

# Aplikasi Pohon dalam Pencarian dan Penempatan Buku di Perpustakaan

Stefan Lauren - 13510034<sup>1</sup>  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
[13510034@std.stei.itb.ac.id](mailto:13510034@std.stei.itb.ac.id)

**Abstract**—Makalah ini akan membahas tentang permasalahan yang sering dialami dalam sebuah perpustakaan yang belum modern atau dengan kata lain tidak menggunakan teknologi komputer untuk mencari dan menempatkan buku. Banyak terjadi bahwa konsumen menunggu terlalu lama untuk meminjam buku atau kasus lainnya adalah petugas perpustakaan yang kesulitan dalam menempatkan buku kembali ke asalnya setelah ada konsumen yang meminjam. Untuk mengatasi kasus seperti ini, dapat digunakan konsep Pohon dalam Matematika Diskrit. Tujuannya sangat jelas, yaitu memberikan saran dan metode yang tepat untuk menyusun buku agar pencarian dan penempatan sebuah buku dapat berjalan dengan efektif dan efisien.

**Kata kunci**—*Binary Search Tree (BST)*, Kunci, Indeks Buku, Klasifikasi Buku.

## I. PENDAHULUAN

Pohon merupakan konsep graf yang paling penting. Banyak aplikasi dan terapan yang dapat dikembangkan dengan teori Pohon. Pohon itu sendiri merupakan graf yang khusus, yaitu graf tak-berarah terhubung yang tidak mengandung sirkuit. Salah satu terapannya adalah pada sistem peminjama buku dalam perpustakaan. Dalam hal ini, akan digunakan jenis pohon yang lebih khusus lagi yaitu Pohon Pencarian Biner / *Binary Search Tree*(BST).

Permasalahan yang akan dibahas adalah cara untuk mencari buku dengan judul, nama pengarang, dan klasifikasi yang spesifik dengan cepat dan tepat. Permasalahan lainnya adalah cara menyisipkan atau menempatkan buku baru ke dalam susunan buku – buku yang sudah terurut.

Dari banyak cara yang bisa ditemukan, metode dengan menggunakan konsep BST dianggap paling layak dipakai diukur dari strategi pencarian dan penempatan yang efektif dan efisien.

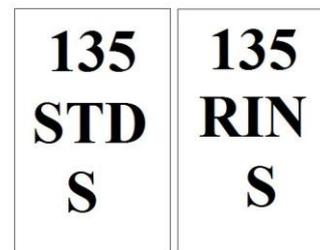
Dari kedua masalah tersebut, akan dicari solusi yang tepat dan akan dijelaskan tahap – tahap yang seharusnya dilakukan. Dengan BST diharapkan seluruh perpustakaan akan menerapkan metode ini demi kelancaran proses pencarian dan penempatan buku.

## II. KLASIFIKASI BUKU

Beberapa perpustakaan yang baik kondisinya akan memakai cara yang sama untuk membedakan sebuah buku dengan buku yang lainnya. Caranya adalah dengan memberikan kode tertentu dari buku yang ada. Kode – kode tersebut adalah

1. Nomor entri  
Nomor entri menandakan nomor urut dari sebuah buku, biasanya diwakilkan dengan 3 angka, tetapi dalam beberapa kasus dapat menjadi 3 angka dengan 1 angka di belakang koma.
2. Tajuk entri  
Tajuk entri dituliskan dengan 3 huruf kapital yang dapat merepresentasikan 3 huruf pertama dari judul buku, nama pengarang, atau penerbit.
3. Huruf judul  
Huruf judul direpresentasikan dengan satu huruf yang menunjukkan huruf awal dari sebuah judul buku.

Ketiga kode ini akan memberikan sebuah nomor klasifikasi pada sebuah buku.



Gambar 1. Contoh nomor klasifikasi

Dari gambar di atas, dapat diambil informasi bahwa buku kiri merupakan buku dengan nomor entri 135, tajuk entri STD, dan huruf judul S. Dalam kasus ini, tajuk entri kemungkinan besar merupakan judul buku karena huruf judul S sama dengan huruf pertama dari tajuk entri STD. Buku kanan mempunyai nomor entri dan huruf judul yang sama dengan buku kiri, tetapi tajuk entri yang berbeda.

