Laporan Penelitian

Analisis Ketunggalan Polinomial Interpolasi untuk Aproksimasi Fungsi

Peneliti:

Drs. Sahid, MSc.

Jurusan Pendidikan Matematika
Fakultas Matematika dan Ilmu Penbetahuan Alam
Universitas negeri Yogyakarta

===

Dilaksanakan dengan Dana DIK UNY Tahun 2003,
No. Kontrak: 1444/J35.13/PL/2003

Abstrak

Penelitian ini bertujuan untuk menyelidiki ketunggalan polinomial interpolasi, menentukan galat polinomial interpolasi sebagai hampiran nilai-nilai suatu fungsi, dan menyusun program komputer untuk membentuk dan menghitung nilai-nilai polinomial interpolasi.

Analisis teoritik memperlihatkan bahwa polinomial interpolasi Newton dan polinomial interpolasi Lagrange indentik dengan polinomial interpolasi bentuk baku. Dari hasil analisis perhitungan untuk memperoleh koefisien-koefisien dan perhitungan nilai-nilai polinomial interplasi dapat diketahui bahwa polinomial Newton (metode selisih terbagi Newton) merupakan polinomial interpolasi yang paling mudah dibentuk dan paling efisien dihitung, dibanding dengan kedua polinomial interpolasi yang lain. Galat hampiran nilai fungsi dengan menggunakan ketiga polinomial interpolasi merupakan polinomial yang perilakunya tergantung pada titik-titik interpolasi.

Polinomial interpolasi bentuk baku dan polinomial Lagrange tidak praktis digunakan dalam interpolasi dengan cacah titik interpolasi semakin bertambah. Meskipun demikian polinomial interpolasi Lagrange dapat digunakan untuk menunjukkan keberadaan polinomial interpolasi. Karena dibentuk secara rekursif, polinomial interpolasi Newton sangat efisien jika digunakan untuk pembentukan dan perhitungan polinomial-polinomial interpolasi dengan cacah titik interpolasi semakin bertambah.

Komputasi numerik dengan menggunakan program Matlab hasil implementasi ketiga polinomial interpolasi memperlihatkan kesesuaiannya dengan hasil analisis teoritis.

Daftar Isi

Kata Pengantar

Puji syukur Alhamdulillah penulis panjatkan ke Hadhirat Allah SwT. atas nikmat kesehatan dan kekuatan serta ilmu yang diberikan kepada peneliti, sehingga kegiatan penelitian dan penulisan laporan penelitian ini dapat diselesaikan.

Penelitian ini merupakan penelitian mandiri yang dilaksanakan dengan dana DIK UNY Tahun 2003, No. Kontrak: 1444/J35.13/PL/2003, dan dilaksanakan selama 6 (enam) bulan.

Peneliti mengucapkan banyak terima kasih kepada berbagai pihak yang terlibat dalam penelitian ini, baik lanhsung maupun tak langsung:

1. Dekan FMIPA UNY, yang telah memberikan kesempatan kepada peneliti dengan memberikan Dana sesuai kontrak yang telah disepakati kedua belah pihak.
2. Ketua Jurusan Pendidikan Matematika, yang telah memberikan layanan dan fasilitas yang diperlukan peneliti gunakan melaksanakan seminar proposal dan seminar hasil penelitian.
3. Para dosen Jurusan Pendidikan Matematika, atas partisipasinya dalam seminar proposal dan seminar penelitian dengan memberikan saran dan masukan yang terkait dengan penelitian ini.

Peneliti berharap agar hasil penelitian ini dapat dimanfaatkan sebagai bahan rujukan, baik oleh peneliti lain, dosen, maupun mahasiswa untuk kegiatan pengajaran mata kuliah yang terkait, misalnya Metode Numerik, atau untuk penelitian lain dalam bidang serupa. Meskipun demikian, peneliti menyadari bahwa tiada gading yang tak retak. Saran dan masukan dari berbagai pihak, apabila di dalam laporan ini terdapat kekeliruan atau kesalahan penulisan, peneliti dengan senang hati bersedia menerimanya untuk keperluan publikasi selanjutnya.

Yogyakarta, 5 Januari 2004

Peneliti

Bab I
Pendahuluan

A. Latar Belakang Masalah

Interpolasi adalah proses pencarian dan perhitungan nilai suatu fungsi berdasarkan nilai-nilai fungsi tersebut pada sekumpulan titik yang diberikan (tabel nilai fungsi). Nilai-nilai fungsi tersebut mungkin merupakan hasil eksperimen dalam sebuah percobaan atau diperoleh melalui pengamatan dan pencatatan, misalnya suhu di suatu tempat, cacah kendaraan yang melewati sebuah ruas jalan raya, cacah penduduk di suatu daerah. Dalam hal ini, rumus fungsi-fungsi yang terkait tidak diketahui secara eksplisit (misalnya sebagai fungsi waktu), dan pengamatan/pencatatan tidak mungkin dilakukan sepanjang waktu, melainkan hanya pada waktu-waktu tertentu. Untuk mengetahui nilai-nilai fungsi pada waktu-waktu selain saat dilakukan pengamatan/pencatatan dapat digunakan interpolasi.

Fungsi interpolasi biasanya dipilih dari sekelompok fungsi tertentu, salah satunya adalah fungsi polinomial yang paling banyak dipakai. Fungsi-fungsi polinomial banyak dipakai dalam interpolasi, karena fungsi-fungsi tersebut mudah dihitung nilainya (hanya menggunakan empat operasi aritmetika dasar: penjumlahan/pengurangan, perkalian/ pembagian). Selain itu, polinomial mudah diturunkan, diintegralkan, dan perilakunya baik - semua turunannya ada dan kontinyu.

Interpolasi digunakan untuk menyelesaikan berbagai masalah dalam bidang teori hampiran yang lebih umum. Berikut adalah beberapa masalah hampiran (aproksimasi) dan kemungkinan pemakaian interpolasi untuk menyelesaikannya.

(1) Diberikan sebuah tabel nilai-nilai $\{(x_k, f(x_k)) : k = 0,1,2,3,4,5\}$, misalnya {(0, 1.23), (1, 2.0), (3,1.53), (4,2.50), (5,2.75)}, interpolasi dapat digunakan untuk mencari nilai-nilai $f(x)$ untuk nilai-nilai x yang tidak terdapat di dalam tabel.
(2) Diberikan sekumpulan data berupa titik-titik (koordinat), seperti di atas, tentukan sebuah fungsi mulus f yang tidak bergoyang (osilatif) dan cocok dengan data tersebut baik secara eksak maupun hampiran.
(3) Diberikan sebuah fungsi f, misalkan $f(x) = e^{\cos(x)}$ dan diperlukan suatu cara untuk menghitung nilai-nilai fungsi tersebut menggunakan komputer. Oleh karena komputer hanya menggunakan operasi aritmetika dasar (penjumlahan/pengurangan dan perkalian/pembagian), maka nilai-nilai fungsi tersebut dihitung dengan menggunakan polinomial hampiran.
(4) Untuk mengintegralkan atau menurunkan suatu fungsi secara numerik, kita sering mengganti fungsi yang bersangkutan dengan ekspresi hampiran yang lebih sederhana, yang biasanya diperoleh dengan menggunakan interpolasi. Beberapa metode numerik untuk menyelesaikan persamaan diferensial yang dipakai secara luas diperoleh dari hampiran interpolasi.

Polinomial $P(x)$ yang menghampiri $f(x)$ pada interval $a \le x \le b$ dapat dicari dengan empat macam metode:

(1) polinomial Taylor

$$P_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(c)}{k!}(x-c)^k, \quad a \le c \le b,$$

(2) metode interpolasi (kolokasi), yakni bahwa kurva polinomial $y = P(x)$ melewati sejumlah titik pada kurva $y = f(x)$,

(3) metode minimax, yakni bahwa polinomial $P(x)$ memenuhi

$$P(x) = \min_{derajad(p) \le derajad(P)} \{\max_{a \le x \le b} |f(x) - p(x)|\}, \text{ dan}$$

(4) metode kuadrat terkecil, yakni bahwa polinomial $P(x)$ memenuhi

$$\int_a^b [f(x) - P(x)]^2 dx = \min_{derajad(p) \le derajad(P)} \int_a^b [f(x) - p(x)]^2 dx.$$

Metode kuadrat terkecil mengarah kepada pemakaian polinomial Legendre, metode minimax mengarah kepada polinomial Chebyshev, sedangkan polinomial-polinomial interpolasi dapat dihitung dengan beberapa cara. Polinomial Taylor dapat memberikan hampiran yang cukup akurat sesuai kriteria yang diberikan, namun tidak praktis jika dipakai untuk menghitung hampiran nilai fungsi di beberapa titik.

Sebagaimana diketahui dalam aljabar, bahwa jika diketahui n buah titik yang berlainan dapat dibentuk sebuah polinomial yang berderajad paling tinggi $n-1$. Di dalam metode numerik terdapat beberapa metode (rumus) untuk mendapatkan polinomial yang menginterpolasikan sejumlah titik. Di dalam buku-buku teks metode numerik, polinomial ini dapat dinyatakan dalam setidaknya tiga macam bentuk. Misalkan diketahui $n+1$ titik yang berlainan $\{(x_i, y_i) \mid i = 1, 2, 3, ..., n+1;\ x_i \ne x_j \text{ jika } i \ne j\}$. Polinomial yang menginterpolasikan $n+1$ titik tersebut dapat dihitung dengan salah satu cara sebagai berikut.

