

Penerapan Pengkodean Huffman dalam Pemampatan Data

Patrick Lumban Tobing

NIM 13510013

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

patrick_ltobing@hotmail.com

Abstrak—Makalah ini membahas mengenai penerapan kode Huffman dalam metode pemampatan data. Secara garis besar pembahasan yang dilakukan adalah mengenai pengertian dari kode Huffman sendiri, penjelasan mengenai beberapa tipe pemampatan data, contoh implementasi kode Huffman dalam suatu data dan penjabaran aplikasi kode Huffman dan perannya dalam pemampatan data saat ini. Pada intinya, kode Huffman ini adalah basis atau dasar dari hampir semua teknik pemampatan data yang ada sekarang. Secara singkat, kode Huffman ini memiliki metode yang sederhana, tetapi padat dan efektif dalam pemampatan data dari ukuran kecil hingga besar.

Kata kunci—kode Huffman, pemampatan data, implementasi, peran

I. PENDAHULUAN

Saat ini teknologi informasi berkembang sangat pesat. Perkembangannya bahkan bukan menurun tetapi bertambah seiring bertambahnya tahun dan generasi juga. Satu objek yang menjadi pusat perkembangan teknologi informasi adalah data. Jika kita melihat bagaimana data atau informasi dapat tersampaikan begitu cepat, seperti itulah perkembangan teknologi informasi masa kini. Contohnya saja, suatu bencana alam yang terjadi di belahan bumi yang lain misalkan benua Eropa dapat diketahui di belahan bumi yang lain misalkan benua Asia dengan begitu cepat melalui berbagai media informasi, baik itu berita di internet, televisi, jejaring sosial seperti facebook dan twitter, dll.

Perkembangan teknologi informasi yang begitu pesat ini tidak lepas dari peran perkembangan teknologi pengiriman dan penyaluran data, tepatnya dalam hal pemampatan/ kompresi data. Mengapa pemampatan data menjadi aktor utama dalam kelancaran kita membaca berita di internet, chatting, menerima atau mengirim email bahkan mengunduh lagu atau film? Hal ini dikarenakan teknologi informasi ini juga memiliki keterbatasan dalam hal ukuran, tepatnya ukuran data. Data yang ukurannya lebih besar akan membutuhkan waktu yang bisa jadi jauh lebih lama untuk dapat disalurkan dari pengirim ke penerima tergantung dari kualitas teknologi yang dimiliki oleh kedua pihak.

Teknologi pemampatan data ini memanjakan para

“konsumer” data sehingga mereka dapat “mengonsumsi” data dengan waktu yang cepat. Salah satu ilmu yang dipakai dalam pemampatan data adalah pengkodean Huffman. Pengkodean Huffman merupakan suatu algoritma pengkodean yang terutama digunakan untuk pemampatan data tipe lossless (untuk jenis-jenis pemampatan data akan dijelaskan nanti). Secara umum pengkodean Huffman dilakukan dengan mengikuti langkah-langkah khusus untuk merepresentasikan simbol-simbol yang akan dikodekan ke dalam suatu *prefix code* (kode awalan). Kode awalan sendiri merupakan suatu deretan bit yang merepresentasikan simbol tertentu, di mana deretan bit untuk simbol itu bukanlah awalan dari representasi deretan bit dari simbol-simbol lain yang akan dikodekan.

Secara umum, pengkodean Huffman ini secara teori akan mampu memampatkan data sedemikian rupa dengan langkah-langkah dan beberapa teori, seperti pendefinisian pemampatan data yang akan dijelaskan pada bab-bab berikutnya.