Proses penyusunan buku dapat dilakukan dengan mengikuti langkah – langkah berikut

1. Bandingkan nomor entri. Nomor entri yang lebih kecil diletakkan di sebelah kiri.

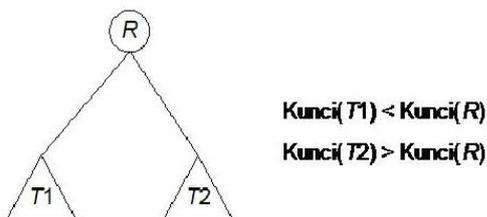
2. Jika nomor entri sama, lihat tajuk entri. Huruf yang mendekati huruf 'A' diletakkan di sebelah kiri.
3. Jika nomor entri dan tajuk entri sama, bandingkan huruf judul. Huruf judul yang mendekati huruf 'A' diletakkan di sebelah kiri.

### III. BINARY SEARCH TREE

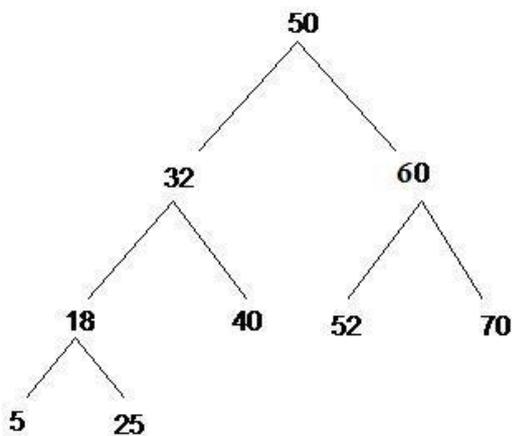
Pohon pencarian biner / *binary search tree* (BST) merupakan struktur pohon yang sering dipakai untuk pemecahan masalah pencarian dan penyisipan. Simpul pada pohon pencarian dapat berupa kunci pada data *record* maupun data itu sendiri. Kunci adalah nilai yang membedakan setiap simpul dengan simpul yang lainnya. Kunci harus unik, yang artinya tidak ada dua buah simpul atau lebih yang mempunyai kunci yang sama.

Pohon pencarian biner adalah pohon biner yang setiap kuncinya diatur dalam suatu urutan tertentu. Ketentuan pengaturan kunci adalah jika R merupakan akar, dan semua kunci yang tersimpan pada setiap simpul tidak ada yang sama, maka :

- a. Semua simpul pada upapohon kiri mempunyai kunci yang lebih kecil dari kunci(R)
- b. Semua simpul pada upapohon kanan mempunyai kunci nilai lebih besar dari kunci(R)



Gambar 2. Skema pohon pencarian



Gambar 3. Contoh pohon pencarian

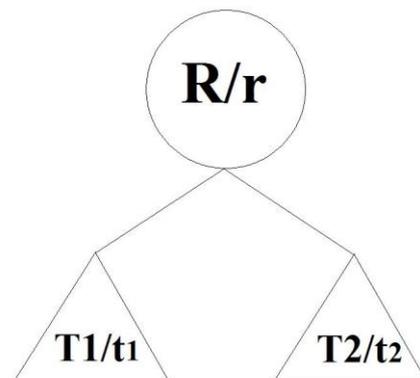
Dari gambar di atas dapat dibuktikan kebenarannya bahwa Gambar 3 menunjukkan sebuah pohon pencarian dengan Kunci utama merupakan akar pohon itu, yaitu 50. Harus diperhatikan juga bahwa untuk setiap upapohon

yang terbentuk, upapohon itu juga membentuk sebuah pohon pencarian.

### IV. MODIFIED BINARY SEARCH TREE

Untuk membantu memecahkan masalah yang ada dalam perpustakaan, pohon pencarian yang seperti biasa tidak akan bisa untuk diterapkan secara efektif. Pohon yang dibutuhkan sekarang adalah pohon pencarian yang kuncinya dapat tidak unik. Oleh sebab itu, modifikasi pertama akan dilakukan pada model pohon. Dalam model pohon hasil modifikasi, dimungkinkan bahwa terdapat pohon pencarian di dalam pohon pencarian. Setelah ini, pohon pertama yang muncul akan disebut sebagai pohon pencarian induk, pohon pencarian yang terdapat di dalam pohon pencarian utama akan disebut pohon pencarian anak pertama, dan pohon pencarian yang terdapat di dalam pohon pencarian anak pertama disebut pohon pencarian anak kedua. Pohon pencarian anak pertama akan memiliki skema pohon yang sama dengan pohon pencarian induk; sedangkan pohon pencarian anak kedua akan memiliki skema yang sama dengan pohon pencarian biasa.