(1) Polinomial bentuk baku,

$$P_n(x) = a_1 + a_2 x + a_3 x^2 + ... + a_{n+1} x^n, \tag{1}$$

dengan koefisien-koefisien a_k diperoleh dari penyelesaian sistem persamaan linier

$$\begin{pmatrix} 1 & x_1 & \cdots & x_1^n \\ 1 & x_2 & \cdots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n+1} & \cdots & x_{x+1}^n \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n+1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n+1} \end{pmatrix}. \tag{2}$$

(2) Polinomial Newton (Atkinson, 1993: 114-115; Mathews, 1992: 230):

$$Q_n(x) = y_1 + \sum_{k=1}^{n} \left(f[x_1, x_2, ..., x_{k+1}] \prod_{i=1}^{k} (x - x_i) \right) \tag{3}$$

dengan $f[\ ]$ *menyatakan selisih terbagi Newton*

(3) Polinomial Lagrange

$$R_n(x) = \sum_{k=1}^{n+1} y_k L_k(x) \text{ dengan } L_k(x) = \frac{\prod_{i=1, i \neq k}^{n+1} (x - x_i)}{\prod_{i=1, i \neq k}^{n+1} (x_k - x_i)}. \qquad (4)$$

Adalah suatu hal yang menarik, yakni menyelidiki kesamaan, efisiensi dan efektivitas ketiga polinomial $P_n(x), Q_n(x),\ dan\ R_n(x)$. Apabila ternyata ketiga polinomial tersebut identik, maka dapat dipilih polinomial mana yang paling mudah dipakai dan paling efisien untuk menghitung hampiran nilai-nilai $f(x)$. Penelitian ini bermaksud untuk melakukan penyelidikian tersebut.

B. Rumusan Masalah

Permasalahan yang hendak dikaji dalam penelitian ini meliputi:

(1) Apakah polinomial interpolasi Newton dan Lagrange identik dengan polinomial interpolasi dalam bentuk baku?
(2) Polinomial interpolasi manakah yang paling mudah dibentuk (memerlukan paling sedikit operasi hitung untuk menentukan koefisien-koefisiennya)?
(3) Polinomial interpolasi manakah yang paling efisien dihitung (memerlukan paling sedikit operasi hitung untuk menghitung nilainya)?
(4) Berapakah galat hampiran nilai fungsi yang diperoleh dengan menggunakan polinomial interpolasi?

C. Tujuan Penelitian

Penelitian dilaksanakan dengan tujuan sebagai berikut:

(1) untuk memperlihatkan bahwa polinomial interpolasi Newton dan Lagrange identik dengan polinomial interpolasi dalam bentuk baku;
(2) untuk menganalisis kompleksitas (sebagai fungsi banyaknya titik interpolasi) dalam bentuk ketiga polinomial interpolasi;
(3) untuk menganalisis kompleksitas (sebagai fungsi banyaknya titik interpolasi) perhitungan hampiran nilai suatu fungsi dengan ketiga polinomial interpolasi;
(4) untuk menganalisis galat hampiran nilai fungsi yang diperoleh dengan menggunakan polinomial interpolasi;
(5) menyusun program komputer yang dapat digunakan untuk menghitung koefisien-koefisien polinomial interpolasi dan menghitung nilai polinomial interpolasi.

D. Manfaat Hasil Penelitian

Hasil penelitian ini akan dapat memberikan penjelasan yang terperinci tentang ketunggalan polinomial interpolasi, yang pada beberapa buku teks Metode Numerik biasanya tidak dijelaskan secara terperinci. Selain itu, program komputer yang disusun sangat bermanfaat untuk menentukan koefisien-koefisien dan menghitung nilai-nilai polinomial yang menginterpolasikan sejumlah titik pada suatu interval. Program ini dapat digunakan oleh siapa saja yang menghadapi masalah interpolasi.

Dengan demikian hasil penelitian ini akan dapat menambah wawasan bagi pembaca tentang sifat-sifat polinomial interpolasi, sehingga dapat memilih polinomial interpolasi yang tepat untuk menghitung hampiran nilai suatu fungsi. Penelitian ini juga dapat menjadi contoh untuk penelitian-penelitian lain dalam metode numerik, khususnya tentang polinomial interpolasi sebagai salah satu metode hampiran fungsi. Pada akhirnya akan terbuka luas khasanah penelitian dalam metode numerik.

Selain manfaat teoritis, manfaat praktis hasil penelitian ini adalah tersedianya program komputer untuk menyelesaikan masalah interpolasi dengan menggunakan polinomial.

E. Pembatasan Permasalahan

Meskipun terdapat beberapa polinomial yang dapat digunakan sebagai hampiran fungsi pada suatu interval, namun dalam penelitian ini hanya akan dibahas polinomial interpolasi, yang disusun berdasarkan data sejumlah titik (nilai-nilai fungsi yang akan diinterpolasikan). Penelitian ini tidak pembahas hampiran fungsi dengan polinomial Taylor maupun hampiran fungsi dengan metode minimax dan kuadrat terkecil.

Pembatasan lain adalah bahwa fungsi-fungsi yang dibahas dalam penelitian ini adalah fungsi-fungsi riil dengan domain himpunan bilangan riil. Penelitian ini tidak membahas interpolasi fungsi kompleks.

Bab II
Kajian Pusataka

Pembahasan metode numerik untuk mencari hampiran nilai fungsi pada suatu interval memerlukan beberapa pengertian dasar sebagai berikut.

Definisi 1 (Akar Persamaan, Pembuat Nol Fungsi, Mathews, 1992: 55)

*Misalkan f adalah suatu fungsi kontinyu. Setiap bilangan r pada domain f yang memenuhi $f(r) = 0$ disebut **akar persamaan** $f(x) = 0$, atau juga disebut **pembuat nol** $f(x)$.*

Definisi 2 (Derajad Akar Persamaan, Atkinson, 1993: 94; Mathews, 1992: 76)

Misalkan r adalah akar persamaan $f(x) = 0$. Jika terdapat bilangan asli m dan fungsi kontinyu h dengan $h(r) \neq 0$, sedemikian hingga $f(x)$ dapat dinyatakan sebagai

$$f(x) = (x - r)^m h(x), \tag{5}$$

*maka r disebut **akar berderajad** m.*

Dari (5) terlihat bahwa jika r pembuat nol $f(x)$ yang berderajad m, maka

$$f(r) = f'(r) = ... = f^{(m-1)}(r) = 0, \text{ dan } f^m(r) \neq 0.$$

Jika $m = 1$, maka r disebut **akar sederhana**. Jika $m > 1$, maka r disebut **akar ganda**. Untuk $m = 2$, maka r disebut **akar dobel**, dst.

Definisi 3 (Bentuk baku polinomial, Conte & de Boor, 1981: 32)

Suatu polinomial $P_n(x)$ yang berderajad kurang atau sama dengan n dalam bentuk baku adalah suatu fungsi yang dituliskan dalam bentuk

$$P_n(x) = a_1 + a_2 x + a_3 x^2 + ... + a_{n+1} x^n \tag{6}$$

dengan koefisien-koefisien $a_1, a_2, ..., a_{n+1}$ bilanyan nyata. Apabila $a_{n+1} \neq 0$, maka polinomial tersebut dikatakan berderajad (tepat) n.

Definisi 4 (Perkalian tersarang, Conte & de Boor, 1981: 33-34; Gerald & Wheatly, 1994: 56)

*Salah satu metode untuk menghitung nilai polinomial (6) adalah menggunakan teknik **perkalian** tersarang*

$$P_n(x) = a_1 + x(a_2 + x(a_3 + ... + x(a_n + a_{n+1}x)...)). \tag{7}$$

Dari (7) dapat disusun suatu **algoritma perkalian tersarang** untuk menghitung nilai polinomial $P_n(x)$ di suatu titik z sekaligus hasil baginya oleh $(x - z)$, yakni dengan mendefinisikan barisan

$$\boxed{b_{n+1} = a_{n+1},\ dan\ b_k = a_k + b_{k+1}z,\ untuk\ k = n, (n-1), \ldots, 1,} \tag{8}$$

sehingga diperoleh

$$P_n(z) = b_1, \tag{9}$$

dan

$$P_n(x) = b_1 + (x - z)q(x) \tag{10}$$

dengan $q(x)$ adalah polinomial hasil bagi $P_n(x)$ oleh $(x - z)$ dan dapat dinyatakan dengan

$$q(x) = b_2 + b_3x + b_4x^2 + \ldots + b_{n+1}x^{n-1}. \tag{11}$$

Dari (9) dan (10) terlihat bahwa apabila z adalah akar (pembuat nol) polinomial $P_n(x)$, yakni $P_n(z) = 0$, maka

$$P_n(x) = (x - z)q(x). \tag{12}$$

Lemma berikut menjelaskan kasus umum dari (12).

Lemma 1. (Conte & de Boor, 1981: 35)

Apabila $z_1, z_2, \ldots, z_n$ adalah akar-akar berlainan polinomial $P(x)$, maka

$$P(x) = (x - z_1)(x - z_2)\ldots(x - z_n)r(x) \tag{13}$$

untuk suatu polinomial $r(x)$.

Misalkan $p(x)$ dan $q(x)$ adalah dua buah polinomial berderajad kurang atau sama dengan n dan $p(z_i) = q(z_i)$ untuk $i = 1, 2, \ldots, n+1$ dengan $z_i \neq z_j$ $jika$ $i \neq j$. Selanjutnya, didefinisikan polinomial $d(x) = p(x) - q(x)$. Jadi, $z_1, z_2, \ldots, z_{n+1}$ adalah akar-akar berlainan polinomial $d(x)$, sehingga dapat dinyatakan dalam bentuk (13) untuk suatu polinomial $r(x)$. Dalam hal ini, ruas kanan $d(x)$ memuat $n+1$ faktor dalam x selain faktor $r(x)$, padahal $d(x)$ adalah selisih dua polinomial berderajad kurang atau sama dengan n, sehingga derajad $d(x)$ adalah kurang atau sama dengan n. Hal ini tidaklah mungkin kecuali $d(x) = 0$, yang berarti $p(x) = q(x)$. Hal ini dinyatakan sebagai akibat Lemma di atas, sebagai berikut.

Akibat 1 (Conte & de Boor, 1981, 36)

Jika $p(x)$ dan $q(x)$ adalah dua buah polinomial berderajad kurang atau sama dengan n dan $p(z_i) = q(z_i)$ untuk $i = 1, 2, \ldots, n+1$ dengan $z_i \neq z_j$ untuk $i \neq j$, maka $p(x)$ dan $q(x)$ identik.

