APLIKASI TEKNOLOGI GREEN COMPUTING MELALUI OPTIMALISASI KINERJA KOMPUTER

Hidayatulah Himawan
Jurusan Teknik Informatika UPN "Veteran" Yogyakarta
Jl. Babarsari 2 Tambakbayan 55281 Telp (0274) 485323
Email : if.iwan@gmail.com

Abstract

Green Computing which is a concept of using technology to become more practical, efficient and cost-efficient in energy use, becomes a concept which is being developed at this time. One method that can be analyzed and developed is the use of a single processor on each computer, but has a performance that could rival the use of multiple processors in order to get better performance. Besides eco-friendly principles also become an integral part of the development of this technology. So optimizing the performance of computers using a single processor of choice in order to optimize the use of technology in today.
Keywords : Bit Array, Processor, Prime Number

Green Computing yang merupakan suatu konsep penggunaan teknologi agar menjadi lebih praktis, efisien dan hemat dalam penggunaan energi, menjadi suatu konsep yang terus dikembangkan pada saat ini. Salah satu metode yang dapat dianalisa dan dikembangkan adalah penggunaan processor tunggal pada setiap komputer, namun memiliki kinerja yang dapat menyamai penggunaan processor ganda agar kinerja menjadi lebih baik. Selain itu prinsip ramah lingkungan juga menjadi bagian yang tidak terpisahkan dari pengembangan teknologi ini. Maka optimalisasi kinerja komputer menggunakan processor tunggal menjadi pilihan agar dapat dioptimalkan dalam penggunaan teknologi di saat ini.
Kata Kunci : Bit Array, Processor, Prime Number

1. Pendahuluan

Green computing yang merupakan suatu konsep penggunaan teknologi agar menjadi lebih ramah lingkungan, menjadi suatu permasalahan bersama di masa yang akan datang. Penggunaan teknologi yang lebih ramah lingkungan menjadi perhatian di seluruh bagian teknologi. Mulai dari PC (personal Computer), processor, memory, media penyimpanan dan segalanya. Salah satu konsep yang dapat dikembangkan adalah tentang bilangan prima. Dimana bilangan prima telah menjadi salah satu bahasan pokok dari riset-riset yang dilakukan saat ini.

Dari riset-riset yang telah dilakukan sebelumnya, ditemukan beberapa teknik dalam pencarian apakah sebuah bilangan adalah bilangan prima atau bukan, seperti *Fermat Little Test*. Namun dalam hal pencarian pola deret bilangan prima tidaklah semudah itu. "Definisi bilangan prima yang sederhana tersebut tidak menjamin bahwa bilangan prima muncul dengan pola-pola yang teratur, bahkan sebaliknya; tidak ada yang dapat memprediksi kemunculan bilangan prima yang selanjutnya (Jansen, 2007)".

Generator bilangan prima merupakan suatu *tool,* yang dapat membangkitkan pola-pola urutan bilangan prima yang teratur, struktur bit-array merupakan metoda pengelompokan variable-variabel yang berisi kumpulan data dengan setiap elemen datanya bertipe sama, digunakan didalam penyimpanan urutan bilangan yang dihasilkan. Bilangan prima sangat bermanfaat untuk kegunaan-kegunaan antara lain dapat diterapkan sebagai basis dari penciptaan algoritma kriptografi kunci public, bagian yang penting dari sistem kriptografi ini adalah faktorisasi bilangan yang sangat besar menjadi factor bilangan primanya, permasalahan yang masih ada yaitu penentuan deret bilangan prima yang skalanya sangat besar. Penggunaan serial prosesor didalam pencarian deret bilangan prima yang sekalanya amat besar kurang efisien mengingat diperlukannya waktu komputasi yang cukup panjang, begitu pula penggunaan prosesor jamak di dalam pencarian deret bilangan prima akan menyangkut kepada masalah harga dan membutuhkan perangkat lunak yang baru. Sehingga dengan cara

menggunakan struktur bit-array diharapkan kesulitan didalam mencari pola-pola urutan bilangan prima dapat diatasi meskipun tanpa menggunakan prosesor jamak sekalipun.