Modifikasi selanjutnya akan dilakukan pada kunci. Data dari kunci seperti contoh pada gambar 3 hanya mengandung satu informasi, yaitu suatu besaran integer. Dalam nomor klasifikasi, yang sebanding dengan besaran integer tersebut adalah nomor entri. Tetapi, sangat memungkinkan bahwa suatu perpustakaan mempunyai buku dengan nomor entri yang sama lebih dari satu. Jadi, selain nomor entri, data pada kunci pohon pencarian induk dan anak(jika perlu) harus mengandung unsur banyak buku dengan nomor entri yang sama. Jika buku hanya ada satu, maka satu tidak dituliskan.

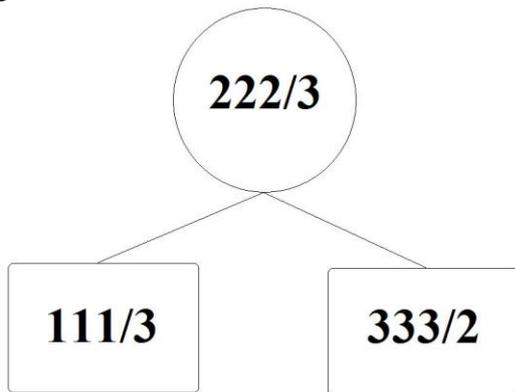


Gambar 4. Skema pohon pencarian induk

Deskripsi : R, T1, dan T2 merupakan data dari kunci dan r,t1, dan t2 adalah banyaknya R, T1, dan T2 secara berurutan. Di dalam pohon ini juga masih berlaku syarat pohon pencarian, yaitu  $Kunci(T1) < Kunci(R)$  dan  $Kunci(T2) > Kunci(R)$ .

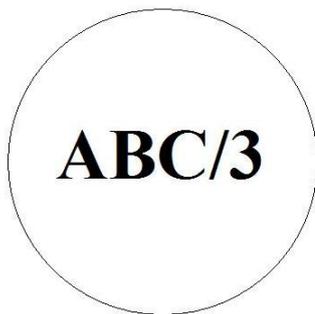
Sebagai contoh; buku – buku akan disusun di suatu rak dalam perpustakaan. Daftar buku – buku tersebut dalam format nomor entri\_tajuk entri\_huruf judul adalah 111\_ABC\_A, 111\_ABC\_A, 111\_ABC\_B, 222\_DEF\_D,

222\_EFG\_E, 222\_FGH\_F, 333\_IJK\_I, 333\_JKL\_J. Dari daftar tersebut, dapat dibuat struktur pohon pencarian sebagai berikut



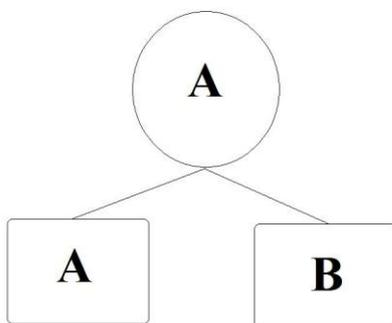
Gambar 5. Pohon pencarian induk

Keterangan : Pohon pencarian induk dibuat berdasarkan nomor entri. Nomor entri dibandingkan dengan operasi lebih besar '>' atau lebih kecil '<' yang diterapkan pada nomor entri yang dapat dianggap sebagai suatu integer.



