

# Penerapan Algoritma Pencocokan String *Boyer-Moore* untuk Keamanan Komputer

Eric Cahya Lesmana 13508097  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
if18097@students.if.itb.ac.id

*Keamanan komputer merupakan hal yang harus dipertimbangkan saat ini. Banyak kejahatan komputer yang mengancam komputer kita, apalagi jika komputer kita terhubung dengan jaringan internet dan memiliki data yang diincar oleh orang lain. Oleh karena itu, keamanan komputer merupakan suatu keharusan dalam komputer. Saat ini algoritma pencocokan string banyak sekali digunakan dalam berbagai keperluan seperti pencocokan sidik jari. Dan dengan algoritma pencocokan string ini bisa dimanfaatkan untuk menjaga keamanan dari komputer. Algoritma ini digunakan untuk melakukan pengujian apakah proses atau file yang ingin dijalankan aman untuk komputer kita atau tidak. Dengan tabel yang berisi informasi mengenai daftar kata kunci yang tidak diperkenankan untuk dijalankan dalam komputer. Jika terjadi kecocokan akan dilakukan mekanisme khusus dari komputer untuk proses atau file tersebut. Dalam penerapannya digunakan algoritma Boyer-Moore beserta turunannya untuk menjaga keamanan komputer karena memiliki kompleksitas yang baik. Sehingga diharapkan dengan algoritma ini tidak mempengaruhi peformansi dari komputer dan sekaligus bisa menjaga keamanan komputer dari gangguan dari luar baik spyware, virus, maupun gangguan lain.*

*Kata kunci : keamanan komputer, pencocokan string, Boyer-Moore*

## I. PENDAHULUAN

Dalam berbagai bidang, keberadaan komputer saat ini sangat diperlukan. Komputer digunakan untuk membantu berbagai pekerjaan seperti pencatatan, analisis data, dan sebagainya. Dengan kebutuhan yang semakin kompleks tersebut, keberadaan komputer sangat vital. Jika terdapat masalah dalam kinerja komputer, maka dapat mengganggu kerja.

Dengan adanya hal tersebut, peformansi diperlukan supaya pekerjaan yang dilakukan dapat dikerjakan dengan efektif dan efisien. Jika peformansinya baik maka pekerjaan juga akan dapat ikut baik. Namun, adanya kalanya walaupun peformansi dari komputer sudah baik, pekerjaan yang dilakukan tidak selamanya akan berjalan. Hal ini dimungkinkan karena adanya gangguan dari luar seperti serangan para *cracker* yang ingin mengganggu,

serangan virus yang ingin merusak peformansi komputer, dan sebagainya. Intinya untuk mengatasi hal tersebut diperlukan mekanisme khusus untuk mengatasi permasalahan tersebut. Oleh karena itu diperlukan sebuah metode untuk keamanan komputer.

Dalam keamanan komputer terdapat beberapa mekanisme. Salah satunya adalah dengan algoritma pencocokan pola. Dengan algoritma pencocokan pola dapat digunakan untuk melakukan pengecekan apakah terdapat serangan atau sesuatu yang aneh yang terdapat di komputer tersebut. Jika terdapat kecocokan pola maka akan terdapat mekanisme khusus yang menanganinya, misalnya tidak diizinkan untuk berjalan.

Dalam makalah ini, akan dibahas penggunaan algoritma pencocokan dengan Boyer-Moore untuk keamanan komputer. Algoritma Boyer-Moore merupakan salah satu algoritma pencocokan pola yang banyak digunakan karena lebih mangkus dari pencocokan string lain seperti Brute-Force dan Knuth-Morris-Prett. Dari algoritma Boyer-Moore masih dapat dibentuk algoritma pencocokan string lain sebagai perbaikan seperti algoritma turbo *Boyer-Moore* dan algoritma Apostolico-Giancarlo. Untuk algoritma turbo *Boyer-Moore* akan dijelaskan juga dalam makalah ini. Diharapkan dengan penerapan algoritma *Boyer-Moore* dan turunannya ini dapat membantu dalam menjaga keamanan komputer.