2. Pemanfaatan Green Computing

Tidak heran bahwa regulasi metode green computing ini sudah lahir di beberapa negara maju seperti Amerika yang mengeluarkan program energy star tahun 1992. Kemudian di Eropa ada *Swedish Confederation of Professional Employees* yang menyertifikasi produk peralatan komputer dan perkantoran agar sesuai standard aturan baku. Lalu kita mengenal adanya RoHS *(Restriction of Hazardous Substances)*, yang biasa kita temui dalam suatu produk yang artinya produk tsb menggunakan bahan material yang telah memenuhi syarat yang berlaku. Beberapa metode pendekatan Green Computing sehingga lebih ramah lingkungan dalam sistem komputasi berikut ini:

1. **Pemanfaatan energi alternatif**
   - Dengan menggunakan sumber energi alternatif lain sistem komputasi dapat digerakkan, misalnya menggunakan energi nuklir, hydroelectric, kincir angin, dsb.
2. **Penggunaan Sistem Teknologi Virtualisasi**
   - Memang sudah jamannya di saat sekarang kita menggunakan sistem teknologi virtualisasi. Sistem ini memungkinkan kita dapat menjalankan suatu proses di dua atau lebih *logical computer* didalam satu unit hardware pemroses utama atau CPU. Jadi dengan sistem virtualisasi maka seorang Administrator akan lebih mudah menggabungkan beberapa sistem CPU fisik ke dalam bentuk *Virtual Machine* yang jauh lebih handal. Tentunya hal ini dapat mengurangi jumlah keseluruhan sumber daya yang ada.
3. **Pengaturan penggunaan sumber daya**
   - Komputasi modern saat ini sudah dilengkapi adanya fasilitas *Advance Configuration and Power Interface*, yang memungkinkan suatu sistem operasi mengendalikan secara langsung faktor *power saving* dari hardware komputer tsb. Misal Anda dapat mengatur monitor atau harddisk secara otomatis akan *idle* (dalam keadaan tidak ada aktivitas). Disamping itu Anda dapat mengatur sistem *power management* tsb didalam BIOS komputer dan fitur-fitur lain dalam masing-masing produk.
4. **Penggunaan Hardware yang *Low Power***
   - Saat ini sudah banyak produk-produk komputer yang menggunakan sumber daya yang rendah. Namun hal ini akan tidak sesuai bila Anda menggunakan komputer untuk game, video editing, dan aplikasi-aplikasi yang membutuhkan *resources* yang besar. Apabila Anda hanya membutuhkan untuk sistem office dan sekedar browsing di internet, komputer dengan hardware yang minimalis ini sudah lebih dari cukup dan tentunya bisa menghemat *budge*t Anda.
5. **Pemanfaatan sistem Daur Ulang**
   - Bila komputer sudah *usang* namun masih dapat dimanfaatkan kembali biasanya sparepart yang masih berfungsi mungkin dapat dipakai sebagai sumber bagi komputer lain. Sedangkan kalo sudah rusak maka sparepart yang ada dapat di daur ulang kembali. Biasanya ini terjadi di negara-negara maju seperti Amerika dan Eropa.
6. **Penggunaan Sistem Mobilitas**
   - Infrastruktur yang ada seperti saluran kabel telpon mungkin saat ini dapat berkurang berkat adanya inovasi dalam teknologi informasi dimana kita sudah dapat melakukan komunikasi menggunakan VOIP *(Voice Over Internet Protocol)*. Secara tidak langsung kita sudah mengurangi penggunaan bahan-bahan dari logam yang berbahaya dari struktur kabel tsb.
   -

3. Algoritma Penelitian

Disini akan dibahas jenis algoritma yang akan digunakan sebagai pengembangan dari *Green Computing*. Yaitu :

