjadi ciri Borland

Delphi sebagai sebuah bahasa pemrograman visual. Semua komponen yang tersedia pada Component Palette disebut dengan Visual Component Library (VCL), dimana VCL terdiri atas dua jenis komponen, yaitu Visual Component (VC) dan Non Visual Component (NVC). Program Interpolasi Newton ini hanya menggunakan Visual Component yang akan dibahas selanjutnya.

Edit Window (Jendela Program)

Edit Window merupakan tempat menuliskan rutin program. Ini termasuk perlengkapan paling penting pada Borland Delphi, karena rutin yang dibuat di dalamnya akan menentukan mekanisme kerja program. Editor Delphi sangat canggih, dengan fasilitas-fasilitas highlight untuk memudahkan menemukan kesalahan kerangka program sehingga tidak perlu menuliskan seluruh program.

Perhitungan Interpolasi Newton secara Manual

Tabel berikut menunjukkan nilai dari tangens, tg x, dengan interval 0,05 untuk 0,10 <= x <= 0,30.

Tabel 5. Nilai Tangens x

X	0,10	0,15	0,20	0,25	0,30
y = tg x	0,1003	0,1511	0,2027	0,2553	0,3093

Tentukan tg 0,12 !

Jawab:

Tabel Selisih dari data di atas adalah seperti berikut:

Tabel 6. Selisih Tangens x

X	y	Delta1	Delta2	Delta3	Delta4
0,10	0,1003				
		0,0508			
0,15	0,1511		0,0008		
		0,0516		0,0002	
0,20	0,2027		0,001		0,0002
		0,0526		0,0004	
0,25	0,2553		0,0014		
		0,0540			
0,30	0,3093				

Polinomial Interpolasi Newton:

$$y(x) = y_0 + (\Delta y_0 / 1! h)(x-x_0) + (\Delta^2 y_0 / 2! h^2)(x-x_0)(x-x_1) + (\Delta^3 y_0 / 3! h^3)(x-x_0)(x-x_1)(x-x_2) + (\Delta^4 y_0 / 4! h^4)(x-x_0)(x-x_1)(x-x_2)(x-x_3)$$

$$\begin{aligned}
 y(0,12) &= 0,1003 + (0,0508 / 1! 0,05^1)(0,12-0,10) + (0,0008 / 2! 0,05^2) \\
 &\quad (0,12-0,10)(0,12-0,15) + (0,0002 / 3! 0,05^3)(0,12-0,10)(0,12-0,15) \\
 &\quad (0,12-0,20) + (0,0002 / 4! 0,05^4)(0,12-0,10)(0,12-0,15)(0,12-0,20) \\
 &\quad (0,12-0,25) \\
 &= 0,1003 + 0,02032 + (-0,000096) + 0,0000128 - 0,00000832 \\
 &= 0,12052848
 \end{aligned}$$

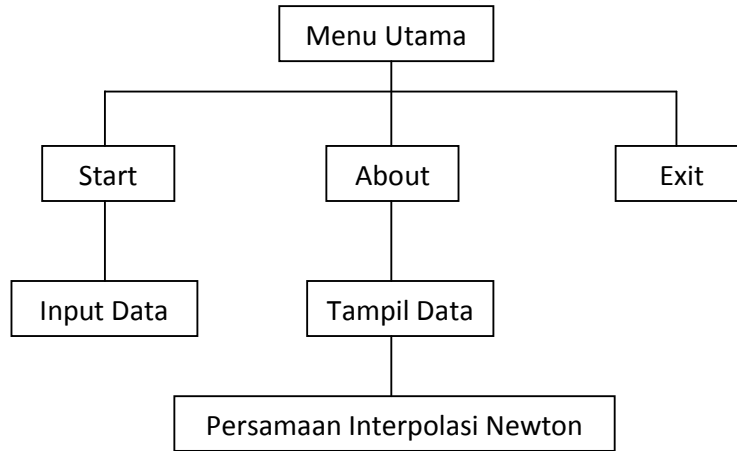
METODE

Dalam penulisan ilmiah ini, penulis melakukan studi pustaka, dimana studi pustaka ini merupakan pengumpulan beberapa buku yang berkaitan dengan interpolasi Newton dan program yang digunakan yaitu Delphi versi 5.0.

