

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 disebarkan 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.

Membilang dalam Bahasa Indonesia

Dalam aplikasi yang berhubungan dengan uang , kadang – kadang diperlukan pernyataan nilai nominal tidak hanya dengan angka , tapi juga diperlukan pernyataan dalam bentuk terbilanganya. Tentu saja jika dibandingkan dengan pernyataan dengan angka , pernyataan dengan membilang lebih banyak membutuhkan ruang . Tapi disamping itu pernyataan dengan membilang bisa meminimkan jumlah kesalahan yang terjadi karena faktor kesalahan pembacaan .

Untuk itu dalam kesempatan ini kita akan membuat sebuah fungsi yang dapat digunakan untuk merubah nilai menjadi bentuk terbilanganya. Di internet , tentunya sudah sangat banyak sekali source code yang di publish untuk keperluan ini , bahkan dengan listing yang cukup pendek. Tapi tentunya sebagai programmer kita akan lebih tertantang jika mampu membuat fungsi tersebut sendiri ☺. Untuk akan saya bahas alur / logika pemrogramannya , sehingga mudah untuk di translasikan dalam bahasa pemrograman lain

Untuk bisa membuat fungsi membilang dalam bahasa Indonesia kita harus tahu dulu , cara manusia membilang bilangan . Untuk itu mari kita analisa proses tersebut !!!!! ☺

Bilangan satuan

0 -> nol

1 -> satu

9 -> sembilan

Ket : “->” berarti “dibaca”

Untuk proses membilang bilangan satuan tentunya tidak terlalu susah jika di implementasikan dalam bahasa C . Anda bisa menggunakan seleksi untuk masing masing bilangan . Tapi penulis akan menggunakan bantuan array

Berikut ialah potongan source codenya fungsi untuk membilang bilangan satuan

```
char StrBilangan[10][10] =
{"nol", "satu", "dua", "tiga", "empat", "lima", "enam", "tujuh", "delapan", "sembi
lan"};

void SaySatuan(char nilai) //0..9
{
    printf("%s", StrBilangan[nilai]);
}
```

Bilangan puluhan

10 -> sepuluh

11 -> sebelas

12 -> dua belas

13 -> tiga belas

19 -> sembilan belas

20 -> dua puluh

21 -> dua puluh satu

45 -> empat puluh lima

99 -> sembilan puluh delapan

Secara umum untuk membilang bilangan puluhan bisa dilakukan dengan cara *membilang puluhannya secara satuan dan kemudian diikuti dengan “puluh” dan dilanjutkan dengan membilang satuannya jika nilaisatuannya bukan 0.*

Untuk lebih jelas mari kita perhatikan contoh berikut

Contoh : Kita akan membilang nilai 35

Kita ambil puluhannya dahulu . Untuk mengambil nilai puluhannya bisa didapat dengan membaginya dengan 10 .

$$\begin{aligned} \text{NilaiPuluhan} &= 35 / 10 \\ &= 3 \end{aligned}$$

Nilai Satuan bisa didapat dengan mengambil sisa pembagian dengan bilangan 10

$$\begin{aligned} \text{NilaiSatuan} &= 35 \% 10 \\ &= 5 \end{aligned}$$

Nah setelah dapat kita tinggal menggunakan cara diatas :
SaySatuan(NilaiPuluhan) + "puluh" + SaySatuan(NilaiSatuan)
*SaySatuan ialah fungsi untuk mencetak nilai satuan
Maka akan terbentuk tiga puluh lima

Contoh berikutnya kita ambil 40
NilaiPuluhan = 4
NilaiSatuan = 0

Karena nilai satuan sama dengan 0 untuk itu kita tidak perlu membilang nilai satuannya jadi cukup dengan :

SaySatuan(NilaiPuluhan) + "puluh"

Maka akan didapat hasil "empat puluh"

Jadi jelaslah maksud kata jika nilainya bukan 0 , yaitu untuk menghindari terjadinya (pada kasus 40) tercetak empat puluh nol

Cara diatas memang benar untuk sebagian besar nilai puluhan tapi ada beberapa nilai yang perlu perkecualian . Dari contoh data analisa diatas dapat dilihat bahwa nilai 10,11,12 .. 19 tidak memenuhi cara diatas. Tapi jika dianalisa lebih lanjut bilangan 12..19 ternyata memiliki pola yang sama yaitu : *membilang satuan dan diikuti dengan kata belas*

Untuk contoh saya ambil 14.

NilaiSatuan = 4 .