Lemma 2. (Conte & de Boor, 1981: 36)

Apabila $z_1, z_2, \ldots, z_n$ adalah akar-akar (mungkin ada yang sama) polinomial $P(x)$, maka

$$P(x) = (x - z_1)(x - z_2)\ldots(x - z_n)r(x) \tag{14}$$

untuk suatu polinomial $r(x)$.

Akibat 2 (Conte & de Boor, 1981, 37)

Jika $p(x)$ dan $q(x)$ adalah dua buah polinomial berderajad kurang atau sama dengan n dan $p(z_i) = q(z_i)$ untuk $i = 1, 2, \ldots, n + 1$, maka $p(x)$ dan $q(x)$ identik.

Definisi 5 (Selisih Terbagi Newton, Atkinson, 1993: 111-112; Mathews, 1992: 229)

Selisih terbagi Newton tingkat pertama dan kedua fungsi f terhadap simpul-simpul a, b, c didefinisikan berturut-turut sebagai

$$f[a,b] = \frac{f(b) - f(a)}{b - a}, \tag{15}$$

$$f[a,b,c] = \frac{f[b,c] - f[a,b]}{c - a}, \tag{16}$$

dan secara rekursif didefinisikan

$$f[x_1, x_2, x_3, \ldots, x_n] = \frac{f[x_2, x_3, \ldots, x_n] - f[x_1, x_2, \ldots, x_{n-1}]}{x_n - x_1}. \tag{17}$$

Teorema 1 (Teorema Nilai Rata-rata, Mathews, 1992: 5)

Jika f adalah fungsi kontinyu pada interval $[a,b]$ dan $f'(x)$ ada untuk semua $a < x < b$, maka terdapat sebuah bilangan c, dengan $a < c < b$, sedemikian hingga

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

Untuk kasus $f(a) = f(b)$, teorema nilai rata-rata menjadi **Teorema Role**.

Lemma 3. (Atkinson, 1993: 111-112)

Misalkan f dan dua turunan pertamanya kontinyu pada interval yang memuat titik-titik x_1, x_2, dan x_3. Dari definisi (15) dan (16) dapat diturunkan sifat-sifat

$$f[x_1, x_2] = f'(x), \tag{18}$$

dengan x adalah bilangan antara x_1 dan x_2, dan

$$f[x_1, x_2, x_3] = \frac{f''(z)}{2}, \tag{19}$$

dengan $\min(x_1, x_2, x_3) \le z \le \max(x_1, x_2, x_3)$.

Bukti:

Sifat (18) dapat diperoleh dengan menggunakan Teorema Nilai Rata-rata, yakni oleh karena f dan f' kontinyu pada interval yang memuat x_1 dan x_2, maka terdapat bilangan x antara x_1 dan x_2 sedemikian hingga

$$f'(x) = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = f[x_1, x_2],$$

dan kesamaan terakhir diperoleh dari definisi (15). Selanjutnya, sifat (19) dapat dibuktikan sebagai berikut. Definisikan fungsi-fungsi $E(t)$ dan $G(t)$ sebagai berikut:

$$E(t) = (t - x_1)(t - x_2)f[x_1, x_2, t], \tag{20}$$

dan

$$G(t) = E(t) - \frac{(t - x_1)(t - x_2)}{(x_3 - x_1)(x_3 - x_2)} E(x_3). \tag{21}$$

Fungsi-fungsi $E(t)$ dan $G(t)$ tersebut memenuhi sifat-sifat sebagai berikut:

1. Karena f, f' dan f'' kontinyu, maka $G(t), G'(t)\ dan\ G''(t)$ juga kontinyu pada interval yang sama.
2. $E(x_1) = E(x_2) = 0\ sehingga\ G(x_1) = G(x_2) = G(x_3) = 0.$

Dengan menggunakan Teorema Nilai Rata-rata dapat diperoleh nilai x_1 antara $x_1\ dan\ x_2$ dan $x_2\ antara\ x_2\ dan\ x_3$ sedemikian hingga $G'(x_1) = G'(x_2) = 0.$ Dengan menggunakan Teorema Nilai Rata-rata lagi pada $G'(t)$, dapat diperoleh z antara $x_1\ dan\ x_2$, atau $min(x_1, x_2, x_3) \le z \le max(x_1, x_2, x_3)$, sedemikian hingga $G''(z) = 0.$

Selanjutnya, dengan sedikit perhitungan aljabar dapat ditunjukkan bahwa $E''(t) = f''(t)$, sehingga

$$G''(t) = E''(t) - \frac{2E(x_3)}{(x_3 - x_1)(x_3 - x_2)} = f''(t) - \frac{2E(x_3)}{(x_3 - x_1)(x_3 - x_2)}. \tag{22}$$

Dari (20) dan (22) dan dengan mengingat $G''(z) = 0$, diperoleh

$$(x_3 - x_1)(x_3 - x_2)f[x_1, x_2, x_3] = E(x_3) = (x_3 - x_1)(x_3 - x_2)\frac{f''(z)}{2}, \tag{23}$$

sehingga diperoleh (19). $\square$

Bab III
Metode Penelitian

A. Materi / Bahan Penelitian

Penelitian ini merupakan penelitian matematis dan komputasi, yang dikerjakan dengan cara melakukan analisis matematis menggunakan penarikan kesimpulan secara deduktif. Kesimpulan dari hasil penalaran deduktif bersifat determinitif, bukan probabilistik, sehingga tidak diperlukan pengujian hipotesis secara statistika. Hasil-hasil komputasi digunakan untuk mengkonfirmasikan hasil-hasil analisis matematis.

Oleh karena itu, materi penelitian ini berupa definisi, aksioma, dan fakta-fakta matematika lain dalam metode numerik, khususnya konsep-konsep yang terkait dengan polinomial dan interpolasi. Informasi-informasi tersebut biasanya terdapat di dalam buku-buku teks matematika secara umum, atau Metode Numerik khususnya, dan jurnal-jurnal matematika dan / atau metode numerik. Selain informasi tercetak, juga terdapat informasi *online*, yakni Internet, yang merupakan sumber informasi melimpah untuk mendukung kegiatan penelitian.

Analisis matematis (penalaran deduktif) dilakukan di atas kertas, sehingga diperlukan kertas dan pena untuk melakukan penelitian ini. Komputasi dilakukan dengan menggunakan bantuan komputer dan program komputer (*software*). Program komputer yang akan disusun untuk menghitung koefisien-koefisien dan nilai-nilai polinomial interpolasi ditulis dengan menggunakan software MATLAB, yang merupakan paket spesifik matematika yang cocok untuk keperluan komputasi numerik.

Penelitian ini tidak memerlukan instrumen yang berupa angket, formulir, soal-soal tes, dan alat-alat pengambilan data statistika sejenis. Dalam penelitian ini tidak diperlukan data statistika yang diperoleh dengan menggunakan instrumen-instrumen penelitian tersebut maupun data statistika sekunder.

B. Cara Penelitian

Penelitian ini merupakan penelitian matematika yang bersifat deduktif. Kesimpulan-kesimpulan yang diperoleh merupakan hasil proses penalaran secara deduktif berdasarkan fakta-fakta matematika yang berupa definisi, aksioma, notasi matematika, dan teorema-teorema yang sudah dibuktikan. Dalam penelitian ini tidak dilakukan analisis statistiks untuk pengambilan kesimpulan, sehingga tidak diperlukan populasi dan pengambilan sampel.

Secara terperinci, langkah-langkah penelitian ini meliputi:

1. Pengumpulan definisi dan aksioma yang diperlukan, khususnya tentang pengertian interpolasi dan polinomial interpolasi;
2. Analisis deduktif untuk mendapatkan polinomial interpolasi dalam bentuk baku;
3. Analisis deduktif untuk mendapatkan polinomial interpolasi Newton;
4. Analisis deduktif untuk mendapatkan polinomial interpolasi Lagrange;
5. Analisis deduktif untuk menunjukkan kesamaan polinomial interpolasi dalam bentuk Newton dan Lagrange dengan bentuk baku;
6. Analisis deduktif untuk mendapatkan galat hampiran nilai fungsi dengan menggunakan polinomial interpolasi;

7. Analisis deduktif untuk membandingkan efektivitas dan efisiensi ketiga bentuk polinomial interpolasi untuk menghitung hampiran nilai fungsi;
8. Penyusunan program MATLAB yang dapat digunakan untuk menentukan koefisien-koefisien dan menghitung nilai polinomial interpolasi; dan
9. Penggunaan program-program MATLAB tersebut untuk menyelesaikan beberapa masalah interpolasi dan membandingkan hasilnya dengan nilai-nilai fungsi yang sebenarnya.

Bab IV
Hasil Penelitian

A. Eksistensi dan Ketunggalan Polinomial Interpolasi

Misalkan diketahui himpunan $n+1$ titik: $\{(x_i, y_i) \mid i = 1, 2, 3, \ldots, n+1;\ x_i \neq x_j \text{ jika } i \neq j\}$. Titik-titik tersebut berlainan. Kita akan membentuk suatu polinomial $P(x)$ berderajad kurang atau sama dengan n yang menginterpolasikan $n+1$ titik tersebut, yakni memenuhi:

$$P(x_i) = y_i,\ \text{untuk } i = 1, 2, 3, \ldots, n+1.$$

Untuk tujuan ini kita pilih **polinomial bentuk Lagrange** (4), yang ternyata memenuhi

$$L_k(x_i) = \frac{\prod_{i=1, i\neq k}^{n+1} (x_i - x_i)}{\prod_{i=1, i\neq k}^{n+1} (x_k - x_i)} = \begin{cases} 1, \text{ jika } i = k \\ 0, \text{ jika } i \neq k \end{cases}, \quad i = 1, 2, 3, \ldots, n+1,$$

sehingga

$$R_n(x_i) = \sum_{k=1}^{n+1} y_k L_k(x_i) = y_i \quad \text{untuk } i = 1, 2, 3, \ldots, n+1.$$

Ini menunjukkan bahwa polinomial $R_n(x)$ menginterpolasikan $n+1$ titik yang diketahui tersebut. Selanjutnya, dari (4) terlihat bahwa $L_k(x)$ merupakan hasilkali n faktor linier, sehingga ia merupakan suatu polinomial berderajad **tepat** n. Dengan demikian $R_n(x)$ juga merupakan suatu polinomial berderajad **tepat** n karena ia merupakan jumlah $n+1$ polinomial berderajad **tepat** n.