Sebelum membuat suatu program, sebaiknya terlebih dahulu dibuat suatu struktur program yang dapat menggambarkan tentang disain output yang akan ditampilkan secara konseptual dan membantu pemahaman tentang aplikasi atau program yang dibuat. Adapun struktur pada program interpolasi Newton yang akan dibuat adalah sebagai berikut:

HASIL DAN PEMBAHASAN

Rancangan Program



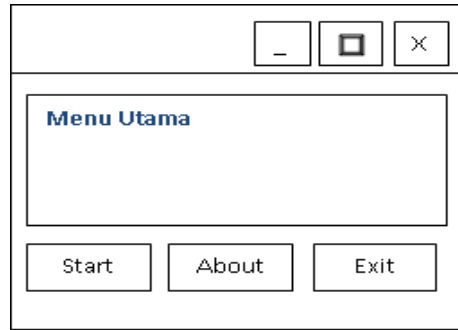
Gambar 2. Struktur Program.

Rancangan Input-Output

Bila struktur program telah dibuat atau dirancang, maka langkah selanjutnya adalah membuat rancangan untuk input dan outputnya yang berfungsi untuk memberikan gambaran secara konseptual pada gambaran tentang tampilan-tampilan program yang akan dihasilkan nanti. Adapun rancangan input/output yang akan dibahas ada 4 bagian/tampilan, yaitu sebagai berikut:

Rancangan Form Menu Utama

Menu Utama dirancang untuk menampilkan bagian-bagian program yang dapat dipilih sesuai dengan yang diinginkan. Pada desain menu utama terdapat *label judul* yang diberi nama **Program Perhitungan Interpolasi Newton Pada Selang Data Sama** dan terdapat juga *menu-menu pilihan* yang dipilih dengan cara mengklik tombolnya atau dengan cara menekan tombol Alt bersamaan dengan menekan huruf pertama pada tombol yang akan dipilih.



Gambar 3. Rancangan Form Menu Utama.



Gambar 4. Form Menu Utama

Gambar diatas terdiri atas judul form yang diberi nama **Menu Utama**, tiga macam kontrol menu yaitu **minimize** (untuk mengecilkan form), **maximize** (untuk membesarkan kolom), dan **close** (untuk menutup form), tampilan judul aplikasi yang berjudul **PROGRAM PERHITUNGAN INTERPOLASI NEWTON PADA SELANG DATA SAMA**, dan tiga macam command button, yaitu command **Start**, command **About**, dan command **Exit**.

Rancangan Form Input Data

Pada rancangan **Input Data** yang akan dibuat, terdiri atas beberapa **label** dan beberapa **edit** yang berfungsi untuk memasukkan data yang akan dihitung Interpolasi Newtonnya. Edit-edit itu antara lain: **edit** yang akan menginput

banyaknya data yang diperlukan, **edit** yang menginput data awal dari data yang diperlukan, **edit** yang meminta inputan interval data-data tersebut, dan **edit** yang menginput data yang akan dicari nilai fungsinya dari data tersebut. Ditambah dengan **label** yang menyatakan judul dari tampilan pada form tersebut. Selain itu dilengkapi dengan **tombol Ok**, yang ketika di klik akan mengaktifkan **tombol List Data**, yaitu tombol yang ketika di klik akan menampilkan semua data yang telah diinput dan menampilkan inputan nilai fungsi dari data-data tersebut. Selain itu juga terdapat **tombol Back To Menu**, yaitu tombol untuk kembali kepada tampilan menu utama, dan **tombol Close** untuk keluar dari menu utama.

