Cepat Mahir Algoritma dalam C

I Putu Gede Darmawan
IPGD_BALI@yahoo.com

Analisa Numerik: Mengecek Bilangan Prima

Bilangan Prima ialah Bilangan yang hanya memiliki 2 faktor yaitu , 1 dan bilangan itu sendiri.

Contoh :

7 merupakan bilangan Prima karena 7 hanya memiliki 2 faktor yaitu 1 dan 7 saja
8 bukan merupakan bilangan Prima karena 8 bisa memiliki lebih dari 2 faktor yaitu :
1 2 4 8

Dengan menggunakan pengertian diatas dapat dibuat algoritma :

1. Input N // N = Bilangan Prima yang akan di cek
2. counter = 0 // counter menyatakan jumlah faktor

3. FOR I = 1 to N step = 1 // step langkah setiap perulangan (i++)
   Jika N % I == 0 maka // Jika N habis di bagi I
   counter++ //Tambahkan Counter
4. Jika counter == 2 maka
   "N Bilangan Prima"
   Jika Tidak
   "N Bukan Bilangan Prima"

Jika kita telaah lebih lanjut , untuk mengecek bilangan ke N maka diperlukan N kali iterasi (langkah 3) . Oke kalau begitu mari kita sederhanakan lagi

Kita tahu bahwa

1. Bilangan genap kecuali 2 bukan merupakan bilangan prima
2. Bilangan Genap ialah bilangan yang kelipatan 2
3. Setiap Bilangan pasti memiliki faktor 1 dan bilangan itu sendiri

Karena semua bilangan genap kecuali bukan prima dan Bilangan genap merupakan kelipatan 2 jadi kita dapat *menskip* pembagian dengan bilangan genap (step = 2) asalkan pertama kali dilakukan pengecekan bilangan genap.
Karena Setiap bilangan pasti memiliki faktor 1 dan bilangan itu sendiri jadi kita tidak perlu mencek 1 dan bilangan itu sendiri , tapi cukup mulai dari 2 sampai dengan n -1 .Jika ada faktor pada range tersebut sudah barang tentu bahwa bilangan tersebut bukan bilangan prima.

Baiklah mari kita sempurnakan algoritmanya

1. Input N
2. Jika N == 2 maka // 2 Merupakan Bilangan Prima
   "N Bilangan Prima"
3. Jika N % 2 == 0 maka // Bilangan Genap bukan bilangan Prima
   "N Bukan Bilangan Prima"
4. FOR I = 3 To N-1 STEP 2 // Tidak perlu mengadakan pengecekan
   Jika N % I == 0 maka // terhadap bilangan genap
   "N Bukan Bilangan Prima"
5. "N Bilangan Prima

Dengan sedikit analisis , algoritma diatas telah mampu dioptimasi . Jika sebelumnya untuk mengecek bilangan ke N pasti memerlukan N iterasi . Maka pada keadaan sekarang jika bilangan itu bilangan genap maka tidak perlu itersi (best case) jika bukan bilangan genap maka diperlukan paling banyak N / 2 iterasi (worst case) .

Nah apakah sampai disini saja ? Mungkin ada yang kepikiran mau optimasi apa lagi ?? ☺
Sudah menyerah ??

Ternyata ada sebuah optimasi lagi yang dapat dilakukan . Apa itu ??? . Sebelumnya mari kita lihat sama – sama tabel di bawah ini:

8	
1	8
2	4
4	2
8	1

Tabel Diatas ialah Tabel Faktor dari 8 . Kalau diperhatikan ternyata pada nilai 4 (kolom 1 baris 3) sudah ada pada kolom 2 . Contoh lain

12	
1	12
2	6
3	4
4	3
6	2
12	1

Tabel diatas ialah Tabel Faktor dari 12 . Kalau diperhatikan ternyata nilai 4 (kolom 1 baris) sudah ada pada kolom 2 .

Apa yang bisa diambil dari contoh diatas ? , apa yang bisa dimanfaatkan untuk mengoptimasi algoritma pengecekan bilangan prima diatas ??

Dari tabel dapat disimpulkan :
faktor 8 :
8 : 1 = 8 ; 8 : 2 = 4 ; 8 : 4 = 2 dst
pada kasus nilai 8 tidak perlu diadakan pengecekan melebihi dari nilai 2
faktor 12 :
12 : 1 = 12 ; 12 : 2 = 6 ; 12 : 3 = 4 ; 12 : 4 = 3 dst
pada kasus nilai 12 tidak perlu diadakan pengecekan melebihi dari nilai 3

Artinya kita tidak perlu mengecek semua faktor dari bilangan. melainkan cukup setengahnya dari faktornya saja yaitu pada akar dari bilangan tersebut , karena faktor sisanya merupakan hasil pembagian dari bilangan tersebut dengan faktornya .

Nah dengan analisis tersebut algoritma diatas dapat disederhanakan menjadi :

1. Input N
2. Jika N == 2 maka
   “N Bilangan Prima”
3. Jika N % 2 == 0 maka

"N Bukan Bilangan Prima"

4. FOR I = 3 To Sqrt(N) STEP 2 // Sqrt(N) -> Akar dari N
   Jika N % I == 0 maka
   "N Bukan Bilangan Prima"
5. "N Bilangan Prima

Algoritma diatas membutuhkan 0 itersi saat N bilangan genap . Sedangkan membutuhkan maksimum Akar(N) iterasi saat bilangan tersebut bukan bilangan genap .

Berikut Source code lengkap dari algoritma diatas .

```
#include <stdio.h>
#include <math.h>

int isprima(int n)
{
 int li;
 if (n == 2)
  return 1;
 if (n % 2 == 0 || n == 1)
  return 0;

 for(li = 3;li <= sqrt(n);li+=2)
 {
  if (n%li == 0)
   return 0;
 }
 return 1;
}

void main(void)
{
 int li;
 printf("Bilangan prima dari 1 sampai 100 : \n");
 for(li = 1;li<=100;li++)
 if (isprima(li))
  printf("%3d",li);
}
```

Berikut Hasil dari source code diatas :

```
Bilangan Prima dari 1 sampai 100 :
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```