## II. METODE

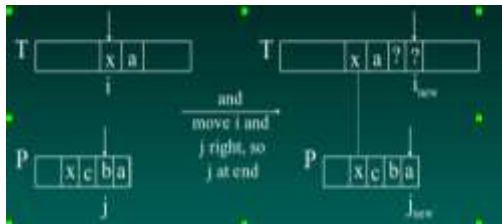
Dalam makalah ini, digunakan metode studi pustaka untuk mendapatkan informasi mengenai algoritma turbo booye *Moore* dan keamanan komputer, serta studi kasus untuk mengetahui efektifitas dari algoritma turbo *Boyer-Moore* dalam pencocokan pola untuk keamanan komputer dibandingkan dengan algoritma *Boyer-Moore*.

### A. Algoritma Boyer-Moore

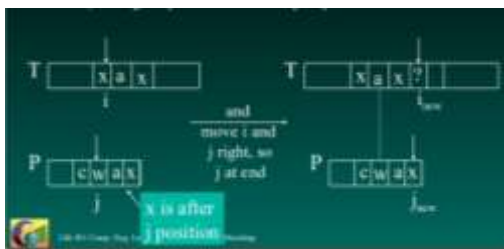
Algoritma Boyer-Moore dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Algoritma *Boyer-Moore* memulai mencocokkan karakter dari sebelah kanan pola. Algoritma *Boyer-Moore* menggunakan dua teknik, yaitu

- Teknik looking-glass mencari pola dalam sebuah teks dengan dimulai dari karakter paling akhir dan dilanjutkan ke depan
- Teknik character-jump merupakan teknik dari algoritma *Boyer-Moore* untuk melakukan lompatan dalam pencocokan pola ketika karakter ke-*i* tidak sama dengan karakter pada pola. Terdapat tiga kemungkinan, yaitu

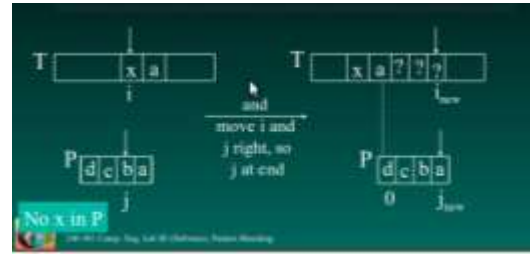
1) jika karakter pada teks tidak sama dengan karakter pada pola dan terdapat karakter pada pola yang dengan karakter pada teks tersebut di sebelah kiri pola. Jika hal ini terjadi maka akan pola akan digeser sampai karakter pada pola sama dengan karakter pada teks yang dicocokkan tersebut.



2) jika karakter pada teks tidak sama dengan karakter pada pola dan terdapat karakter pada pola yang dengan karakter pada teks tersebut di sebelah kanan pola. Jika hal ini terjadi maka akan pola akan digeser ke kanan sebanyak 1 karakter.



3) kondisi pada poin 1 dan poin 2 tidak terpenuhi maka pola digeser ke kanan sehingga karakter paling kiri pada pola berada pada satu karakter di sebelah kanan karakter yang baru saja dicocokkan.



Dalam algoritma *Boyer-Moore* terdapat fungsi kemunculan terakhir dari tiap karakter pada teks untuk pola tertentu yang dicocokkan. Fungsi ini berguna untuk melakukan lompatan ketika melakukan pencocokan sehingga pola bisa loncat sesuai tga kondisi di atas.

Algoritma *Boyer-Moore* memiliki kompleksitas sebesar  $O(nm)$  dan pada kemungkinan terbaik memiliki kompleksitas sebesar  $O(n/m)$ . Pada umumnya, algoritma *Boyer-Moore* cocok untuk teks yang panjang, dan kurang baik untuk teks yang pendek.

```

Procedure preBmBc
input P : array[0..n-1] of char,
input n : integer,
input/output bmBc : array[0..ALPHABETSIZE-1] of integer
)
Deklarasi:
i: integer
Algoritma:
for (i := 0 to ALPHABETSIZE-1)
  bmBc[i] := n
endfor
for (i := 0 to n - 2)
  bmBc[P[i]] := n - i - 1
endfor