Jadi, dapat disimpulkan bahwa terdapat **sedikitnya** sebuah polinomial berderajad kurang atau sama dengan n yang menginterpolasikan $n+1$ titik di atas. Sementara itu, dari Akibat 2 dapat diketahui bahwa terdapat **paling banyak** sebuah polinomial berderajad kurang atau sama dengan n yang menginterpolasikan $n+1$ titik berlainan. Kesimpulan dari kedua hal tersebut dinyatakan dalam teorema berikut ini.

Teorema 2 (Eksistensi dan Ketunggalan Polinomial Interpolasi)
*Terdapat **tepat sebuah** polinomial berderajad kurang atau sama dengan n yang menginterpolasikan $n+1$ titik berlainan $\{(x_i, y_i) \mid i = 1, 2, 3, \ldots, n+1;\ x_i \neq x_j \text{ jika } i \neq j\}$. Polinomial interpolasi ini dapat dinyatakan dalam bentuk Lagrange (4).*

Selanjutnya akan ditinjau polinomial baku (1) dan polinomial Newton (3). Dari (1) diperoleh, untuk $k = 1, 2, 3, \ldots, n+1$,

$$P_n(x_k) = a_1 + a_2 x_k + a_3 x_k^2 + \ldots + a_{n+1} x_k^n = y_k$$

karena koefisien-koefisien a_k merupakan penyelesaian dari (2). Sistem persamaan linier (2) dijamin mempunyai penyelesaian karena matriks koefisiennya merupakan matriks

Vandermonde non-singular, mengingat $x_1 \neq x_2 \neq x_3 \neq \dots \neq x_{n+1}$ sehingga setiap persamaan bebas terhadap persamaan-persamaan yang lain. Jadi, polinomial bentuk baku $P_n(x)$ yang didefinisikan pada (1) menginterpolasikan $n+1$ tersebut di atas. Jelas bahwa $P_n(x)$ berderajad kurang atau sama dengan n.

Berikutnya, akan ditinjau polinomial Newton (3). Sekarang akan ditunjukkan bahwa (3) benar-benar menginterpolasikan $n+1$ titik tersebut di atas. Untuk tujuan ini diperlukan lemma di bawah ini.

Lemma 4 (Rumus Eksplisit Selisih Terbagi Newton)

Jika diketahui $n+1$ *titik berlainan* $\{(x_i, y_i) \mid i = 1,2,3,\dots,n+1;\ x_i \neq x_j \text{ jika } i \neq j\}$ *dengan* $y_k = f(x_k)$ *untuk* $k = 1,2,3,\dots,n+1$, *maka*

$$f[x_1, x_2, x_3, \dots, x_n, x_{n+1}] = \sum_{k=1}^{n+1} \frac{y_k}{\prod\limits_{j=1, j\neq k}^{n+1} (x_k - x_j)}. \tag{24}$$

Bukti:

Bukti (24) dapat diberikan secara induktif terhadap nilai-nilai n. Untuk $n = 1$, diperoleh

$$f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{y_1}{x_1 - x_2} + \frac{y_2}{x_2 - x_1}.$$

Misalkan (24) berlaku untuk $n = r$, yakni

$$f[x_1, x_2, x_3, \dots, x_r, x_{r+1}] = \sum_{k=1}^{r+1} \frac{y_k}{\prod\limits_{j=1, j\neq k}^{r+1} (x_k - x_j)}.$$

Selanjutnya, untuk $n = r+1$ kita lihat

$$\begin{aligned}
f[x_1, x_2, x_3, \dots, x_{r+2}] &= \frac{f[x_2, x_3, \dots, x_{r+2}] - f[x_1, x_2, \dots, x_{r+1}]}{x_{r+2} - x_1} \\
&= \frac{1}{x_{r+2} - x_1} \left\{ \sum_{k=2}^{r+2} \frac{y_k}{\prod\limits_{j=2, j\neq k}^{r+2} (x_k - x_j)} - \sum_{k=1}^{r+1} \frac{y_k}{\prod\limits_{j=1, j\neq k}^{r+1} (x_k - x_j)} \right\} \\
&= \frac{1}{x_{r+2} - x_1} \left\{ \frac{-y_1}{\prod\limits_{j=1}^{r+1} (x_1 - x_j)} + \frac{y_2 \left[(x_2 - x_1) - (x_2 - x_{r+2}) \right]}{\prod\limits_{j=1, j\neq 2}^{r+2} (x_2 - x_j)} \right. \\
&\qquad \left. + \frac{y_3 \left[(x_3 - x_1) - (x_3 - x_{r+2}) \right]}{\prod\limits_{j=1, j\neq 3}^{r+2} (x_2 - x_j)} + \dots + \frac{y_{r+2}}{\prod\limits_{j=2}^{r+1} (x_{r+2} - x_j)} \right\}
\end{aligned}$$

$$f[x_1,x_2,x_3,\ldots,x_{r+2}]=\frac{y_1}{\prod_{j=2}^{r+2}(x_1-x_j)}+\frac{y_2}{\prod_{j=1,j\neq 2}^{r+2}(x_2-x_j)}+\frac{y_3}{\prod_{j=1,j\neq 3}^{r+2}(x_3-x_j)}+\ldots+\frac{y_{r+2}}{\prod_{j=1}^{r+1}(x_{r+2}-x_j)}$$
$$=\sum_{k=1}^{r+2}\frac{y_k}{\prod_{j=1,j\neq k}^{r+2}(x_k-x_j)}.$$

Perhitungan di atas dilakukan dengan mengelompokkan suku-suku yang memuat y_1, y_2, y_3, ..., y_{r+2}. □

Sekarang akan dibuktikan teorema di bawah ini.

Teorema 3 (Keberadaaan Polinomial Interpolasi Newton)

Jika diketahui $n+1$ titik berlainan $\{(x_i,y_i)\,|\,i=1,2,3,\ldots,n+1;\ x_i\neq x_j \text{ jika } i\neq j\}$ dengan $y_k=f(x_k)$ untuk $k=1,2,3,\ldots,n+1$, maka polinomial $Q_n(x)$ yang menginterpolasikan $n+1$ titik tersebut dapat dinyatakan dalam bentuk

$$Q_n(x)=Q_{n-1}(x)+f[x_1,x_2,x_3,\ldots,x_{n+1}]\prod_{i=1}^{n}(x-x_i) \qquad (25)$$

dengan $Q_{n-1}(x)$ menginterpolasikan (x_k,y_k) untuk $k=1,2,3,\ldots,n$.

Bukti:

$Q_n(x)$ adalah polinomial berderajad kurang atau sama dengan n dengan $Q_n(x_k)=y_k$ untuk $k=1,2,3,\ldots,n+1$ dan $Q_{n-1}(x)$ adalah polinomial berderajad kurang atau sama dengan $(n-1)$ dengan $Q_{n-1}(x_k)=y_k$ untuk $k=1,2,3,\ldots,n$. Selanjutnya, definisikan polinomial

$$g(x)=Q_n(x)-Q_{n-1}(x).$$

Jelas bahwa $g(x)$ adalah suatu polinomial berderajad paling tinggi n, dan mempunyai akar $x_1,x_2,x_3,\ldots,x_n$, karena $g(x_k)=Q_n(x_k)-Q_{n-1}(x_k)=y_k-y_k=0$ untuk $k=1, 2, 3, .n$. Dengan demikian $g(x)$ dapat dituliskan dalam bentuk

$$g(x)=A_n\prod_{k=1}^{n}(x-x_k)$$

untuk suatu konstanta pengali A_n (lihat
Lemma 1 dan Lemma 2), sehingga

$$Q_n(x)=Q_{n-1}(x)+A_n\prod_{k=1}^{n}(x-x_k).$$

Jadi, A_n merupakan koefisien x^n mengingat $Q_{n-1}(x)$ berderajad kurang atau sama dengan $(n-1)$ dan $Q_n(x)$ adalah polinomial berderajad kurang atau sama dengan n. Sementara itu, pada Teorema 2 sudah dibuktikan bahwa polinomial interpolasi bersifat tunggal dan dapat dinyatakan dalam bentuk Lagrange (4). Koefisien x^n pada polinomial interpolasi bentuk Lagrange (4) adalah $\sum_{k=1}^{n+1} \frac{y_k}{\prod_{j=1, j\neq k}^{n+1} (x_k - x_j)}$. Mengingat ketunggalan polinomial interpolasi, maka dengan menggunakan hasil pada Lemma 4 diperoleh

$$A_n = \sum_{k=1}^{n+1} \frac{y_k}{\prod_{j=1, j\neq k}^{n+1} (x_k - x_j)} = f[x_1, x_2, x_3, \ldots, x_{n+1}],$$

dan (25) pun telah terbukti. ∎

Dengan memperhatikan persyaratan polinomial (25), jelaslah bahwa polinomial $Q_n(x)$ tidak lain adalah polinomial Newton (3). Dari syarat-syarat $Q_n(x)$ pada Teorema 3 dapat diketahui fakta-fakta sebagai berikut:

1. $Q_1(x_1) = y_1$ dan $Q_1(x_2) = y_2$
2. $Q_2(x_1) = y_1, \quad Q_2(x_2) = y_2, \quad dan \quad Q_2(x_3) = y_3$
3. Oleh karena $Q_n(x)$ adalah polinomial berderajad kurang atau sama dengan n yang dapat dibentuk secara rekursif dengan rumus (25) dan koefisien x^n pada $Q_n(x)$ adalah $A_k = f[x_1, x_2, x_3, \ldots, x_k]$, yang tidak lain adalah koefisien x^n pada polinomial interpolasi Newton (3), maka (25) identik dengan (3).

Jadi polinomial interpolasi Newton (3) menginterpolasikan $n+1$ titik berlainan sebagaimana dimaksudkan semula, yakni $\{(x_i, y_i) \mid i = 1, 2, 3, \ldots, n+1;\ x_i \neq x_j \ jika\ i \neq j\}$, dan dapat diperoleh recara rekursif dengan rumus (25), dengan polinomial interpolasi awal $Q_1(x) = y_1 + (x - x_1) f[x_1, x_2]$ dan $y_k = f(x_k)$, untuk $k = 1, 2, 3, \ldots, n+1$.