3.1 Algoritma *Sieve Of Eratosthenes*

Eratosthenes (276-194 SM) adalah petugas perpustakaan ketiga dari perpustakaan terkenal di Alexandria dan dia adalah seorang sarjana yang sangat hebat. *Erastothenes*

dikenang dengan pengukurannya terhadap keliling bumi dengan memperkirakan jarak antara bumi dengan matahari dan bulan. Algoritma *Sieve Of Eratosthenes* adalah sebuah algoritma klasik untuk menemukan seluruh bilangan prima sampai ke sebuah N yang ditentukan. Mulai dengan *array of integer* yang belum dicoret dari 2 ke N. Integer pertama yang belum dicoret yaitu 2, adalah bilangan prima pertama. Coret seluruh kelipatan dari bilangan prima ini. Ulangi pada integer selanjutnya yang belum dicoret. Algoritma *Sieve of Eratosthenes* ini digunakan penulis sebagai dasar untuk membuat Generator bilangan prima dengan menggunakan struktur bit array. Urutan langkah dari Algoritma *Sieve of Eratosthenes* dapat dilihat pada penjelasan berikut ini :

1. Buat sebuah *list* bilangan dari 2 sampai sebuah bilangan terbesar yang kita nyatakan dengan *n*.
2. Eliminasi atau hilangkan dari *list* semua bilangan kelipatan 2.
3. Angka selanjutnya yang belum dihilangkan adalah bilangan prima.
4. Eliminasi atau hilangkan dari *list* semua bilangan yang merupakan kelipatan dari angka pada langkah sebelumnya.
5. Ulangi langkah 3 dan langkah 4 sampai angka yang lebih besar dari angka yang tersisa setelah langkah 5 adalah bilangan prima.

Algoritma ini mempunyai kompleksitas waktu O (n log log n). Versi algoritma ini dengan optimasi proses, seperti *wheel factorization*, mempunyai kompleksitas waktu O(n).
Sebagai contoh, berikut adalah *array* pada awalnya:
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
Karena 2 belum dicoret, maka 2 adalah bilangan prima pertama kita. Kita coret seluruh kelipatan 2, yaitu 4, 6, 8, 10, 12, dst.
2 3 5 7 9 11 13 15 17 19 21 23 25 27...
Integer selanjutnya yang belum dicoret adalah 3, maka 3 adalah prima dan kita coret seluruh kelipatan 3, seperti 6, 9. 12, dst.
2 3 5 7 11 13 17 19 23 25 ...
5 adalah bilangan prima selanjutnya dan kita mencoret seluruh kelipatan 5. Satu-satunya bilangan yang dicoret dalam *range* ini adalah 25, yaitu **2 3 5** 7 11 13 17 19 23 ...

3.2 Algoritma *Sieve of Atkins*

Dalam penelitian ini dipilih algoritma *Sieve of Atkins* sebagai Algoritma pembanding. Karena algoritma *Sieve of Atkin* adalah sebuah algoritma yang cepat dan modern untuk mendapatkan seluruh bilangan prima sampai pada suatu batas integer yang telah ditentukan. Algoritma ini adalah versi optimasi dari algoritma kuno *Sieve of Eratosthenes*. *Sieve of Atkin* melakukan dahulu beberapa pekerjaan persiapan dan kemudian mencoret kelipatan dari kuadrat bilangan prima, bukannya kelipatan bilangan prima. Algoritma ini diciptakan oleh A. O. L. Atkin dan Daniel J. Bernstein pada tahun 2004. Pada algoritma ini, suatu bilangan dapat disebut bilangan prima jika bilangan tersebut memenuhi beberapa persyaratan, yaitu:

1. Persamaan berikut ini harus memiliki jumlah solusi n yang ganjil:
   a. $4x^2 + y^2 = n$ dimana n mod 4 =1
   b. $3x^2 + y^2 = n$ dimana n mod 6 =1
   c. $4x^2 - y^2 = n$ dimana n mod 12 =11
2. Bilangan tersebut haruslah *square-free*. Sebuah bilangan yang disebut *square-free* adalah bilangan yang tidak memiliki faktor prima lebih dari sekali. (Contoh: 20=2*2*5, 20 memiliki 2 buah faktor 2 sehingga bukan merupakan *square-free*. 21=7*3, 21 tidak memiliki faktor yang dobel dan merupakan bilangan yang *square free*).