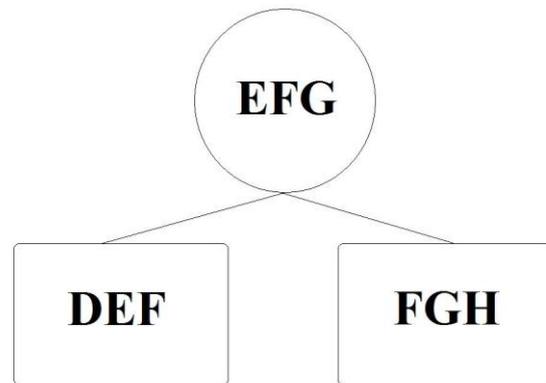
Gambar 6. Pohon pencarian anak pertama untuk 111/3

Keterangan : Pohon pencarian anak pertama untuk kunci 111/3 dari pohon pencarian induk. Pohon pencarian anak pertama didapatkan dengan membandingkan tajuk entri dari buku – buku sesuai dengan urutan abjad.

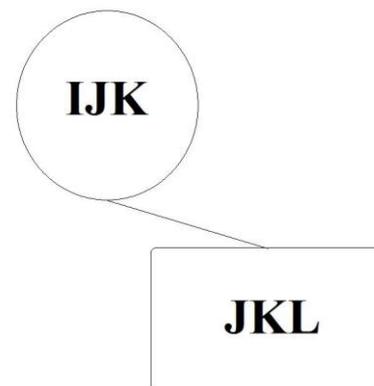


Gambar 7. Pohon pencarian anak kedua untuk 111/3

Keterangan : Pohon pencarian anak kedua untuk kunci 111/3 dari pohon pencarian induk. Pohon pencarian anak kedua didapatkan dengan membandingkan huruf judul dari nomor klasifikasi sesuai dengan urutan abjad.



Gambar 8. Pohon pencarian anak pertama untuk 222/3  
Keterangan : Pohon pencarian anak pertama untuk kunci 222/3 dari pohon pencarian induk. Pada kasus ini pohon pencarian anak kedua tidak dibutuhkan karena huruf judul dari nomor entri 222/3 sama semua.



Gambar 9. Pohon pencarian anak pertama untuk 333/3  
Keterangan : Pohon pencarian anak pertama untuk kunci 333/3 dari pohon pencarian induk. Pada kasus ini, pohon pencarian anak kedua tidak dibutuhkan walaupun huruf judulnya berbeda karena jumlah buku hanya satu.

## V. SEARCHING

Proses pencarian sebuah buku akan lebih mudah dengan struktur pohon pada bagian IV. Dengan model pohon pencarian yang baru, proses pencarian dapat menjadi lebih efisien karena dalam prosesnya tidak akan semua elemen dalam pohon tersebut akan diperiksa. Agar lebih mudah dimengerti, akan diberikan sebuah ilustrasi.

Misalkan daftar buku yang sudah disusun seperti pada bagian IV gambar 5 – 9. Seorang konsumen ingin meminjam buku dengan nomor klasifikasi 111\_ABC\_B. Pertama proses pencarian akan membandingkan nomor entri buku yang akan dipinjam(111) dengan kunci pertama pada pohon pencarian utama(222). Karena 111 lebih kecil dibandingkan dengan 222, maka proses pencarian akan berlanjut ke upapohon pencarian utama bagian kiri. Sekarang akan dibandingkan 111 dari buku pinjaman konsumen dan 111 dari kunci upapohon kiri. Karena 111 sudah sama dengan 111, maka proses pencarian berlanjut ke pohon pencarian anak pertama dari kunci 111/3 (gambar 6).

Pada pohon pencarian anak pertama sudah dipastikan bahwa tajuk entri dari buku pesanan konsumen sama

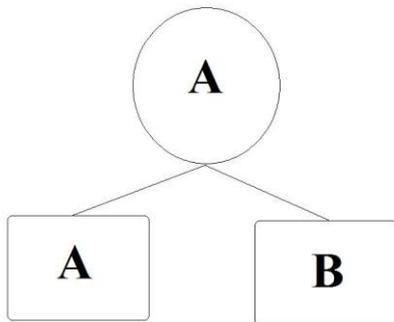
dengan ABC, oleh sebab itu pencarian langsung berlanjut ke pohon pencarian anak kedua. Pada pohon pencarian anak kedua akan dibandingkan huruf judul buku pesanan (B) dengan kunci pertama dari pohon pencarian anak kedua(A). 'B' merupakan abjad setelah 'A' sehingga pencarian dilanjutkan pada upapohon kanan dari pohon pencarian anak kedua. Pada titik inilah buku konsumen yang diinginkan ditemukan karena kunci dari upapohon kanan(B) sama dengan buku pesanan(B).