```

Pseudocode dari pembuatan tabel fungsi kemunculan terakhir

```

Procedure BoyerMooreSearch(
input m, n : integer,
input P : array[0..n-1] of char,
input T : array[0..m-1] of char,
output ketemu : array[0..m-1] of boolean
)
Deklarasi:
i, j, shift, bmBcShift, bmGsShift: integer
BmBc : array[0..ALPHABETSIZE] of integer
BmGs : array[0..n-1] of integer
Algoritma:
preBmBc(n, P, BmBc)
preBmGs(n, P, BmGs)
i:=0
while (i<= m-n) do
  j:=n-1
  while (j >=0 n and T[i+j] = P[j]) do
    j:=j-1
  endwhile
  if(j < 0) then
    ketemu[i]:=true
    shift := bmGs[0]
  else
    bmBcShift:= BmBc[chartoint(T[i+j])] - n + j + 1
    bmGsShift:= BmGs[j]
    shift:= max(bmBcShift, bmGsShift)
  endif
  i:= i+shift
endwhile

```

pseudocode dari algoritma *Boyer-Moore*

## B. Algoritma Turbo Boyer-Moore

Konsep dasar dari algoritma turbo *Boyer-Moore* adalah sama dengan algoritma *Boyer-Moore*, dengan dua teknik yang telah dijelaskan pada bagian (A). Dalam algoritma turbo *Boyer-Moore* ini memiliki fitur yaitu

1. Teknik ini memungkinkan untuk melompati faktor dari teks tersebut.
2. Teknik ini mengijinkan sebuah penggeseran turbo.

Fitur pergeseran turbo ini memungkinkan jika terdapat potongan teks yang telah dicari sebelumnya sama dengan potongan dari pola yang dicari maka tidak perlu untuk melakukan pencocokan ulang sehingga bisa menghemat dari pencocokan karakter. Hal ini disebabkan oleh algoritma ini mengingat faktor dari teks yang cocok dengan akhiran dari pola selama pencocokan terakhir. Bila dibandingkan dengan algoritma *Boyer-Moore*, algoritma ini tidak membutuhkan pemrosesan ekstra. Dari segi kompleksitas waktu, algoritma ini memiliki kompleksitas sebesar  $O(n)$ .

```
Procedure TurboBoyerMooreSearch
  input n, m : integer;
  input P : array[0..m-1] of char;
  input T : array[0..n-1] of char;
  output ketemu : array[0..n-1] of boolean
}

Deklarasi:
i, j, v1, v2, shift, bmBcShift, bmGsShift, turboShift: integer;
bmBc : array[0..255] of integer;
bmGs : array[0..n-1] of integer;

Algoritma:
preBmBc(n, P, bmBc);
preBmGs(n, P, bmGs);
i := v1 := 0;
shift := n;
while (i <= n-m) do
  j := m-1;
  while (j >= 0 and T[i+j] = P[j]) do
    j := j-1;
  endwhile;
  if (j < 0) then
    ketemu[i] := true;
    shift := bmGs[0];
    v1 := n - shift;
  else
    v2 := m-1-j;
    turboShift := v1 - v2;
    bmBcShift := bmBc[chartoint(T[i+j])] - n + j + 1;
    bmGsShift := bmGs[j];
    shift := max(bmBcShift, bmGsShift);
    shift := max(shift, turboShift);
    if (shift = bmGs[j]) then
      v1 := min(n - shift, v2);
    else
      if (turboShift < bmBcShift)
        shift := max(shift, v1+1);
      endif;
      v1 := 0;
    endif;
  endif;
  i := i + shift;
endwhile;
```

pseudocode dari algoritma turbo *Boyer-Moore*

## C. Keamanan Komputer

Keamanan komputer merupakan bidang yang cukup baru. Dengan semakin maraknya penggunaan internet maka keberadaan dari keamanan komputer dirasakan sangat perlu. Terdapat dua bagian dari keamanan komputer, yaitu keamanan online dan keamanan offline.

Dalam keamanan online misalnya dengan firewall.

Dalam hal ini, perlu adanya batasan untuk mengakses sistem komputer. Pada keamanan online dapat dibagi juga keamanan untuk komputer pribadi dengan keamanan untuk jaringan. Keamanan offline merupakan keamanan dari komputer pribadi terhadap serangan virus. Diperlukan antivirus untuk mencegah terjadinya serangan dari sebuah virus.

