

Cepat Mahir Algoritma dalam C

I Putu Gede Darmawan
ipgd_bali@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarakan secara bebas untuk tujuan bukan komersial (*nonprofit*), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.

Analisa Numerik: Menghitung Nilai Cosinus

Tahukah anda , bagaimana caranya komputer menghitung nilai dari $\cos(30)$? . Mungkin kalau hanya sekedar menghitung nilai dari $\cos(30^\circ)$ kita tidak membutuhkan bantuan komputer. Nah bagaimana halnya dengan $\cos(30.5^\circ)$.Komputer tentu saja tidak menyimpan nilai – nilai tersebut dalam sebuah tabel.Selain memakan tempat , juga terbatas hanya pada sudut yang tertentu yang disimpan saja. Dalam beberapa kasus pemrograman, penyimpanan nilai dalam tabel dapat dipertimbangkan terutama dalam program – program yang berjalan secara *real time* dimana dengan bantuan tabel dapat mengurangi beban dari proses komputasi . Pada kesempatan kali ini saya akan membahas salah satu algoritma yang dapat digunakan sebagai alat membantu mendapatkan nilai dari beberapa fungsi dasar matematika . Untuk itu ada baiknya kita menengok sejenak pembahasan mengenai deret khususnya **Deret Taylor** dan **Deret Maclaurin** yang terdapat pada mata kuliah Kalkulus.

Deret Taylor dan Deret Maclaurin

Bentuk Umum Deret Taylor :

$$f(x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(a)}{i!} x^k$$

Nilai a adalah sebuah constanta , Deret MacLaurin merupakan Deret Taylor khusus yang didapat dengan cara mensubstitusi nilai a dengan 0.

Dengan bantuan rumus diatas maka menghitung nilai cos bukan merupakan hal yang misteri lagi . Persamaan diatas berlaku tidak hanya untuk menghitung nilai cos saja tapi juga dapat menghitung beberapa fungsi matematika lainnya , seperti fungsi fungsi yang berhubungan dengan trigonometri , exponensial , logaritma , Hyperbolic dan lainnya .

Oke saya rasa kita langsung menuju contoh saja . Kali saya akan membuat fungsi untuk menghitung nilai cos dengan menggunakan deret MacLaurin.

Pertama – tama kita mencari dulu turunan masing masing tingkat dari cos(x) serta hasilnya bila x = 0 (Maclaurin)

Turunan	Fungsi	x = 0
0	cos(x)	1
1	-sin(x)	0
2	-cos(x)	-1
3	sin(x)	0
4	cos(x)	1

dan seterusnya

jika diterjemahkan dalam bahasa C maka aka menjadi :

```
int turunan(int n)
{
    int hasil;
    n = n % 4;
    switch(n)
    {
        case 0 : hasil = 1;break;
        case 1 : hasil = 0;break;
        case 2 : hasil = -1;break;
        case 3 : hasil = 0;break;
    }
}
```

Fungsi turunan diatas ialah untuk menghitung nilai turunan tingkat ke - n. Melihat dari tabel bahwa pada turunan ke 4 sama dengan turunan ke 0 maka bisa disimpulkan:

n = n % 4;

Setelah kita tahu turunannya maka dapat disusun algoritma global untuk penyelesaian semua masalah yang menggunakan deret Maclaurin dengan input X.

1. $n = 0$ // menyatakan turunan ke $-n$
2. pembilang = 1 // $X^0 = 1$
3. penyebut = 1 // $0! = 1$
4. hasil = 0 // hasil keseluruhan
5. suku = turunan(n) * pembilang / penyebut // hitung nilai dari suku ke n
6. hasil = hasil + suku
7. Tambahkan nilai n dengan 1
8. pembilang = pembilang * X // untuk mendapatkan $X^{k+1} = X^k * X$
9. penyebut = penyebut * n // untuk mendapatkan faktorial ke n
10. Ulangi terus langkah ke 4 – 8 sampai *keadaan tertentu*

Dalam perhitungan menggunakan deret MacLaurin dimana fungsi diturunkan terus sampai tak hingga, maka diperlukan adanya sebuah pemotongan pada *keadaan tertentu* yang menyatakan kapan perhitungan tersebut harus berhenti. *Keadaan tertentu* yang dipakai biasanya ditentukan oleh nilai dari suku. Pada contoh yang saya berikan, jika nilai suku sudah lebih kecil dari galat maka tidak perlu dilakukan perhitungan lagi alias keluar dari looping.

Berikut listing program lengkap dari contoh diatas :

```
#include <stdio.h>

#define abs(x) ((x >= 0)?x:-x)

int turunan(int n)
{
    int hasil;
    n = n % 4;
    switch(n)
    {
        case 0 : hasil = 1;break;
        case 1 : hasil = 0;break;
        case 2 : hasil = -1;break;
        case 3 : hasil = 0;break;
    }
    return hasil;
}
```

```
double cos(double x)
{
    double pembilang,penyebut,hasil,suku;
    double galat;
    double n;
    pembilang = penyebut = 1;
    galat = 0.00001;
    hasil = 0.0;
    n = 0;
    do
    {
        if (turunan(n) != 0) // untuk mengurangi perkalian dengan 0
        {
            suku = turunan(n) * pembilang / penyebut;
            hasil = hasil + suku;
        }
        n++;
        pembilang = pembilang * x; // pangkat
        penyebut = penyebut * n; // faktorial
    }while(abs(suku) >= galat);
    return hasil;
}

void main(void)
{
    const double PI = 3.14159265;

    printf("%8.4f\n",cos(0));
    printf("%8.4f\n",cos(PI/6));

    printf("%8.4f\n",cos(PI/4));
    printf("%8.4f\n",cos(PI/2));
    printf("%8.4f\n",cos(PI));
}
```

Hasil :

```
1.0000
0.8660
0.7071
0.0000
-1.0000
```

Nah coba dibandingkan hasil yang anda dapat dari program ini dengan kalkulator .
Hasilnya pasti sama.