Cara implementasi dari algoritma ini adalah:

1. Ciptakan sebuah *array of Boolean* dengan ukuran n dan seluruh elemen diset menjadi *false*
2. Ulangi melalui setiap nilai valid untuk x dan y, dan balikkan elemen dalam posisi hasil perhitungan (Contoh: x = 3, y = 5, $4x^2 + y^2 = 61$ maka apapun nilai elemen ke 61 dari *array of Boolean* tadi harus dibalik).
3. Langkah terakhir adalah dengan mencoret semua bilangan yang tidak *square-free*. Hal ini dapat dilakukan dengan memulai pada awal dari *array of Boolean*, dan jika elemen pertama bernilai *true* (yang berarti bilangan tersebut prima) maka set seluruh elemen yang lain yang merupakan kelipatan dari kuadrat bilangan tersebut menjadi *false*. Lakukan untuk semua elemen kurang dari akar dari batas angka.
4. Elemen yang berisi nilai *true* mewakilkan bilangan prima.

4. Hasil dan Pembahasan

4.1 *Source code* didalam PHP

Pada gambar 1 di bawah ini adalah contoh penerapan algoritma di atas dalam bahasa pemrograman PHP. Script ini sudah ditest untuk menampilkan bilangan prima dibawah 1.000.000 dan berhasil menampilkannya dalam waktu 3 detik.

```
<?php
function bilangan_prima($limit) {
 $prima = array();
 for ($i=2; $i<=$limit; $i++)
 $prima[$i] = true;
 $akarLimit = (int)sqrt($limit);
 for ($i=2; $i<=$akarLimit; $i++) {
 if ($prima[$i]) {
 for ($j=$i*$i; $j<=$limit; $j+=$i) {
 $prima[$j] = false;
 }
 }
 }
 $i = 0;
 foreach ($prima as $bilangan=>$status) {
 if ($status) { echo "$bilangan ";$i++; }
 }
 echo "Jumlahnya:". $i;
}
$start=mktime();
bilangan_prima(1000000); //menampilkan bilangan prima dari 1 - 1 juta
$finish=mktime();
$result=$finish-$start;
echo "Time: $result seconds";
?>
```

Gambar 1. Script PHP penerapan algoritma pada Bilangan Prima

4.2. Algoritma *Sieve Of Eratosthenes*

Algoritma yang digunakan didalam menentukan deret bilangan prima sampai ke n bilangan yang ditentukan adalah Algoritma *Sieve Of Eratosthenes*, disajikan dalam *Pseudocode* dibawah ini :

```
// tentukan batas
limit ← 1.000.000
// set semua bilangan dengan true
is_prime (i) ← true, i [2, limit]
for n in [2, √limit]:
if is_prime (n) :
is_prime (i) ← false,
i {n^2, n^2+n, n^2+2n, …, limit}
for n in [2, limit] :
if is_prime (n) : print n
```

Dari desain pseudocode di atas, maka dapat kita terapkan ke dalam bahasa java seperti program di bawah ini :

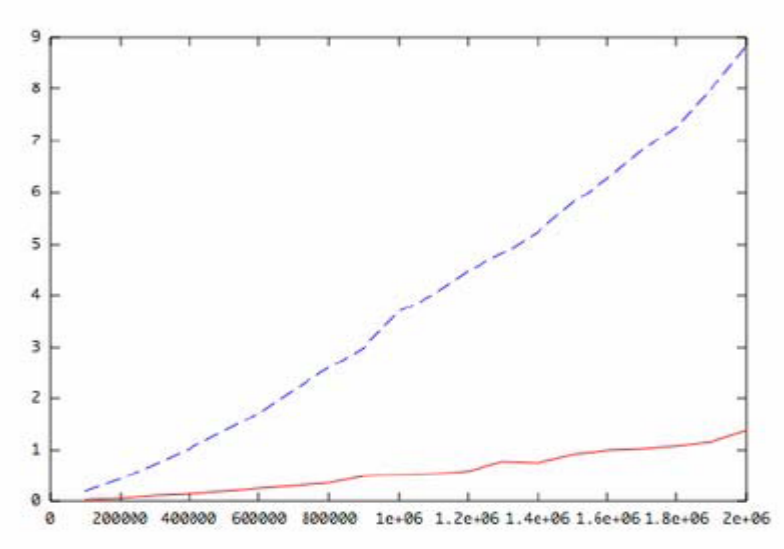
```
Class primaEratosthenes{
public static void main (strings args [] ) {
int m = 10000;//batas
boolean [] prima=new boolean [m+1];
for (int i = 0; i <= m; i++)
        {
          prima [i] = true;
        }
prima[0]=prima[1]=false;
double akarN=math.sqrt(m);
For (int i=2; i<=akarN; i++)
{
        if (prima[i])
        {
                for (int j=i*i; j<=m; j++)
                {
                        if ((j%i)==0)
                        {

        Prima[j]=false;
                        }
                }
        }
}

for (int i=0; i<=m; i++)
{
        if (prima[i]&&(i>=n))
        system.out.println (i + "\n");
        }
        }
}
```