II. PENGERTIAN KODE HUFFMAN

DAN PEMAMPATAN DATA

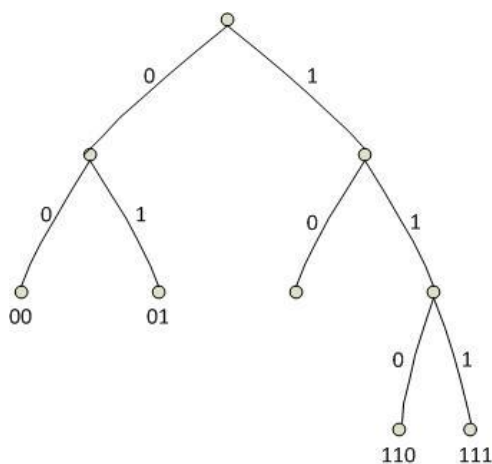
II.1 Kode Huffman

Seperti sudah dijabarkan sedikit pada bagian pendahuluan, pengkodean Huffman merupakan suatu algoritma yang dapat digunakan dalam ruang lingkup pemampatan data. Objek dari pengkodean Huffman sendiri ini adalah kode Huffman. Jadi, kode Huffman merupakan hasil dari suatu proses algoritma yang dinamakan pengkodean Huffman.

Mengulas sedikit sejarah dari kode Huffman ini. Pada tahun 1951, David A. Huffman membuat suatu makalah untuk tugas akhir yang mengulas masalah untuk menemukan kode biner paling efisien. Teori yang dikemukakan beliau pada tugas akhirnya didasarkan pada pohon biner frekuensi yang terurut. Teori ini akhirnya dinamakan kode Huffman dan menjadi salah satu basis dalam teknologi pemampatan data.

Seperti yang sudah dijelaskan juga secara sekilas pada bagian pendahuluan bahwa kode Huffman ini menghasilkan kode-kode awalan (*prefix code*) yang

merupakan representasi dari simbol-simbol yang akan dikodekan. Kode awalan ini banyak digunakan untuk mengirimkan pesan pada komunikasi data. Tiap simbol direpresentasikan dengan deretan bit (angka 0 dan 1). Pengiriman data dilakukan dengan mengirimkan representasi data dalam bentuk deretan bit. Deretan bit tersebut akan diuraikan oleh penerima menjadi simbol-simbol awal sebelum dikodekan. Setiap simbol dalam pengkodean ini tidak boleh mempunyai deretan bit yang menjadi awal deretan bit untuk simbol-simbol yang lain. Hal ini dilakukan untuk mencegah timbulnya kerancuan dalam menguraikannya kembali ke dalam simbol yang seharusnya. Sebagai contoh, deretan bit {00,01,10,111,110} adalah kode awalan, tetapi deretan bit {1110,111,10,11,0} bukanlah kode awalan karena 11 adalah awalan dari 111 serta 1110 dan 111 adalah awalan dari 1110. Kode awalan ini memiliki pohon biner yang bersesuaian. Setiap sisi kanan harus dilabeli 1 atau 0 saja, begitupula dengan sisi kirinya (kebalikan dari sisi kanan). Kode awalan dinyatakan oleh barisan dari sisi-sisi yang dilalui lintasan dari akar ke daun. Kode awalan yang dibentuk dituliskan pada daun. Untuk contoh deretan bit di atas yang merupakan kode awalan memiliki representasi pohon biner seperti pada Gambar1 di bawah ini.



Gambar1. Pohon biner untuk kode awalan {00,01,10,111,110}

II.2 Pemampatan Data

Secara umum, dalam bidang informatika, pemampatan data adalah proses pengkodean informasi dengan menggunakan jumlah bit yang lebih sedikit dibandingkan jumlah bit asli yang digunakan oleh simbol yang akan dikodekan tersebut. Pemampatan ini berguna untuk mengurangi penggunaan sumber daya, terutama yang memerlukan biaya mahal, seperti tempat penyimpanan data dan transmisi bandwidth. Di lain sisi, pemampatan data juga memiliki kekurangan, seperti misalnya sumber daya yang diperlukan untuk memampatkan dan menguraikan kembali data tertentu memerlukan sumber daya komputasi yang berlebih, gangguan-gangguan yang didapat jika menggunakan metode *lossy compression*, dan

pengorbanan beberapa faktor untuk perancangan metode pemampatan seperti derajat pemampatan.