## III. ANALISIS PENYELESAIAN MASALAH

Untuk menjaga keamanan komputer dengan pencocokan pola memiliki mekanisme tersendiri. Mekanisme yang dilakukan tersebut bisa dilakukan dengan menggunakan tabel aplikasi atau sistem yang dilarang untuk dijalankan, serta dengan menggunakan variabel penanda dari sebuah file pada sistem untuk dicocokkan dengan tabel atau data terkait ukuran variabel yang dicatat sebagai serangan.

Dari pencocokan dengan menggunakan tabel file atau aplikasi yang dilarang, kita bisa menggunakan beberapa indikator sebagai pola yang dicari dari sekumpulan data aplikasi yang dilarang. Indikator tersebut bisa berupa ekstensi dari sebuah file, nama sistem yang akan berjalan, pola byte yang terdapat dalam file, dan sebagainya. Jika ditemukan kecocokan maka akan dapat dilakukan penanganan khusus lain, seperti menolak paket atau file tersebut untuk dijalankan, melakukan karantina terhadap file tersebut, atau melakukan pembersihan file dari kode yang bisa merusak sistem atau jaringan.

Dari pencocokan dengan menggunakan tabel dengan variabel penanda tertentu, memiliki konsep yang sama dengan bagian sebelumnya. Dalam hal ini, dibentuk sebuah hash atau variabel sementara dari sebuah file dan kemudian dicocokkan dengan tabel hash dari kumpulan file atau proses yang dianggap sebagai pengganggu.

Untuk melakukan hal tersebut, digunakan pencocokan pola dengan algoritma *Boyer-Moore* supaya memiliki kompleksitas waktu yang baik sehingga peformansi komputer yang digunakan tidak terlalu terganggu. Jika pencocokan yang dilakukan lama, maka respon dari komputer juga akan lama. Hal ini akan sangat mengganggu dari kinerja komputer juga. Maksud hati ingin menjaga keamanan komputer, di sisi lain peformansi yang diharapkan baik terganggu.

Cara kerja dari pencocokan pola ini adalah sebagai berikut

- jika terdapat sebuah file yang ingin dieksekusi atau terdapat, maka file atau proses tersebut tersebut tidak langsung diizinkan untuk dijalankan, namun akan dilakukan pengecekan terlebih dahulu
- ketika dilakukan pencocokan dengan algoritma *Boyer-Moore* maka diperlukan pola yang ingin dicari

dengan teks yang ingin diperiksa. Pola yang ingin dicari diisi dengan kata kunci dari file atau proses yang ingin diperiksa, dan teks yang ingin diperiksa diisi dengan kumpulan data dari tabel yang terkait dengan kata kunci tersebut.

- dengan pola dan teks yang ingin dicari tersebut, maka pencocokan pola dilakukan, dan dalam hal ini menggunakan algoritma *Boyer-Moore* atau turunannya untuk menjaga performansi.
- jika ditemukan kecocokan pola, maka file atau proses tersebut tidak diperbolehkan untuk dijalankan, dan akan dilakukan penanganan khusus seperti dikarantina, dihapus atau dibersihkan.
- jika pola tersebut tidak terdapat pada teks yang dicari maka file tersebut bisa dijalankan secara garis besar, mekanismenya seperti itu.

Supaya mekanisme ini bisa menjamin keamanan komputer maka diperlukan data tabel dari kata kunci yang besar. Hal ini ditujukan supaya variasi dari serangan yang dapat dikenali menjadi lebih banyak juga. Oleh karena itu, perlu ada pembaharuan dari tabel supaya jika terjadi penambahan kata kunci baru dapat dimasukkan ke dalam tabel dan komputer bisa mengenalinya. Jika pembaharuan tidak dilakukan maka pengenalan dari kata kunci akan sedikit dan jika terdapat jenis kata kunci baru tidak dapat dikenali dan proses yang bisa merusak komputer akan diizinkan untuk berjalan.

#### IV. PEMBAHASAN

Dari analisis permasalahan yang ada di atas, diharapkan dengan penerapan algoritma pencocokan string dapat menjaga keamanan komputer. Dan diharapkan pencocokan string ini tidak terlalu menghabiskan sumber daya komputer sehingga performansi komputer masih terjaga.