Dari uraian di atas telah jelas bahwa baik polinomial bentuk baku (1), polinomial Newton (3), maupun polinomial bentuk Lagrange (4) menginterpolasikan $n+1$ titik yang dimaksud semula. Jadi menurut Teorema 2, bahwa polinomial interpolasi tunggal**, ketiga polinomial tersebut identik**.

B. Galat pada Polinomial Interpolasi

Misalkan f adalah fungsi riil yang didenfinisikan pada suatu interval $I = [a, b]$, dan misalkan $P_n(x)$ adalah polinomial interpolasi berderajad kurang atau sama dengan n yang menginterpolasikan $n+1$ titik berlainan pada I, yakni

$$\{(x_i, f(x_i)) \mid x_i \in I,\ i = 1, 2, 3, \ldots, n+1;\ x_i \neq x_j \ jika\ i \neq j\}.$$

Galat interpolasi $e_n(x)$ pada polinomial interpolasi $P_n(x)$ diberikan oleh

$$e_n(x) = f(x) - P_n(x). \tag{26}$$

Sudah tentu dari persyaratan polinomial interpolasi kita tahu bahwa $e_n(x_i) = 0$ untuk $i = 1, 2, 3, \ldots, n+1$. Permasalahannya bagaimana mengetahui nilai galat $e_n(x)$ untuk sebarang $x \in I$.

Misalkan $\bar{x} \in I$ dan berlainan dengan $x_1, x_2, x_3, \ldots, x_{n+1}$. Apabila $P_{n+1}(x)$ adalah polinomial berderajad kurang atau sama dengan $(n+1)$ yang menginterpolasikan $x_1, x_2, x_3, \ldots, x_{n+1}, \bar{x}$, maka $P_{n+1}(\bar{x}) = f(\bar{x})$, sedangkan menurut (25),

$$P_{n+1}(x) = P_n(x) + f[x_1, x_2, \ldots, x_{n+1}, \bar{x}] \prod_{i=1}^{n+1} (x - x_i),$$

sehingga

$$f(\bar{x}) = P_{n+1}(\bar{x}) = P_n(\bar{x}) + f[x_1, x_2, \ldots, x_{n+1}, \bar{x}] \prod_{i=1}^{n+1} (\bar{x} - x_i).$$

Jadi,

$$e_n(\bar{x}) = f[x_1, x_2, \ldots, x_{n+1}, \bar{x}] \prod_{i=1}^{n+1} (\bar{x} - x_i), \quad \forall \bar{x} \in I, \ \bar{x} \neq x_1, x_2, \ldots, x_{n+1}. \tag{27}$$

Ruas kanan pada (27) memperlihatkan bahwa galat polinomial interpolasi "mirip suku terakhir" dalam polinomial interpolasi Newton.

Untuk menghitung galat (27) diperlukan informasi nilai $f(\bar{x})$, guna menghitung $f[x_1, x_2, \ldots, x_{n+1}, \bar{x}]$. Akan tetapi, seperti ditunjukkan pada Lemma 3, bahwa selisih terbagi Newton dapat dihitung menggunakan turunan. Teorema berikut merupakan generalisasi Lemma 3.

Teorema 4 (Hubungan Selisih Terbagi Newton dan Turunan)

Misalkan f adalah fungsi riil yang kontinyu dan mempunyai turunan sampai ke-$n+1$ yang kontinyu pada suatu interval $I = [a,b]$. Jika $x_1, x_2, x_3, \ldots, x_{n+1}$ adalah *$n+1$ titik berlainan pada I*, maka terdapat $x \in (a,b)$ sedemikian hingga

$$f[x_1, x_2, x_3, \ldots, x_{n+1}] = \frac{f^{(n)}(x)}{n!}. \tag{28}$$

Bukti:

Untuk kasus $n = 1$ dan $n = 2$ kebenaran (28) sudah ditunjukkan pada Lemma 3. Untuk membuktikan kasus umum, misalkan $P_n(x)$ adalah polinomial berderajad kurang atau sama dengan n yang menginterpolasikan $x_1, x_2, x_3, \ldots, x_{n+1}$. Selanjutnya, didefinisikan fungsi g,

$$g(t) = f(t) - P_n(t) - C \prod_{i=1}^{n+1} (t - x_i), \ \forall t \in I,$$

dengan C adalah suatu konstanta yang hendak dicari. Mengingat $f(x_i) = P_n(x_i)$, maka $g(x_i) = 0$ untuk $i = 1, 2, 3, \ldots, n+1$. Misalkan C dipilih sedemikian hingga $g(x) = 0$ untuk suatu $x \in I$ yang berlainan dengan $n+1$ titik yang diketahui, yakni

$$C = \frac{f(x) - P_n(x)}{\prod_{i=1}^{n+1} (x - x_i)}. \tag{29}$$

Jadi $g(t) = 0$ untuk $t = x_1, x_2, \ldots, x_{n+1}, x$. Dari definisinya dapat diketahui bahwa fungsi g kontinyu dan mempunyai turunan sampai ke-$n+1$ yang kontinyu, karena f demikian dan

$P_n(x)$ adalah polinomial berderajad kurang atau sama dengan n. Jadi kita dapat menggunakan teorema Rolle secara berulang pada $g, g', g'', \ldots, g^{(n+1)}$. Akhirnya, dapat ditemukan suatu $x \in (a,b)$ yang memenuhi

$$0 = g^{(n+1)}(x) = f^{(n+1)}(x) - P_n^{(n+1)}(x) - C\frac{d^{n+1)}}{dt^{(n+1)}}\prod_{i=1}^{n+1}(t - x_i)\Big|_{t=x}.$$

Selanjutnya, karena $P_n(t)$ adalah polinomial berderajad kurang atau sama dengan n, maka turunan ke-($n+1$)-nya pastilah nol. Demikian pula, $\prod_{i=1}^{n+1}(t - x_i)$ merupakan suatu polinomial berderajad tepat $n+1$ de-ngan koefisien t^{n+1} adalah 1, maka turunan ke-($n+1$)-nya adalah $(n+1)!$. Dengan demikian diperoleh

$$0 = g^{(n+1)}(x) = f^{(n+1)}(x) - 0 - (n+1)!C \text{ atau } C = \frac{f^{(n+1)}(x)}{(n+1)!}.$$

Apabila hasil terakhir dimasukkan ke dalam (29) maka diperoleh

$$\frac{f^{(n+1)}(x)}{(n+1)!} = \frac{f(x) - P_n(x)}{\prod_{i=1}^{n+1}(x - x_i)}$$

atau

$$e_n(x) = \frac{f^{(n+1)}(x)}{(n+1)!}\prod_{i=1}^{n+1}(x - x_i), \quad \forall x \in I, x \neq x_i. \tag{30}$$

Jika rumus galat (30) digabung dengan rumus galat (27) maka diperoleh

$$\frac{f^{(n+1)}(x)}{(n+1)!}\prod_{i=1}^{n+1}(x - x_i) = f[x_1, x_2, \ldots, x_{n+1}, x]\prod_{i=1}^{n+1}(x - x_i), \quad \forall x \in I,\ x \neq x_1, x_2, \ldots, x_{n+1}$$

atau

$$f[x_1, x_2, \ldots, x_{n+1}, x] = \frac{f^{(n+1)}(x)}{(n+1)!}.$$

Dengan menuliskan $x = x_{n+1}$ dan rumus terakhir diterapkan pada $n+1$ titik pertama, maka diperoleh (28). ∎

Dari bukti di atas ternyata telah diperoleh rumus lain untuk menghitung galat polinomial interpolasi, yakni persamaan (30). Dari rumus (30) dapat dihitung batas-batas galat polinomial interpolasi.

C. Perhitungan dalam Polinomial Interpolasi

Perhitungan yang berkaitan dengan polinomial-polinomial interpolasi meliputi dua macam, yakni: (1) perhitungan untuk membentuk polinomial interpolasi, dan (2) perhitungan nilai-nilai polinomial tersebut.

1. Polinomial Bentuk Baku

Untuk polinomial bentuk baku (1), perhitungan koefisien dilakukan dengan menyelesaikan suatu sistem persamaan linier khusus (2), yang mempunyai matriks koefisien berupa matriks **Vandermonde**. Pembentukan matriks koefisien sendiri memerlukan $(n-1)(n)(n+1)/2$ perkalian. Selanjutnya, untuk menyelesaikan sistem persamaan liniear dengan ukuran matriks koefisien $n \times n$ menggunakan metode eliminasi Gauss dan penyulihan mundur diperlukan $n(n-1)(2n+5)/6$ operasi penjumlahan/pengurangan dan $n(n^2+3n-1)/3$ operasi perkalian/pembagian (Atkinson, 2002: 254-255). Setelah koefisien-koefisien polinomial interpolasi bentuk baku diperoleh, perhitungan nilai-nilai polinomial tersebut dapat dilakukan secara efisien menggunakan algoritma perkalian tersarang, yang hanya memerlukan n penjumlahan/ pengurangan dan n perkalian.

Polinomial bentuk baku tidak dapat digunakan untuk membentuk polinomial interpolasi baru apabila titik interpolasinya diperbanyak. Jadi, pemakaian polinomial interpolasi bentuk baku tidak efisien apabila digunakan untuk memperoleh polinomial-polinomial interpolasi dengan cacah titik interpolasi semakin bertambah banyak.