Gambar 3. Algoritma Sieve of Eratosthenes di dalam Bahasa Java

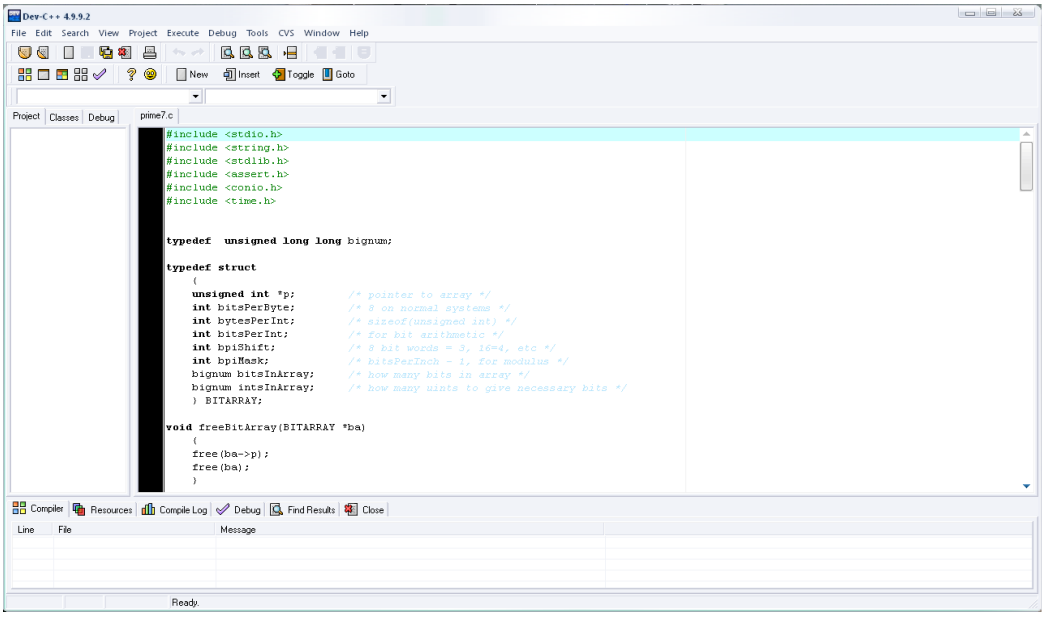
Dengan kode implementasi java di atas, maka kita akan mendapatkan suatu kecepatan dari algoritma untuk mendapatkan bilangan prima dengan suatu batas yang telah kita tentukan sebelumnya.Berikut perbandingan bilangan yang dapat kita lihat antara algoritma *Eratosthenes* dengan algoritma *Brue Force.*



Gambar 4. Perbandingan Algoritma Sieve of Eratosthenes dengan Algoritma Brute Force (naif)