Maka dengan menggunakan cara diatas :

SaySatuan(NilaiSatuan) + "belas"

Untuk 10 dan 11 anda bisa gunakan seleksi biasa .

Berikut ialah source code untuk membilang bilangan puluhan .

```
void SayPuluhan(char nilai) // 10..99
{
    if (nilai < 10)          // seleksi , jika lebih kecil dari 10
        SaySatuan(nilai);   // tentunya masuk ke fungsi SaySatuan
    else
    {
        if (nilai == 10)
            printf("sepuluh");
        if (nilai == 11)
            printf("sebelas");
        if (nilai >= 12 && nilai <= 19) //Cetak satuan diikuti dengan belas
        {
            SaySatuan(nilai % 10);      // Satuan bisa dicari dengan memodulo
            printf("belas");            // bilangan dengan 10 ( mencari hasil sisa
        }                                // pembagian

        if (nilai >= 20 && nilai <= 99) //Cetak puluhan diikuti kata puluh
        {
            SaySatuan(nilai / 10);      // kemudian cetak satuan
            printf(" puluh ");          // Nilai Puluhan bisa dicari dengan
            SaySatuan(nilai % 10);      // membaginya dengan 10
        }
    }
}
```

```
}  
}  
}
```

Bilangan ratusan

100 -> seratus
101 -> seratus satu
110 -> seratus sepuluh
189 -> seratus delapan puluh sembilan
200 -> dua ratus
201 -> dua ratus satu
999 -> sembilan ratus sembilan puluh sembilan

Dapat dilihat bilangan ratusan juga memiliki pola yang secara umum hampir sama yaitu : *membilang nilai ratusannya diikuti kata ratus dan dilanjutkan dengan membilang nilai puluhannya jika nilai puluhannya bukan 0.*

Kita ambil contoh 223

NilaiRatusan = 223 / 100
 = 2

NilaiPuluhan = 223 % 100
 = 23

Dengan cara diatas didapat :

SaySatuan(NilaiRatusan) + "ratus" + SayPuluhan(NilaiPuluhan)

Tentunya jika fungsi SayPuluhan anda benar akan tercetak :

dua ratus dua puluh tiga

Untuk perkecualian yaitu bilangan antara 100..199 . Untuk bilangan tersebut dapat dibilang dengan rumus Cetak "seratus" diikuti dengan membilang bilangan puluhannya

Berikut ialah cuplikan SourceCodennya dalam C

```
void SayRatusan(int nilai) // 100..999  
{  
    if (nilai < 100)  
        SayPuluhan(nilai); // seleksi , jika puluhan  
    else  
    {  
  
        if (nilai >= 100 && nilai <= 199) // jika dibawah 200 gunakan kata  
  
            printf("seratus "); // seratus  
        if (nilai >= 200 && nilai <= 999) // jika diatasnya , bilang nilai  
        { // ratusannya diikuti dengan kata  
            SaySatuan(nilai / 100);  
            printf(" ratus "); // ratus  
        }  
  
        if (nilai % 100 != 0) //Hindari jika kelipatan 100  
            SayPuluhan(nilai % 100); //Bilang nilai puluhannya  
    }  
}
```

Bilangan ribuan

1.000 -> seribu

1.100 -> seribu seratus

2.111 -> dua ribu seratus sebelas

10.000 -> sepuluh ribu

25.250 -> dua puluh lima ribu dua ratus lima puluh

130.750 -> seratus tiga puluh ribu , tujuh ratus lima puluh

999.999 -> sembilan ratus sembilan puluh sembilan ribu , sembilan ratus sembilan puluh sembilan

Jika diperhatikan ternyata untuk mencetak bilangan ribuan caranya sama sampai mencetak bilangan ratusan ribu yaitu : *bilang nilai ribumannya diikuti dengan kata "ribu" dan dilanjutkan dengan bilang nilai ratusannya* jika nilai ratusannya bukan 0.