Dari ilustrasi di atas terlihat dengan jelas alasan proses pencarian dengan pohon pencarian menjadi sangat efisien. Sepanjang proses pencarian, hanya dilakukan beberapa proses perbandingan. Jumlah perbandingan yang dilakukan pada struktur pohon jauh lebih sedikit jika diharuskan mencari buku yang sama dengan tidak memakai struktur pohon (contoh: struktur daftar linier biasa).

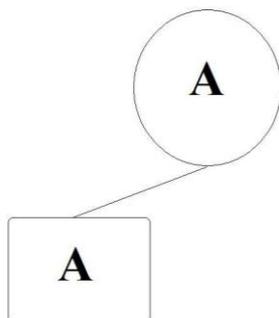
## VI. DELETING

Proses penghapusan dapat dianggap sebagai pengambilan buku yang akan dipinjam konsumen. Proses penghapusan merupakan proses yang cukup kompleks karena pohon setelah penghapusan harus tetap membentuk pohon pencarian. Agar lebih mudah dimengerti, akan diberikan dua buah ilustrasi.

Ilustrasi pertama akan mengambil kasus yang sama dengan ilustrasi sebelumnya pada bagian *searching*. Setelah proses pencarian selesai, diketahui letak buku yang ingin diambil untuk diberikan pada konsumen, pada hal ini adalah buku dengan nomor klasifikasi 111\_ABC\_B. Pada kasus ini proses penghapusan menjadi mudah, karena yang dicari terletak pada kunci paling bawah dari pohon pencarian. Berikut akan diperlihatkan kondisi awal dan akhir dari proses penghapusan :

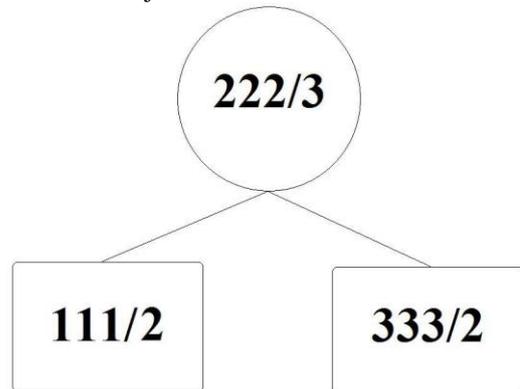


Gambar 10. Kondisi sebelum penghapusan

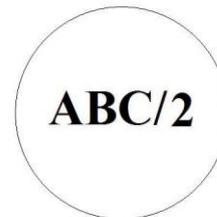


Gambar 11. Kondisi setelah penghapusan

Karena kunci 'B' pada pohon pencarian anak kedua tidak mempunyai upapohon, maka penghapusan dapat dilakukan secara langsung. Perubahan lainnya terdapat pada pohon pencarian utama dan pohon pencarian anak pertama karena jumlah buku berkurang. Pohon – pohon tersebut akan menjadi :



Gambar 12. Pohon pencarian utama setelah penghapusan



Gambar 13. Pohon pencarian anak pertama setelah penghapusan.

Ilustrasi selanjutnya berbeda dengan kasus sebelumnya. Anggaplah terdapat konsumen lain yang akan meminjam setelah konsumen pertama mendapatkan bukunya(proses penghapusan telah terjadi). Konsumen kedua ini akan meminjam buku dengan nomor klasifikasi 111\_ABC\_A. Setelah melalui proses pencarian, didapatkan bahwa buku tersebut ada pada pohon pencarian anak kedua yang sama seperti pada gambar 11. Karena buku 111\_ABC\_A terdapat dua buah, akan diambil buku yang paling pertama ditemukan, yaitu buku dengan kunci paling atas. Untuk memudahkan penjelasan secara visual, bentuk lingkaran dan kotak akan tetap seperti itu. Jika buku dengan lingkaran diambil, maka ada yang harus menggantikannya sebagai kunci pertama dari pohon pencarian anak kedua tersebut. Proses yang dilakukan adalah mencari buku – buku dari upapohonnya yang dapat menggantikannya sehingga hasil akhirnya merupakan pohon pencarian kembali. Dalam kasus ini, buku yang tepat adalah buku yang terdapat dalam kotak. Maka pohon pencarian anak kedua akan menjadi :



Gambar 14. Hasil akhir pohon pencarian anak kedua setelah penghapusan.