Metode pemampatan data dapat dikelompokkan ke dalam dua kelompok besar, yaitu: metode lossless dan metode lossy. Pemampatan data tipe lossless merupakan suatu tipe dari algoritma pemampatan data yang memungkinkan data yang asli dapat disusun kembali dari data pemampatan. Pemampatan data lossless digunakan dalam berbagai aplikasi seperti format ZIP dan GZIP. Beberapa format gambar seperti GIF atau PNG hanya menggunakan pemampatan lossless, sedangkan tipe gambar lainnya menggunakan metode lossy maupun lossless. Metode lossless menghasilkan data hasil pemampatan yang identik dengan data aslinya. Hal ini diperlukan untuk beberapa tipe data, seperti dokumen teks dan *executable files*. Pemampatan tipe lossless ini pada intinya digunakan ketika suatu data yang dimampatkan harus berada dalam kondisi sama seperti sebelum dimampatkan saat diuraikan. Walaupun begitu rasio pemampatan dari metode ini sangat rendah karena memang dengan metode ini tidak akan ada sama sekali data yang hilang dari data aslinya. Kode Huffman adalah salah satu contoh algoritma yang digunakan pada metode lossless ini.

Metode yang kedua adalah metode lossy. Data yang diperoleh dengan metode lossy ini mungkin berbeda dari yang aslinya tetapi tetap memiliki kesamaan yang sangat dekat. Pemampatan tipe ini sering digunakan untuk pemampatan data multimedia, seperti audio dan gambar. Pemampatan ini mengalami generation loss dalam pengaplikasian formatnya. Generation loss ini menyebabkan hilangnya kualitas secara progresif apabila dilakukan pemampatan dan penguraian data secara berulang-ulang. Hal ini berbeda dengan metode lossless yang tidak mengalami penurunan kualitas apabila dilakukan pemampatan dan penguraian bahkan secara berulang-ulang. Sebagai gambaran, ketika seorang pengguna menerima dan menguraikan file yang dimampatkan dengan metode lossy, maka file tersebut akan sedikit berbeda di level bit. Perbedaan ini tidak akan terlihat atau terasa oleh mata dan telinga manusia (apabila file berupa audio atau gambar). Di lain sisi, metode ini dapat menghasilkan rasio kompresi yang jauh lebih besar daripada metode lossless. Contoh file hasil metode lossy adalah JPEG dan MPEG.

III. IMPLEMENTASI KODE HUFFMAN DAN VARIASINYA

III.1 Metode Implementasi Kode Huffman

Untuk menjalankan pengkodean Huffman ini memerlukan beberapa langkah-langkah yang harus dilakukan sampai mendapatkan suatu kode Huffman. Berikut adalah langkah-langkahnya:

1. Hitung frekuensi/kekerapan kemunculan tiap

simbol pada data yang akan dimampatkan.

2. Lakukan pembentukan pohon biner, yang dinamakan pohon Huffman, caranya adalah sebagai berikut:

- Pilihlah dua simbol dengan peluang kemunculan paling kecil.
- Kombinasikan kedua simbol itu menjadi simpul orangtua dari simbol-simbol itu sendiri dengan peluang kemunculannya adalah jumlah peluang kedua anaknya.
- Simbol baru hasil gabungan kedua simbol awal ini menjadi suatu simpul baru yang akan dikombinasikan dengan simbol selanjutnya yang memiliki peluang terkecil.
- Pilih dua simbol selanjutnya, termasuk simbol yang baru, yang mempunyai peluang terkecil.
- Langkah-langkah yang sama dilakukan untuk simbol-simbol selanjutnya sampai simbol terakhir.
- Setelah semua simbol selesai dilibatkan dalam langkah-langkah di atas maka pohon Huffman telah selesai dibuat.

Setiap daun pada pohon Huffman menyatakan simbol yang digunakan dalam data tersebut. Kodekan setiap simbol dengan memberi label 0 atau 1 pada setiap cabang (sisi) kiri dan label 1 atau 0 pada setiap cabang (sisi) kanan. Agar dapat menghasilkan kode untuk setiap simbol, buatlah lintasan dari akar ke daun dan tuliskan deretan bit yang didapat pada setiap daun. Kode Huffman untuk setiap simbol adalah deretan bit yang tertulis pada setiap daun yang merepresentasikan simbol tersebut.

Terlihat dari kode Huffman yang terbentuk nanti bahwa simbol yang paling sering muncul memiliki kode dengan jumlah bit paling sedikit. Sesuai dengan definisi di bab II di atas bahwa tidak akan ada simbol yang kodenya merupakan kode awalan untuk simbol yang lain. Kode Huffman ini juga tidak bersifat unik, maksudnya adalah kode untuk setiap simbol berbeda-beda pada setiap pesan tergantung pada kekerapan munculnya simbol-simbol tersebut pada data yang akan dimampatkan.