Sebagai contoh saya akan ambil 256.750

NilaiRibuan = 256.750 / 1000
= 256

NilaiRatusan = 256.750 % 1000
= 750

Dengan menggunakan cara diatas :

SayRatusan(NilaiRibuan) + "ribu" + SayRatusan(NilaiRibuan)

Tentunya jika Fungsi SayRatusan anda benar akan tercipta :

dua ratus lima puluh enam ribu tujuh ratus lima puluh

Dan seperti biasa perkecualian terdapat pada bilangan antara 1000..1999 , yaitu cukup dengan *mencetak "seribu" diikuti dengan membilang nilai ratusannya jika tidak 0*

```
void SayRibuan(unsigned long nilai) //1000...999999
{
    if (nilai < 1000)
        SayRatusan(nilai); //Seleksi jika ratusan
    else
    {
        if (nilai >= 1000 && nilai <= 1999) //Seleksi nilai dibawah 2000
            printf("Seribu ");
        if (nilai >= 2000 && nilai <= 999999)
        {
            SayRatusan(nilai/1000); //Cetak digit ribuan secara Ratusan
            printf(" ribu "); // diikuti kata ribu
        }

        if (nilai % 1000 != 0) // seleksi kelipatan 1000
            SayRatusan(nilai % 1000); // diikuti dengan mencetak digit sisanya
    }
}
```

Begitu pula halnya untuk bilangan diatas 999.999 . Tidak terlalu susah bukan ?

Ternyata untuk membuat fungsi membilang anda hanya perlu operator bagi (/) dan modulus (%), dan tentunya sedikit analisa ☺ Sebenarnya program diatas masih cukup panjang dan tentunya masih bisa dipendekkan dan dioptimasi lagi. Nah untuk tugas ini saya serahkan sepenuhnya kepada pembaca saja ☺.

Oke saya rasa sekian saja Kita jumpa lain waktu lain kesempatan

Berikut ialah source code lengkap program membilang tersebut

```
#include <stdio.h>

char StrBilangan[10][10] =
{"nol", "satu", "dua", "tiga", "empat", "lima", "enam", "tujuh", "delapan", "sembi
lan"};

void SaySatuan(char nilai) //0..9
{
    printf("%s", StrBilangan[nilai]);
}

void SayPuluhan(char nilai) // 10..99
{
    if (nilai < 10)
        SaySatuan(nilai);
    else
    {
        if (nilai == 10)
            printf("sepuluh");
        if (nilai == 11)
            printf("sebelas");
        if (nilai >= 12 && nilai <= 19)
        {
            SaySatuan(nilai % 10);
            printf("belas");
        }
        if (nilai >= 20 && nilai <= 99)
        {
            SaySatuan(nilai / 10);
            printf(" puluh ");
            SaySatuan(nilai % 10);
        }
    }
}

void SayRatusan(int nilai) // 100..999
{
    if (nilai < 100)
        SayPuluhan(nilai);
    else
    {
        if (nilai >= 100 && nilai <= 199)
            printf("seratus ");
        if (nilai >= 200 && nilai <= 999)
```

```
{
    SaySatuan(nilai / 100);
    printf(" ratus ");
}

if(nilai % 100 != 0) //untuk menghindari seratus nol
    SayPuluhan(nilai % 100);
}

void SayRibuan(unsigned long nilai) //1000...999999
{
    if (nilai < 1000)
        SayRatusan(nilai);
    else
    {
        if (nilai >= 1000 && nilai <= 1999)
            printf("Seribu ");
        if (nilai >= 2000 && nilai <= 999999)
        {
            SayRatusan(nilai/1000);
            printf(" ribu ");
        }

        if (nilai % 1000 != 0)
            SayRatusan(nilai % 1000);
    }
}

void SayJuta(unsigned long nilai) //1.000.000 -> 999.999.999
{
    if (nilai < 1000000)
        SayRibuan(nilai);
    else
    {
        SayRatusan(nilai / 1000000);
        printf(" juta ");
        if (nilai % 1000000 != 0)
            SayRibuan(nilai % 1000000);
    }
}

void SayMilyar(unsigned long nilai) // 1.000.000.000 -> 999.999.999.999
{
    if (nilai < 1000000000)
        SayJuta(nilai);
    else
    {
        SayRatusan(nilai / 1000000000);
        printf(" Milyar ");
        if (nilai % 1000000000 != 0)
            SayJuta(nilai % 1000000000);
    }
}

void SayBilangan(unsigned long nilai) // Fungsi pengarah
{
```

```
if (nilai <= 9)
    SaySatuan(nilai);
if (nilai >= 10 && nilai <= 99)
    SayPuluhan(nilai);
if (nilai >= 100 && nilai <= 999)
    SayRatusan(nilai);
if (nilai >= 1000 && nilai <= 999999)
    SayRibuan(nilai);
if (nilai >= 1000000 && nilai <= 999999999)
    SayJuta(nilai);
if (nilai >= 1000000000)
    SayMilyar(nilai);
}

void main(void)
{
    SayBilangan(163);
    SayBilangan(25234);
    SayBilangan(-1);
}
```