2. Polinomial Newton (Selisih Terbagi)

Pembentukan polinomial interpolasi Newton (3) juga memerlukan perhitungan koefisien-koefisien, yang tidak lain adalah selisih-selisih terbagi Newton. Polinomial Newton (3) dapat dituliskan dalam bentuk

$$Q_n(x) = D_1 + \sum_{k=1}^{n} D_{k+1} \prod_{i=1}^{k} (x - x_i). \tag{31}$$

Perhitunan nilai-nilai D dilakukan sebagai berikut:

1). Inisialisasi: $D_k = y_k = f(x_k), \quad untuk \ k = 1, 2, 3, \ldots, n+1$

2). Untuk $k = 2, 3, 4, \ldots, n+1$, lakukan
   Untuk $j = n+1, n, n-1, \ldots, k$, hitung ulang
   $D_j = (D_j - D_{j-1})/(x_j - x_{j-k+1})$. (32)

Pada akhir perhitungan akan diperoleh nilai-nilai $D_1 = y_1$ dan $D_k = f[x_1, x_2, \ldots, x_{k+1}]$, untuk $k = 2, 3, 4, \ldots, n+1$. Perhitungan nilai-nilai D tersebut memerlukan operasi hitung sebanyak $n(n+1)$ penjumlahan/pengurangan dan $n(n+1)/2$ perkalian/pembagian. Setelah nilai-nilai D tersebut diperoleh, perhitungan nilai-nilai polinomial Newton dapat dilakukan secara efisien menggunakan algoritma perkalian tersarang, yang hanya memerlukan $2n$ penjumlahan/ pengurangan dan n perkalian.

Oleh karena sifatnya yang rekursif, polinomial interpolasi Newton sangat efisien digunakan untuk membentuk polinomial-polinomial interpolasi dengan cacah titik interpolasi semakin bertambah. Pembentukan polinomial interpolasi baru yang berderajad lebih tinggi dengan adanya penambahan titik interpolasi baru dapat dilakukan dengan menggunakan polinomial-polinomial berlerajad lebih rendah yang sudah diperoleh.

3. Polinomial Interpolasi Lagrange

Meskipun tidak dilakukan secara eksplisit, pembentukan polinomial interpolasi Lagrange (4) juga memerlukan perhitungan koefisien-koefisien, yakni perhitungan penyebut pada $L_k(x)$

memerlukan n penjumlahan/pengurangan dan $n-1$ perkalian. Selanjutnya, untuk menghitung nilai $L_k(x)$ diperlukan n penjumlahan/pengurangan dan n perkalian/pembagian. Setelah nilai-nilai polinomial $L_k(x)$ dihitung, akhirnya untuk menghitung nilai polinomial interpolasi Lagrange $R_n(x)$ diperlukan $n+1$ perkalian dan n penjumlahan. Jadi, untuk menghitung nilai-nilai polinomial interpolasi Lagrange $R_n(x)$ dari data interpolasi diperlukan

$$(n+1)\{n+n\}+n=2n^2+3n \quad penjumlahan/pengurangan$$

dan

$$(n+1)\{n-1+n\}+(n+1)=2n^2+2n \quad perkalian/pembagian.$$

Sebagai alternatif lain untuk membentuk dan menghitung polinomial interpolasi Lagrange $R_n(x)$ dapat dilakukan dengan cara menyatakan $R_n(x)$ sebagai

$$R_n(x)=\sum_{k=1}^{n+1} l_k \prod_{i=1,i\neq k}^{n+1}(x-x_i) \quad dengan \; l_k=\frac{y_k}{\prod_{i=1,i\neq k}^{n+1}(x_k-x_i)}. \tag{33}$$

Dari (33) jelas terlihat bahwa perhitungan nilai koefisien-koefisien l dapat dilakukan dengan menggunakan $(n+1)n$ penjumlahan/pengurangan dan $(n+1)n$ perkalian/pembagian. Setelah nilai-nilai l diperoleh, untuk menghitung nilai $R_n(x)$ diperlukan $(n+1)n+n$ penjumlah-an/pengurangan dan $(n+1)n$ perkalian, sehingga total operasi yang diperlukan untuk membentuk dan menghitung nilai $R_n(x)$ adalah $4(n+1)n+n=4n^2+5n$ operasi, sama seperti hasil analisis pada paragraf sebelumnya.

D. Implementasi Polinomial Interpolasi dengan Matlab

Seperti pada uraian di atas, perhitungan dalam interpolasi polinomial meliputi dua macam, yakni perhitungan koefisien-koefisien dan perhitungan nilai-nilai polinomial interpolasi (sebagai hampiran nilai fungsi yang diketahui data atau rumusnya). Dalam implementasi, prosedur untuk menghitung koefisien-koefisien polinomial interpolasi dan prosedur untuk menghitung nilai polinomial perlu dipisahkan, sehingga perhitungan lebih efisien – tidak setiap hendak menghitung nilai polinomial harus menghitung koefisien-koefisiennya.

Pada polinomial bentuk baku, perhitungan koefisien diperoleh dengan menyelesaikan sebuah SPL yang mempunyai matriks koefisien khusus, yakni matriks Vandermonde. Pada Matlab terdapat fungsi **vander** untuk menghasilkan matriks Vandermonde dari data yang diberikan. Akan tetapi, karena di dalam Matlab polinomial ditulis dari suku berpangkat tertinggi, maka kita perlu mencerminkan matriks/vektor-vektor pada Matlab. Hal ini dapat dilakukan dengan perintah **fliplr** (untuk vektor baris) atau **flipud** (untuk vektor kolom). Perhitungan polinomial de-ngan perkalian tersarang pada Matlab juga sudah diimplementasikan dalam fungsi **polyval**. Dengan demikian ketiga fungsi Matlab tersebut dapat digunakan dalam perhitungan polinomial interpolasi bentuk baku.

Program (fungsi) Matlab untuk menghitung koefisien-koefisien polinomial interpolasi bentuk umu adalah **interpolum.m**, disajikan pada **Lampiran A**. Pemakaian fungsi tersebut dilakukan dengan menuliskan perintah di bawah ini pada baris perintah Matlab.

```
a=interpolum(x,y)
```

Masukan (argumen) untuk fungsi **interpolum** adalah pasangan vektor **x** dan **y**, keduanya harus berukuran sama dan elemen-elemen vektor **x** tidak ada yang sama. Hasilnya berupa vektor **a** yang elemen-elemennya merupakan koefisien-koefisien polinomial

$$a_1 z^n + a_2 z^{n-1} + \ldots + a_n z + a_{n+1}.$$

Selanjutnya, untuk menghitung nilai-nilai polinomial interpolasi dapat digunakan perintah (fungsi) Matlab **polyval** dengan masukan vektor **a,** hasil fungsi **interpolum,** dan **z**, misalnya

```
y1=polyval(a,z).
```

Program Matlab untuk menghitung koefisien-koefisien dan nilai-nilai polinomial interpolasi Newton (metode selisih terbagi) adalah **interpolistn.m** dan **nilaipolin.m**, disajikan pada **Lampiran A.** Kedua fungsi tersebut didasarkan pada algoritma (32) dan perkalian tersarang untuk rumus (31). Perintah Matlab

```
D=interpolistn(x,y)
```

menghasilkan vektor **D** yang berisi selisih-selisih terbagi Newton. Selanjutnya, untuk menghitung nilai-nilai polinomial interpolasi dapat digunakan perintah (fungsi **nilaipolin**) dengan cara menuliskan, misalnya

```
y2=nilaipolin(D,z).
```

Yang terakhir, program Matlab untuk menghitung koefisien-koefisien dan nilai-nilai polinomial interpolasi Lagrange yang didasarkan pada (33) adalah **interpolag.m** dan **nilaipolag.m**, dapat dilihat pada **Lampiran A**. Format umum pemakaian kedua fungsi tersebut adalah sebagai berikut

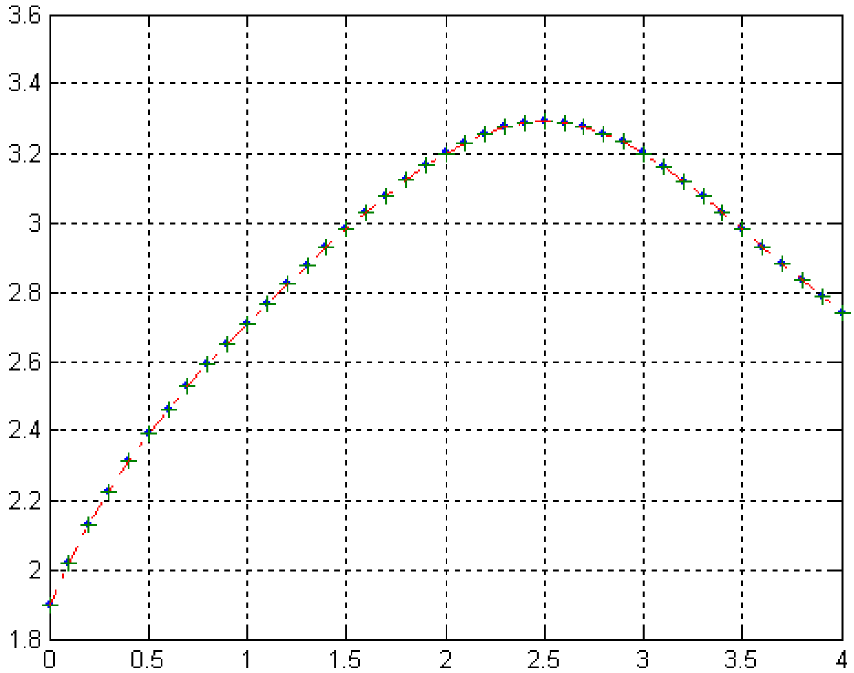
```
lambda=interpolag(x,y)
y3=nilaipolag(lambda,z).
```

Berikut adalah dua contoh data untuk interpolasi yang diselesaikan dengan ketiga polinomial interpolasi di atas. Pada contoh pertama, data fungsi tidak diketahui rumusnya sedangkan pada contoh kedua, rumus fungsi diketahui.

```
>> x =[0      0.5     1.0     1.5     2.0    3.0     3.5     4.0];
>> y =[1.90  2.39    2.71    2.98    3.20   3.20    2.98    2.74];
>> a=interpolum(x,y)'
a =
  -0.0024    0.0312   -0.1385    0.2115    0.0858   -0.6348    1.2572    1.9000
>> polyval(a,2.5)
ans =
   3.2907
>> D=interpolistn(x,y)
D =
   1.9000    0.9800   -0.3400    0.1600   -0.0800    0.0138    0.0034   -0.0024
>> nilaipolin(D,x,2.5)
ans =
   3.2907
>> l=interpolag(x,y)
l =
  -0.0302    0.2428   -0.7227    1.0596   -0.7111    0.2844   -0.1514    0.0261
>> nilaipolag(l,x,2.5)
ans =
   3.2907
```