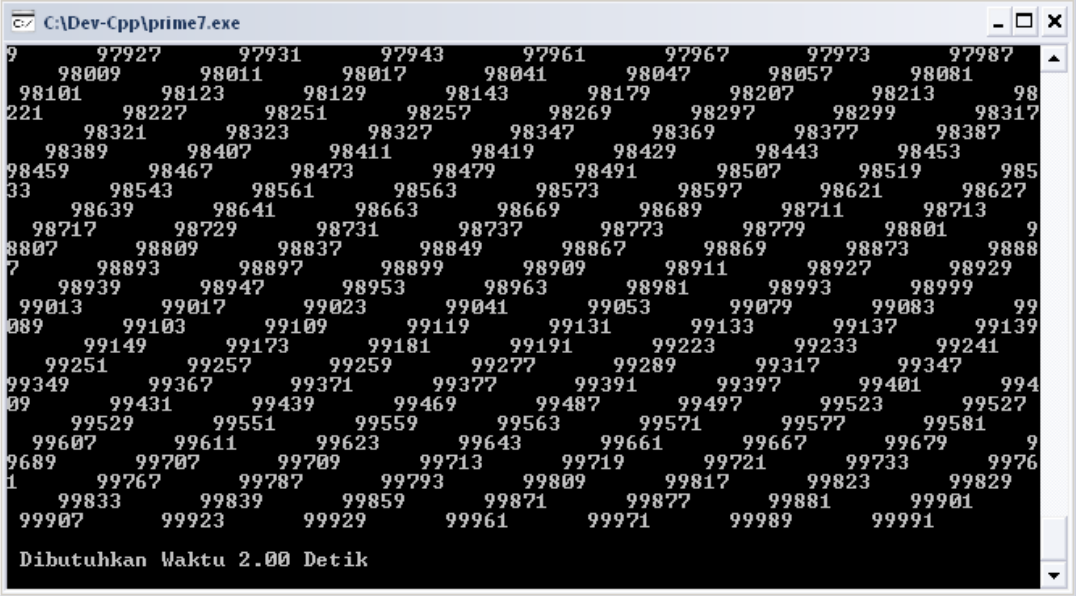
4.3 Eksekusi Program

Eksekusi generator bilangan prima dengan menggunakan struktur bit array yang menggunakan algoritma *Sieve of Eratosthenes* menggunakan *tool* Dev-C++, dapat dilihat pada Gambar 5.