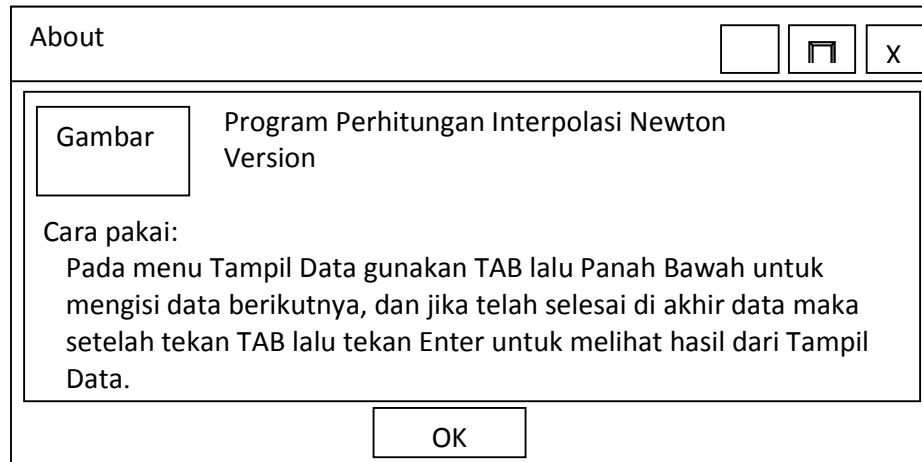
Gambar 5. Rancangan Form Input Data.

Gambar 6. Form Input Data

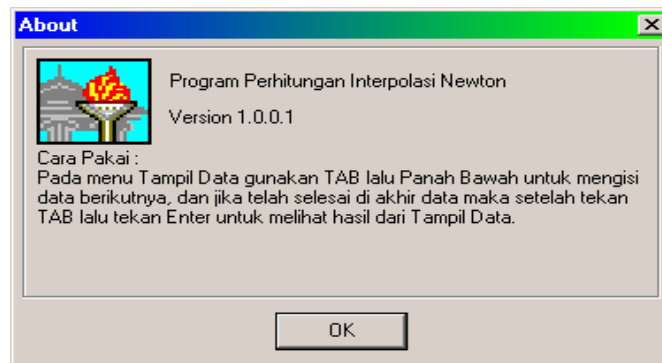
Gambar di atas terdiri atas judul form, tiga macam menu kontrol, satu label untuk judul tampilan, empat label untuk menginput data, dua GroupBox, empat komponen edit, dan empat komponen button.

Rancangan Form About

Rancangan tampilan **About** akan menampilkan *judul program*, dan *pembuat program* perhitungan Interpolasi Newton. Selain itu di sini juga terdapat *informasi* mengenai instruksi yang harus dilakukan pada saat memproses menu Tampil Data dan pada saat selesai memproses menu ini.



Gambar 7. Rancangan Form About.



Gambar 8. Form About

Gambar terdiri atas judul form, tiga macam kontrol menu (minimize, maximize, dan close), satu komponen image yang terletak di kiri atas form, tiga komponen label, dan satu komponen tombol (button).

Rancangan Form Tampil Data

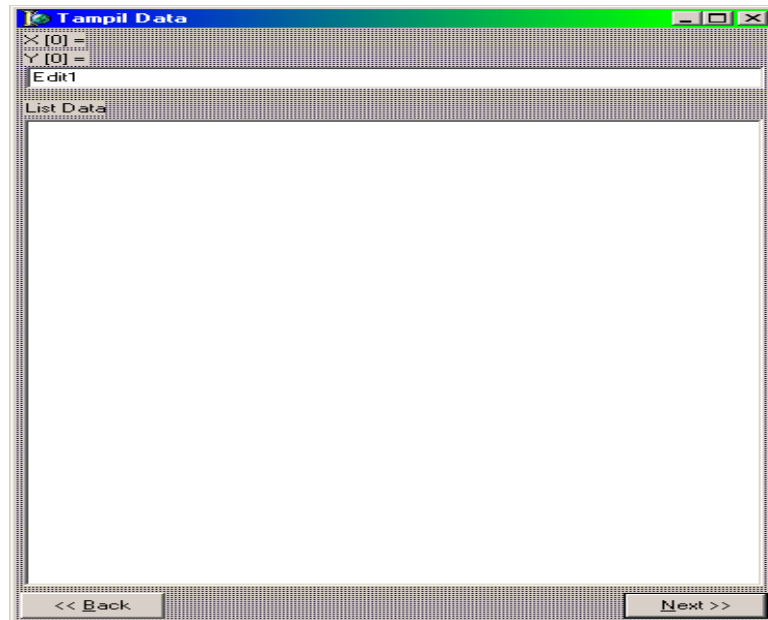
Rancangan **Tampil Data** akan menampilkan data-data yang telah dimasukkan pada menu Input Data, dalam hal ini adalah nilai dari $x[i]$ yang tampil secara otomatis sebagai akibat dari penginputan data yang dilakukan pada form Input Data. Selain itu terdapat sebuah kotak yang berfungsi untuk