Dari hasil perhitungan di atas terlihat bahwa ketiga polinomial interpolasi mempunyai nilai yang sama di x=2.5. Hal ini sesuai dengan hasil analisis teoritis sebelumnya bahwa ketiga polinomial interpolasi adalah identik. Fakta ini akan lebih jelas jika kurva ketiga polinomial digambar, ketiganya akan tampak berimpit, seperti pada Gambar 1, yang dihasilkan dari perintah-perintah sebagai berikut.

```
>> t=0:.1:4;
>> y1=polyval(a,t);
>> y2=nilaipolin(D,x,t);
>> y3=nilaipolag(l,x,t);
>> plot(t,y1,'.',t,y2,'+',t,y3,'-.')
>> grid on
```



Gambar 1 Ketiga kurva polinomial interpolasi berdasarkan data (0, 1.90), (0.5, 2.39), (1.0, 2.71), (1.5, 2.98), (2.0, 3.20), (3.0, 3.20), (3.5, 2.98), (4.0, 2.74)

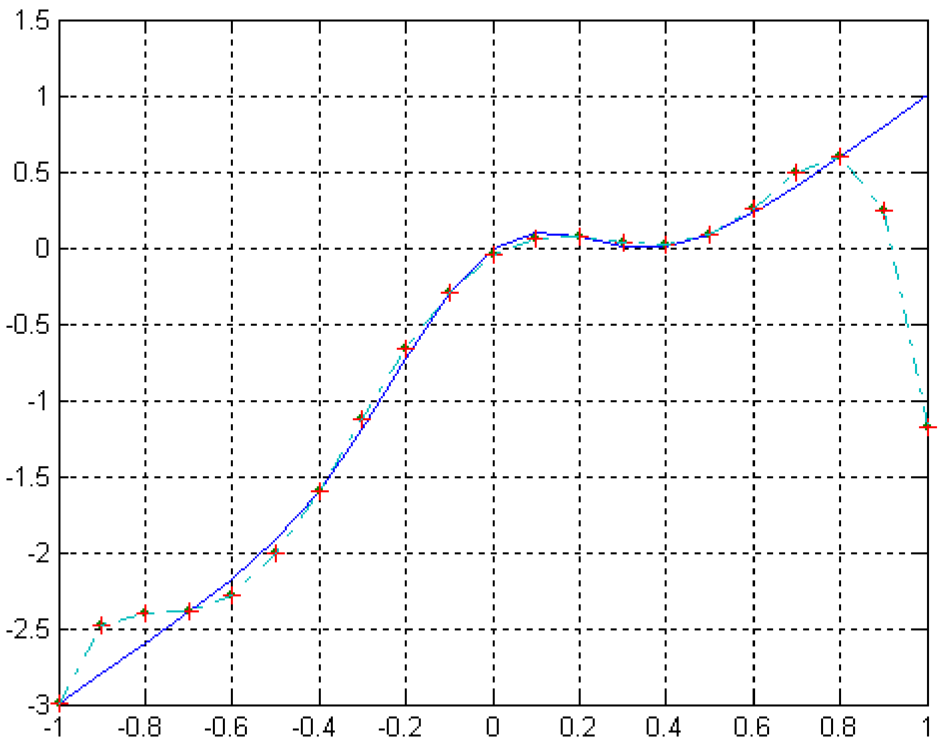
Eksperimen kedua dilakukan terhadap fungsi $f(x) = 2x - 1 + e^{-10x^2}$ dengan data interpolasi nilai-nilai f di $x = -1, -0.7, -0.4, -0.1, 0.2, 0.5, 0.8$. Berikut adalah perintah-perintah Matlab yang digunakan untuk menghasilkan kurva fungsi tersebut dan ketiga polinomial interpolasi.

```
>> f=inline('2*x-1+exp(-10*x.^2)')
f =
     Inline function:
     f(x) = 2*x-1+exp(-10*x.^2)
>> x=-1:.3:1
x =
   -1.0000   -0.7000   -0.4000   -0.1000    0.2000    0.5000    0.8000
>> y=f(x)
y =
   -3.0000   -2.3926   -1.5981   -0.2952    0.0703    0.0821    0.6017
>> A=interpolum(x,y)'
```

```
A =
  -10.5719   -2.8363   15.3658    1.9793   -6.8310    1.7705   -0.0494
>> D=interpolistn(x,y)
D =
   -3.0000    2.0247    1.0392    1.9842   -9.0915   13.0215  -10.5719
>> L=interpolag(x,y)
L =
   -5.7155   27.3497  -45.6705   11.2469    2.0096   -0.9383    1.1463
>> T=-1:.1:1;
>> f0=f(T);
>> f1=polyval(A,T);
>> f2=nilaipolin(D,t,T);
>> f3=nilaipolag(L,x,T);
>> f2=nilaipolin(D,x,T);
>> plot(T,f0,'-',T,f1,'.',T,f2,'+',T,f3,'-.')
>> grid on
```

Hasil perintah-perintah di atas diperlihatkan pada Gambar 2. Dari gambar tersebut dapat diketahui bahwa:

1. Ketiga kurva polinomial interpolasi tampak berimpit
2. Ketiga polinomial interpolasi cocok dengan fungsi aslinya pada **bagian tengah** interval di mana interpolasi dilakukan, sedangkan di **bagian ujung** interval polinomial interpolasi kurang cocok dengan fungsi aslinya.



Gambar 2 Kurva fungsi $f(x) = 2x - 1 + e^{-10x^2}$ dan ketiga polinomial interpolasi berdasarkan nilai-nilai f di $x = -1, -0.7, -0.4, -0.1, 0.2, 0.5, 0.8$

Hasil (pengamatan) nomor 2 di atas sebenarnya tidak terlepas dari perilaku polinomial pada fungsi galat (30), yakni $(x - x_1)(x - x_2)\ldots(x - x_{n+1})$. Untuk sub-subinterval yang berjarak sama dan nilai-nilai n yang cukup besar, misalnya $n \geq 5$, nilai-nilai polinomial ini berubah naik turun secara cepat pada interval interpolasi. Untuk titik-titik interpolasi yang berjarak sama, perubahan di ujung-ujung interval jauh lebih besar daripada perubahan di bagian tengah interval interpolasi (Atkinson, 2002:129). Hal ini juga tampak pada analisis analitik dengan bantuan Maple untuk kasus kedua di atas, yang disajikan pada **Lampiran B**. Semakin besar nilai n, semakin besar perubahan naik-turun nilai polinomial pada galat tersebut.

Dengan demikian, rumus galat (30) sebenarnya memberikan peringatan bahwa sesungguhnya galat pada polinomial interpolasi nampaknya akan lebih kecil pada bagian tengah interval interpolasi. Jadi, agar diperoleh polinomial interpolasi yang cukup akurat untuk suatu fungsi yang diketahui rumusnya, perlu ditentukan simpul-simpul interpolasi $x_1, x_2, x_3, \ldots, x_{n+1}$ yang **meminimumkan maksimum** $(x - x_1)(x - x_2)\ldots(x - x_{n+1})$. Pendekatan ini yang sering disebut pendekatan **near-minimax**, yang sudah di luar jangkauan penelitian ini.

Bab V
Kesimpulan dan Saran

A. Simpulan

Berdasarkan hasil analisis teoritik dan komputasi numerik pada bab sebelumnya, dapat disimpulkan beberapa hal yang berkaitan dengan rumusan masalah pada bab I sebagai berikut.

1. Telah ditunjukkan bahwa polinomial interpolasi Newton dan polinomial interpolasi Lagrange indentik dengan polinomial interpolasi bentuk baku.
2. Hasil analisis perhitungan untuk memperoleh koefisien-koefisien dan perhitungan nilai-nilai polinomial interplasi dapat dirangkum pada Tabel 1. Dari tabel tersebut dapat diketahui bahwa polinomial Newton (metode selisih terbagi Newton) merupakan polinomial interpolasi yang paling mudah dibentuk dan paling efisien dihitung, dibanding dengan kedua polinomial interpolasi yang lain.

Tabel 1 Rangkuman Cacah Operasi pada Perhitungan Polinomial Interpolasi

Polinomial Interpolasi	Cacah operasi hitung yang diperlukan		
	Perhitungan koefisien	Perhitungan nilai polinomial	Total
1. Bentuk umum: $P_n(x)=a_1+a_2x+a_3x^2+\ldots+a_{n+1}x^n$ Nilai koefisien-koefisien a diperoleh dengan menyelesaikan (2)	$(n-1)(n)(n+1)/2$ $n(n-1)(2n+5)/6$ $n(n^2+3n-1)/3$	$2n$	$\frac{7n^3+7n^2+2n}{6}$
2. Polinomial Newton (selisih terbagi): $Q_n(x)=D_1+\sum_{k=1}^{n} D_{k+1}\prod_{i=1}^{k}(x-x_i)$ D_k dihitung menggunakan (32)	$n(n+1)$ $n(n+1)/2$	$2n$	$\frac{3n^2+7n}{2}$
3. Polinomial Lagrange: $R_n(x)=\sum_{k=1}^{n+1} l_k \prod_{i=1,i\neq k}^{n+1}(x-x_i)$ l_k dihitung menggunakan (33)	$(n+1)n$ $(n+1)n$	$(n+1)n+n$ $(n+1)n$	$4n^2+5n$

3. Besar galat hampiran nilai fungsi dengan menggunakan ketiga polinomial interpolasi tersebut adalah sama, karena ketiga polinomial identik, dan dinyatakan oleh rumus (30). Galat ini juga merupakan polinomial yang perilakunya tergantung pada titik-titik interpolasi.