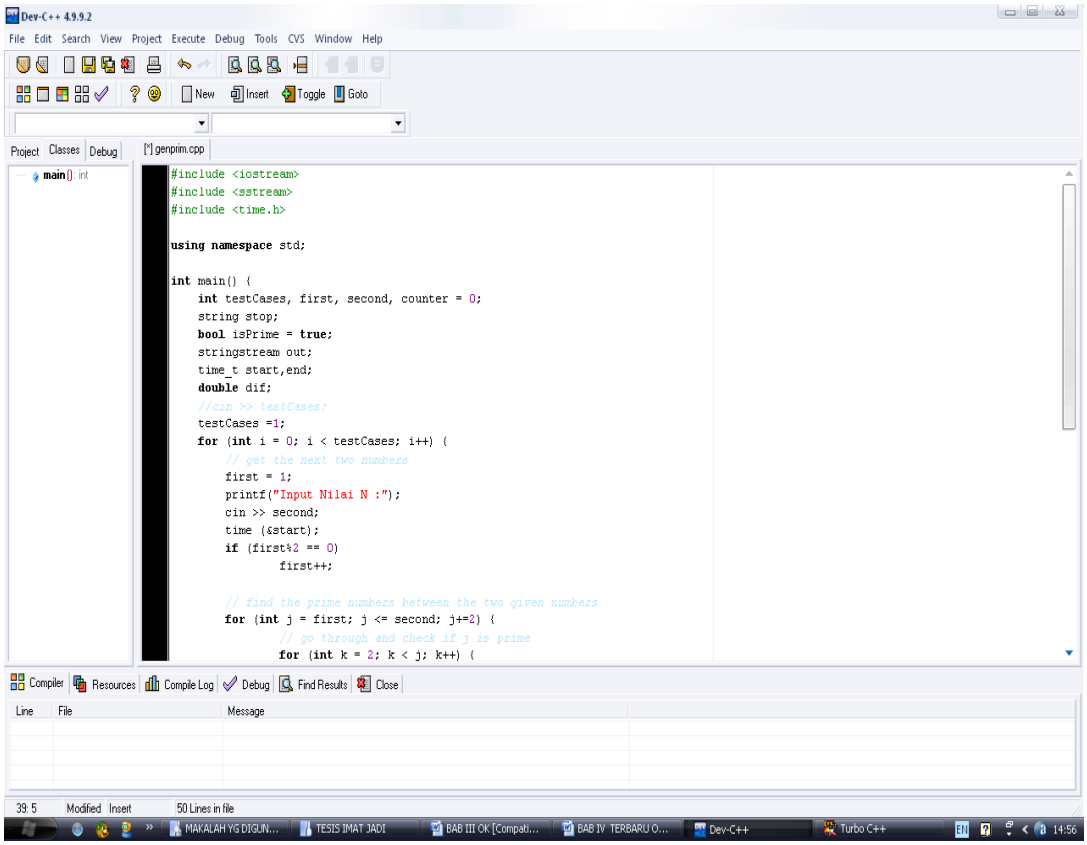
Gambar 5. Tampilan eksekusi program

- Program yang sudah di *Compile* dan sudah tidak ada kesalahan, selanjutnya program di *Run*. Tampilan setelah program di *Run* terlihat pada gambar 6. Didalam contoh kasus ini input bilangan dimasukan angka 100.000. tampak terlihat lama waktu eksekusi selama 2.00 detik.



Gambar 6. Tampilan Setelah di *run*

- Program generator bilangan prima tanpa struktur bit array menggunakan algoritma *Sieve of Atkins* dilakukan pengecekan dengan memakai *tool* Dev-C++, dapat dilihat pada Gambar 7.



Gambar 7. Tampilan Eksekusi Program

5. Kesimpulan

Kesimpulan yang dapat diambil dari hasil percobaan dan analisa adalah :

1. Bilangan prima adalah salah satu metode yang dapat dikembangkan untuk mengahsilkan suatu produk teknologi yang memiliki konsep ramah lingkungan (*Green Computing*).
2. Teknologi *green computing* dapat dikembangkan ke dalam berbagai macam aspek teknologi. Sehingga teknologi yang dihasilkan akan menjadi lebih baik dan memiliki peran serta kemampuan yang lebih optimal.
3. Dengan cara menggunakan struktur bit-array kesulitan didalam mencari pola-pola urutan bilangan prima yang sekalanya sangat besar dapat diatasi dengan menggunakan prosesor tunggal, dan dapat mempercepat waktu akses.

6. Daftar Pustaka

Abdul Kadir & Heriyanto. 2005. *Algoritma Pemrograman Menggunakan C++,* Andi.

Abdul Kadir. 2004. *Panduan Pemrograman Visual C++*, Andi.

Adrianto, 2008, Pentingnya Metode Pendekatan Green Computing, http://wss-id.org/blogs/adriant/archive/2008/10/24/pentingnya-pendekatan-metode-green-computing.aspx

David Wells. 2005 *Prime Numbers: The Most Mysterious Figures in Math*. John Wiley & Sons. Inc., ISBN 0-471-46234-9.

Eko Budi Purwanto. 2008. *Perancangan & Analisis Algoritma*,Graha ilmu.

Gok Asido Haro. (2007, Mei 06). Re : *Algoritma Pencarian Bilangan Prima*. [Online]. available: if15072@students.if.itb.ac.id [2007, Mei 06].

Hanugra Abidianto. (2009, Januari 04). Re : *Bilangan Prima dan Aplikasinya dalam Bidang Informatika*. [Online]. Available: that.kid.is.sherlock@gmail.com [2009, Januari 04].

Jansen. (2007, Februari 27). Re: *Studi sejarah dan perkembangan bilangan prima.* [Online]. Available: if16028@students.if.itb.ac.id [2007, Februari 27].

John R.Hubbard. 2000. *Data Structures With C++*,McGraw-hill Companies.

Rakhmat, Imat H, 2010. *Generator Bilangan Prima Dengan Menggunakan Struktur Bit Array.* UGM, Yogyakarta.

Rinaldi munir & Leoni lidya. 2002. *Algoritma dan Pemrograman dalam Bahasa Pascal dan C,* Informatika.

Hamonangan Situmorang. (2009, Mei 18). *Algoritma dan Struktur Data,* telecom.ee.itb.ac.id/~monang/if2031/Array_dalam_Bahasa C.

Wahyuni. 2004. *Sistem Berkas,* Andi.

http://www.troubleshooters.com/codecorn/primenumbers/primenumbers.htm

http://en.wikipedia.org/wiki/Bit_array