Algoritma *Boyer-Moore* merupakan alternatif yang baik untuk memenuhi harapan tersebut. Algoritma ini merupakan algoritma yang mangkus, jika dibandingkan dengan algoritma pencocokan string lainnya seperti *Brute Force* dan *Knuth-Morris-Pratt*. Keunggulan dari algoritma *Boyer-Moore* untuk menjaga keamanan komputer seperti yang telah dijelaskan adalah sebagai berikut

1. dari pencocokan pola yang ada, algoritma ini pasti akan menemukan pola yang terdapat dalam teks (dengan penerapan algoritma yang benar juga tentunya). Untuk itu, jika digunakan untuk mengenali file atau proses yang dianggap tidak aman maka akan bisa mengenalinya. Asalkan tidak ada mekanisme tertentu yang menyebabkan pola yang dicari disamarkan dan membuat pola tidak ditemukan.

Oleh karena itu, dalam algoritma ini pencocokan string ini pola yang diterima harus sama dengan jenis teks yang ingin dicocokkan.. Jadi jika ada metode kriptografi harus ditangani sebelum dilakukan pencocokan string.

Terlepas dari itu, algoritma *Boyer-Moore* ini akan dapat menemukan apakah file aman dijalankan atau tidak untuk dijalankan pada komputer.

2. dari sisi performansi, algoritma *Boyer-Moore* memiliki kompleksitas waktu yang baik. Dengan adanya teknik lompatan memungkinkan pencocokan karakter dilakukan lebih cepat.

Berikut akan diilustrasikan seberapa cepat pencocokan pola dari algoritma *Boyer-Moore* beserta algoritma turbo *Boyer-Moore*, dibandingkan dengan algoritma *Brute-Force* dan KMP.

Misalkan pola yang dicari adalah

G C A G A G A G

Dan kumpulan data yang dicari sebagai teks adalah

G C A T C G C A G A G A G T A T A C A G T A C G

Dengan algoritma *Brute-Force* akan memerlukan pencocokan sebanyak 16 kali (sampai ditemukan pola).

Ilustrasinya sebagai berikut :

G C A T C G C A G A G A G T A T A C A G T A C G  
 1 2 3 4  
 G C A G A G A G

G C A T C G C A G A G A G T A T A C A G T A C G  
 5  
 G C A G A G A G

G C A T C G C A G A G A G T A T A C A G T A C G  
 6  
 G C A G A G A G

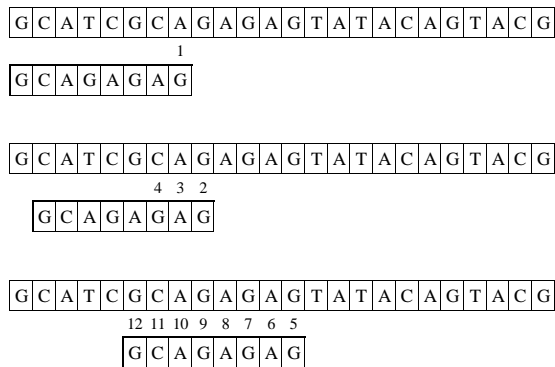
G C A T C G C A G A G A G T A T A C A G T A C G  
 7  
 G C A G A G A G

G C A T C G C A G A G A G T A T A C A G T A C G  
 8  
 G C A G A G A G

G C A T C G C A G A G A G T A T A C A G T A C G  
 9 10 11 12 13 14 15 16  
 G C A G A G A G

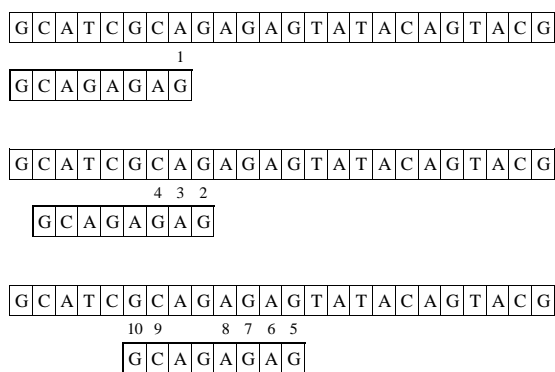
Dengan algoritma *Boyer-Moore* memerlukan pencocokan sebanyak 12 kali (sampai ditemukan pola)

Ilustrasinya sebagai berikut :



Dengan algoritma turbo *Boyer-Moore* memerlukan pencocokan sebanyak 10 kali (sampai ditemukan pola).