menginput nilai fungsi dari data-data yang tampil secara otomatis tersebut, dalam hal ini nilai fungsi digambarkan sebagai $y[i]$, yang kemudian inputan tersebut akan ditampilkan pada kotak di bawah **List Data**. Pada rancangan ini juga terdapat tombol **<< Back** dan tombol **Next >>**, tombol **back** berfungsi untuk kembali kepada menu input data, sedangkan tombol **next** berfungsi untuk menampilkan tampilan berikutnya yaitu menu untuk menampilkan Persamaan Interpolasi Newton. Jika data yang diisi tidak lengkap, maka akan ditampilkan sebuah pesan.

Tampil Data

$x[1] =$	$y[1] =$
$x[2] =$	$y[2] =$
$x[3] =$	$y[3] =$
.	.
.	.
.	.
$x[i] =$	$y[i] =$

Gambar 9. Rancangan Form Tampil Data.



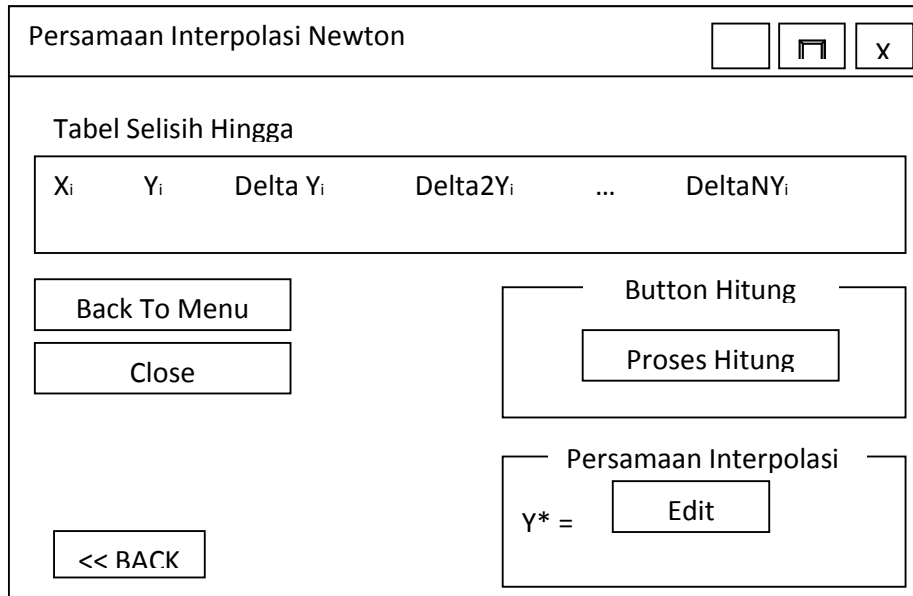
Gambar 10. Form Tampil Data

Gambar terdiri atas judul form, tiga menu kontrol, tiga komponen label, satu komponen edit, satu komponen ListBox untuk menampilkan $x[i]$ dan $y[i]$, dan dua komponen button.