Selain kesimpulan sebagai jawaban rumusan masalah tersebut, dapat pula disimpulkan sebagai berikut:

- Polinomial interpolasi bentuk baku dan polinomial Lagrange tidak praktis digunakan dalam interpolasi dengan cacah titik interpolasi semakin bertambah. Meskipun demikian polinomial interpolasi Lagrange dapat digunakan untuk menunjukkan keberadaan polinomial interpolasi.
- Karena dibentuk secara rekursif, polinomial interpolasi Newton sangat efisien jika digunakan untuk pembentukan dan perhitungan polinomial-polinomial interpolasi dengan cacah titik interpolasi semakin bertambah.

B. Saran

1. Analisis ketunggalan polinomial interpolasi dapat dilanjutkan apakah hasilnya tetap konsisten untuk interpolasi pada bidang (fungsi dua variabel).
2. Untuk keperluan praktis interpolasi dengan polinomial berderajad tinggi dengan titik-titik interpolasi berjarak sama tidak dianjurkan, karena sifat fungsi galatnya yang fluktuatif. Sebagai alternatif dapat dicari polinomial interpolasi dengan pendekatan **minimax** atau **near-minimx** (Tentang hal ini perlu diselidiki lebih lanjut).

Daftar Pustaka

Atkinson, Kendal (1993). *Elementary Numerical Analysis*. second edition. John Wiley & Sons, Singapore.

Atkinson, Kendal (2002). *Elementary Numerical Analysis*. revision edition (Chapter 1 – 8). download dari http://www.cs.uiowa.edu/~atkinson

Atkinson, Kendal (1993). *Elementar Numerical Analysis*. second edition. John Wiley & Sons, Singapore.

Borse, G.J (1997). *Numerical Methods with MATLAB, A Resource for Scientiests and Engineers*. PWS Publishing Company, Boston.

Conte, Samuel D. & Carl de Boor (1981). *Elementary Numerical Analysis, An Algorithmic Approach*. 3rd edition. McGraw-Hill Book Company, Singapore

Gerald, Curtis F. & Patrick O. Wheatly (1994). *Applied Numerical Analysis*. 5th edition. Addison-Wisley Pub. Co., Singapore

Jacques, Ian & Colin Judd (1987). *Numerical Analysis*. Chapman and Hall, New York.

Mathews, John H (1992). *Numerical Methods for Mathematics, Science, and Engineering*. second edition. Prentice-Hall, Inc. Englewood Cliffs, New York.

Scheid, Francis (1989). *Schaum's Outline Series Theory and Problems of Numerical Analysis*. 2/ed. McGraw-Hill Book Company, Singapore.

Volkov, E. A (1990). *Numerical Methods*. Hemisphere Publishing Company, New York.

Lampiran

A. Fungsi-fungsi Matlab

```
function  a= interpolum(x,y)
% menghitung nilai-nilai a_1, a_2, ..., a_{n+1}
% yang memenuhi y=a_1+a_2x+a_3x.^2+...+a_{n+1}x.^n
% x dan y adalah vektor kolom/baris dengan cacah elemen sama (=n+1)
% elemen-elemen pada vektor x TIDAK boleh ada yang sama (kontrolnya?)
% Kontrol kebenaran masukan:
[m,n] = size(y);
if (nargin~=2)>0 | find((size(x)~=[m,n]))>0 | (min(m,n)~=1)>0,
   error('Salah menggunakan fungsi!'), end
% Perhitungan koefisien:
V=fliplr(vander(x));
if m==1, a=V\y'; else a=V\y; end  %cek apakah y berupa vektor kolom/baris
% vektor a yang didapat di atas perlu dicerminkan agar tidak kebalik
% jika dipakai untuk menghitung nilai polinomial di z dengan fungsi 'polyval(a,z)'
% karena Matlab mengasumsikan koefisien polinomial dari suku berpangkat tertinggi
a=flipud(a);
```

```
function D=interpolistn(x,y);
% menghitung nilai-nilai selisih terbagi Newton D
% pada koefisien-koefisien polinomial interpolasi
% Newton : Q(x)=D1+ D2(x-x1)+D3(x-x1)(x-x2)+...+Dn+1(x-x1)(x--x2)...(x-xn)
% dengan Dk=y[x1,x2,...,x(k+1)]
% elemen-elemen pada vektor x TIDAK boleh ada yang sama (kontrolnya?)
% Kontrol kebenaran masukan:
[m,n] = size(y);
if (nargin~=2)>0 | find((size(x)~=[m,n]))>0 | (min(m,n)~=1)>0,
   error('Salah menggunakan fungsi!'), end
D=y; n=length(D);
for k=2:n
   for j=n:-1:k,
      D(j)=(D(j)-D(j-1))/(x(j)-x(j-k+1));
   end
end
```

```
function Q=nilaipolin(D,x,z)
% menghitung nilai polinomial interpolasi Newton pada z
% vektor x harus sama dengan yang digunakan
% untuk menghitung D=interpolistn(x,y)
% jika tidak demikian, perhitungan tak berarti!!!
% elemen-elemen pada vektor x TIDAK boleh ada yang sama (kontrolnya?)
n=length(D);
Q=D(n)*ones(size(z));
for j=n-1:-1:1
   Q=D(j)+Q.*(z-x(j));
end
```

```
function lambda=interpolag(x,y)
% menghitung koefisien-koefisien
% polinomial interpolasi Lagrange
% elemen-elemen pada vektor x TIDAK boleh ada yang sama (kontrolnya?)
% Kontrol kebenaran masukan:
[m,n] = size(y);
if (nargin~=2)>0 | find((size(x)~=[m,n]))>0 | (min(m,n)~=1)>0,
```

```
    error('Salah menggunakan fungsi!'), end
n=length(x);
for k=1:n
    p=x(k)-x;
    lambda(k)=y(k)/prod(p(find(p)));
end
```

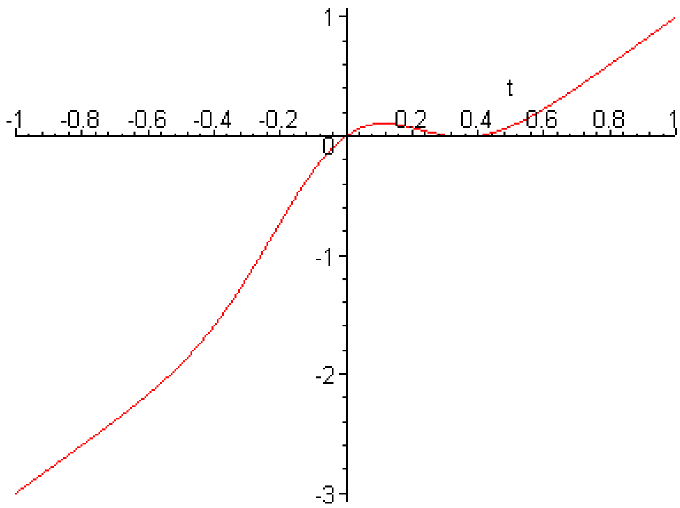
```
function R=nilaipolag(lambda,x,z)
% menghitung nilai polinomial interpolasi Lagrange pada z
% vektor x harus sama dengan yang digunakan
% untuk menghitung lambda=interpolag(x,y)
% jika tidak demikian, perhitungan tak berarti!!!
% elemen-elemen pada vektor x TIDAK boleh ada yang sama (kontrolnya?)
[m,n]=size(z);
R=zeros(m,n);
q=length(x);
for i=1:m
    for j=1:n
        for k=1:q
            L=lambda(k);
            for r=1:q
                if r~=k L=L*(z(i,j)-x(r));
                end
            end
            R(i,j)=R(i,j)+L;
        end
    end
end
```

B. Perhitungan Analitik dengan Maple untuk Contoh Komputasi Kedua

```
> f:= proc(x)
        2*x-1+exp(-10*x^2);
        end;
> f(t);
```

$$2\,t-1+\mathbf{e}^{(-10\,t^2)}$$

```
> plot([f(t)],t=-1..1);
```




```
> df7:=proc(x)
        local t;
        eval(diff(f(t),t$7),t=x)/7!; end;
> factor(df7(t));
```

$$-\frac{10000}{63}\, t\, \mathbf{e}^{(-10\, t^2)}\, (-21 + 420\, t^2 - 1680\, t^4 + 1600\, t^6)$$

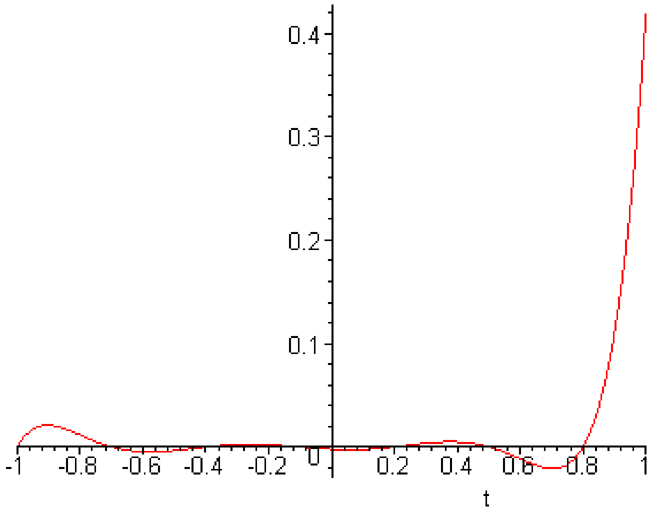
```
> with(linalg):x:=[-1, -.7, -.4, -.1, .2, .5, .8];
```

$$x := [-1, -.7, -.4, -.1, .2, .5, .8]$$

```
> p:=proc(t,x)
        local k,p;
        p:=1;
        for k from 1 to nops(x) do  p:=p*(t-x[k]); end do; end;
> p(t,x); # polinomial galat polinomial interpolasi berderajad 6 untuk f(t)
```

$$(t+1)\,(t+.7)\,(t+.4)\,(t+.1)\,(t-.2)\,(t-.5)\,(t-.8)$$

```
> plot(p(t,x),t=-1..1); # terlihat galat interpolasi relatif kecil di bagian tengah interval interpolasi
```




```
> maximize(df7(t),t=-1..1):  M:=evalf(%);
```

$$M := 252.2775786$$

```
> minimize(df7(t),t=-1..1):  m:=evalf(%);
```

$$m := -252.2775786$$