Ilustrasinya adalah sebagai berikut :



Dari hasil tersebut diperoleh

- Jika tiap pencocokan pola pemrosesan memerlukan waktu 0.0001 detik maka untuk melakukan pencocokan string pada pola tersebut dengan algoritma *Brute-Force* memerlukan waktu 0.0016 detik, dengan algoritma *Boyer-Moore* memerlukan waktu 0.0012 detik, dan dengan algoritma turbo *Boyer-Moore* memerlukan waktu 0.0010 detik.
- Jika pola yang dicocokkan hanya memiliki panjang 24 karakter hal tersebut tidak terlalu kelihatan perbedaannya. Namun jika teks yang dicocokkan sangat banyak maka hasilnya akan kelihatan. Misalkan dengan pola yang sama dicocokkan dengan teks yang sama namun terus berulan sampai 1000 kali. Hasil yang diperoleh adalah dengan algoritma *Brute-Force* memerlukan waktu 1.6 detik, dengan algoritma *Boyer-Moore* memerlukan waktu 1.2 detik, dan dengan algoritma turbo *Boyer-Moore* memerlukan waktu 1.0 detik.
- Jika kondisi tersebut dilakukan untuk banyak proses maka tinggal dikalikan dan akan terlihat

perbedaan dari waktu pencocokan dari setiap algoritma. Hal ini akan sangat mempengaruhi performansi dari komputer jika menggunakan algoritma *Brute-Force* dan juga KMP.

- Setelah ditemukan pola tersebut maka akan dilakukan penanganan dari sistem yang juga akan memakan waktu. Selain itu, dalam melakukan inisialisasi pola dan teks juga diperlukan waktu. Khusus untuk algoritma *Boyer-Moore* dalam membentuk tabel dari fungsi kemunculan terakhir diperlukan waktu juga. Sehingga waktu total dalam melakukan pengecekan apakah pola tersebut aman atau tidak adalah jumlah waktu inisialisasi awal, waktu pembentukan tabel (jika ada), waktu pencocokan pola, serta waktu pengambilan respon.

Dari penjelasan di atas, diketahui bahwa algoritma *Boyer-Moore* cukup baik digunakan dalam menjaga keamanan komputer dan masih memberikan performansi yang cukup baik. Dengan performansi tersebut, diharapkan respon dari sistem pada komputer dalam mengatasi berjalannya sebuah program bisa lebih cepat ditentukan. Jika tidak maka respon bisa berjalan dengan lambat.

Jika kita ingin memperbaiki peformansi bisa juga menggunakan turunan dari algoritma *Boyer-Moore* yang pada ilustrasi di atas melakukan 10 kali pencocokan karakter, memiliki selisih 2 lebih baik daripada algoritma *Boyer-Moore*. Jika pola dan teks yang ingin dicari memiliki pengulangan karakter (*suffix*) maka algoritma turbo *Boyer-Moore* bisa memberikan performansi lebih baik.

## V. SIMPULAN

Dari penjelasan pada bab sebelumnya, maka dapat diambil beberapa kesimpulan sebagai berikut

- Untuk menjaga keamanan komputer dapat dilakukan pencocokan string
- Algoritma pencocokan string yang memiliki performansi yang baik untuk keamanan komputer adalah algoritma *Boyer-Moore* dan turunannya.
- Algoritma turbo *Boyer-Moore* lebih baik dari algoritma *Boyer-Moore* untuk keamanan komputer dalam hal peformansi karena tidak perlu melakukan pemrosesan ekstra dan memungkinkan untuk mengingat potongan teks yang telah diperiksa sebelumnya.

## VI. REFERENSI

- [1] Munir, Rinaldi. 2009. "Diktat Kuliah IF3051 Strategi Algoritma". Institut Teknologi Bandung.
- [2] Lecroq, Thierry Charras, Christian. 2001. Handbook of Exact String Matching Algorithm
- [3] Gee, Alexander. 2009. Research into GPU accelerated pattern matching for applications in computer security. University of Canterbury, Christchurch. NewZealand

## VII. PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Desember 2010



Eric Cahya Lesmana 13508097