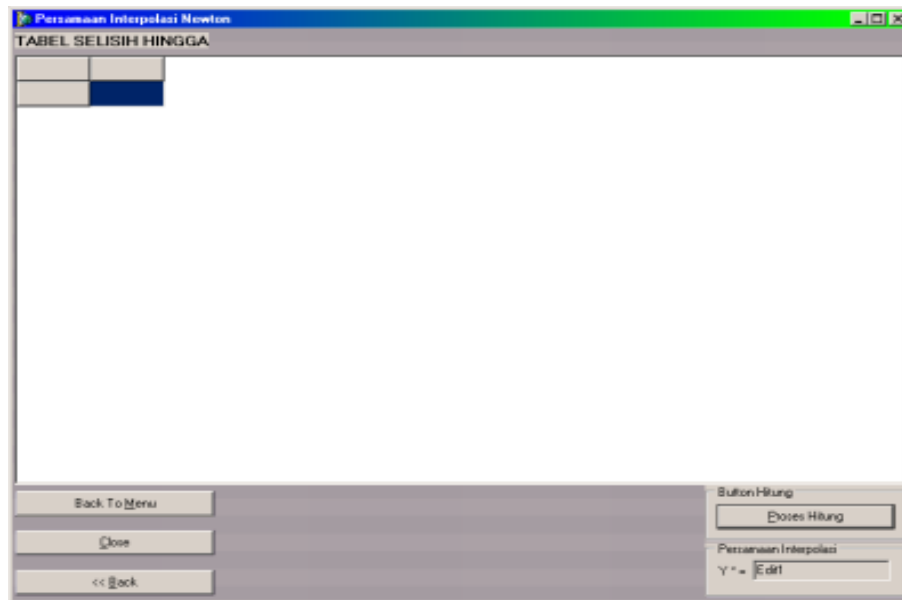
Rancangan Form Persamaan Interpolasi Newton

Rancangan ini akan menampilkan sebuah *tabel selisih hingga* yaitu tabel

selisih antara nilai suatu fungsi dengan nilai fungsi sebelumnya. Selain itu juga terdapat tombol **Back To Menu** untuk kembali kepada tampilan menu utama, **tombol Close** untuk keluar dari menu utama, tombol **Proses Hitung** untuk melakukan perhitungan Interpolasi Newton yang hasilnya akan ditampilkan pada komponen **edit**.



Gambar 11. Form Persamaan Interpolasi Newton.



Gambar 12. Form Persamaan Interpolasi Newton

Gambar terdiri atas judul form, satu menu kontrol yaitu menu close, dua komponen label untuk judul tabel dan tampilan fungsi y^* , satu komponen StringGrid untuk menampilkan tabel, lima komponen button, dua komponen GroupBox.

PENUTUP

Kesimpulan

Implementasi program aplikasi dengan menggunakan bahasa pemrograman visual misalnya Borland Delphi 5.0 sangat membantu dalam pembuatan aplikasi-aplikasi yang membutuhkan suatu antar muka yang memudahkan bagi setiap penggunanya. Pada penulisan ini dapat disimpulkan bahwa Interpolasi Newton berhasil dibuat dengan langkah-langkah pembuatan program yang menggunakan bahasa pemrograman Borland Delphi versi 5.0.

User dapat mengoperasikan program aplikasi ini dengan mudah, tanpa harus mempelajarinya secara khusus. Tampilan program aplikasi Interpolasi Newton dalam Borland Delphi 5.0 ini selain memberikan tampilan program yang menarik juga mempermudah user

tanpa harus melakukan perhitungan yang rumit.

Saran

Walaupun program ini telah selesai namun tidak mustahil untuk mengembangkannya agar lebih baik. Untuk itu penulis sarankan agar pembuatan tampilan atau antar muka lebih menarik, lebih efisien dan bervariasi yang akan menambah performansi sehingga tidak terkesan kaku, dengan memanfaatkan fasilitas yang terdapat pada pemrograman Borland Delphi 5.0

DAFTAR PUSTAKA

- Kadir, Abdul. 2001. **Dasar Pemograman Delphi 5.0 Jilid 1**, cetakan pertama. Yogyakarta: Andi.
- D. Suryadi H. S. 1995. **Pengantar Metode Numerik**, cetakan keempat. Jakarta: Gunadarma.
- Martina, Inge. 2001. **36 Jam Belajar Komputer Delphi 5.0**, cetakan kedua. Jakarta: PT. Elex Media Komputindo.