Proses penghapusan dengan menggunakan struktur pohon terlihat lebih kompleks jika dibandingkan dengan proses

pencarian ataupun penyisipan yang akan dijelaskan selanjutnya. Akan tetapi, proses ini masih dapat dikatakan unggul dibandingkan dengan proses penghapusan dengan struktur lain karena dalam menghapus diharuskan melakukan proses pencarian yang dalam hal ini dengan struktur pohon akan lebih efektif.

## VII. INSERTING

Proses penyisipan dalam suatu sistem perpustakaan dapat diilustrasikan dengan proses pengembalian suatu buku oleh konsumen dan petugas perpustakaan diharuskan untuk mengembalikan buku ke tempat yang tepat. Inti dari proses penyisipan adalah mencari (*searching*) tempat yang tepat, lalu menambahkannya ke dalam pohon pencarian. Syarat tempat yang tepat adalah saat buku baru disisipkan, bentuk pohon pencarian tidak berubah dengan arti seluruh peraturan pohon pencarian harus dipenuhi.

Agar lebih mudah dipahami, akan diberikan suatu ilustrasi. Misalkan konsumen yang sama seperti ilustrasi bagian V(*searching*) akan mengembalikan buku yang dipinjamnya. Kondisi sebelum buku disisipkan adalah sama dengan kondisi akhir proses *deleting*. Proses awal yang harus dicari adalah mencari buku yang paling mendekati buku yang dipinjam konsumen (111\_ABC\_B), yaitu 111\_ABC\_A. Proses pencarian agar mendapatkan 111\_ABC\_A sama dengan proses pencarian pada bagian V sehingga tidak akan dijelaskan ulang di sini. Setelah mendapat buku yang paling mirip, dilakukan proses perbandingan kembali. Karena antara 111\_ABC\_A dan 111\_ABC\_B yang berbeda adalah nomor halamannya, maka akan dibandingkan nomor halamannya. Karena 'B' merupakan abjad lebih kanan daripada 'A', maka buku 111\_ABC\_B harus diletakkan sebagai upapohon kanan dari pohon pencarian anak kedua. Setelah itu harus dibandingkan lagi dengan kunci dari upapohon kanan tersebut. Karena pada ilustrasi ini upapohon kanan tidak ada, maka dapat dengan langsung buku 111\_ABC\_B yang dikembalikan oleh konsumen tersebut ditempatkan sebagai upapohon kanan.

Pada proses *inserting* pun dengan menggunakan pohon akan menjadi efektif dan efisien dengan alasan yang sama, yaitu proses perbandingan tidak dilakukan dengan sia – sia. Dalam proses penyisipan, segala kemungkinan kasus dapat dilakukan dengan mudah tanpa hambatan.

## V. KESIMPULAN

Kesimpulan yang dapat diambil dari makalah ini adalah bahwa dengan menggunakan pohon, proses pencarian, penghapusan, dan penyisipan dapat dilakukan dengan mudah dan cepat. Jika proses ini diterapkan pada sistem peminjaman buku dalam perpustakaan, dapat dipastikan petugas perpustakaan dapat dengan cepat melayani konsumen. Hasil akhirnya adalah tidak ada lagi kasus salah penempatan buku, lama dalam mencari buku, dan konsumen yang tidak puas karena terlalu lama menunggu.

## REFERENCES

- [1] R. Munir, Diktat Kuliah IF2091 Struktur Diskrit. Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2008, pp. IX – 01 – V – 30.
- [2] [http://eprints.uny.ac.id/260/1/penyusunan\\_rak\\_perpustakaan.pdf](http://eprints.uny.ac.id/260/1/penyusunan_rak_perpustakaan.pdf), diakses pada 10 Desember 2011 pk.19.22.
- [3] <http://www.informatika.org/~rinaldi/Matdis/matdis.htm>, diakses pada 10 Desember 2011 pk.18.23
- [4] [http://en.wikipedia.org/wiki/Library\\_classification](http://en.wikipedia.org/wiki/Library_classification), diakses pada 10 Desember 2011 pk.19.14

## 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, 10 Desember 2011



Stefan Lauren  
13510034