III.2 Implementasi Kode Huffman pada Suatu Teks

Pada subbab berikut ini akan diberikan satu contoh implementasi kode Huffman pada data yang merupakan suatu teks yang berisi “institutteknologibandung”. Seperti yang telah dijabarkan pada subbab sebelumnya, untuk mendapatkan kode Huffman dari suatu data perlu untuk mendaftarkan kekerapan muncul tiap simbol dalam suatu data untuk memudahkan proses selanjutnya. Oleh karena itu di bawah ini diberikan Tabel1 yang berisi kekerapan muncul tiap simbol pada teks “institutteknologibandung”.

Simbol	Kekerapan	Peluang
i	3	3/24
n	4	4/24
s	1	1/24
t	4	4/24
u	2	2/24
e	1	1/24
k	1	1/24
o	2	2/24
l	1	1/24
g	2	2/24
b	1	1/24
a	1	1/24
d	1	1/24

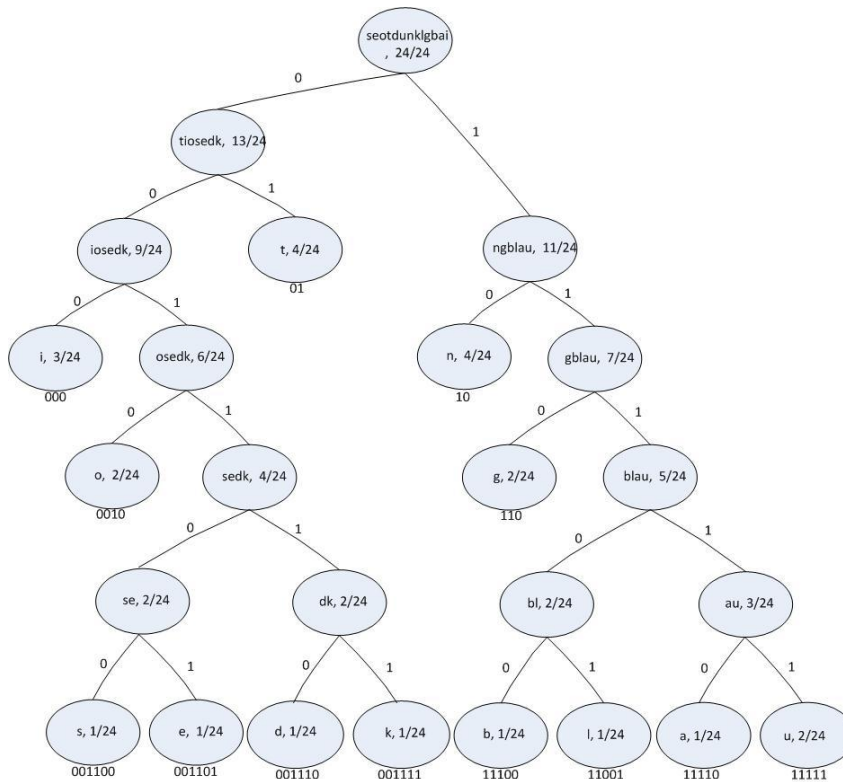
Tabel 1. Tabel kekerapan kemunculan simbol untuk pesan “institutteknologibandung”

Setelah melakukan langkah pertama, yaitu mendaftarkan semua simbol beserta kekerapan muncul dan peluang muncul dalam suatu data, maka dilakukan pembuatan pohon Huffman berdasarkan tabel kekerapan yang telah dibuat tersebut. Pohon Huffman yang telah dibuat untuk teks “institutteknologibandung” dapat dilihat pada Gambar2 di bawah ini (karena tempat yang tidak cukup, gambar ditaruh di halaman berikutnya). Pada pembuatan pohon Huffman ini, yang perlu diperhatikan adalah memulai penggabungan simpul menjadi satu simpul baru dari bawah dan dari simbol dengan kekerapan terkecil yang ada seperti langkah-langkah yang sudah dijelaskan di atas. Jika langkah itu dilakukan terus hingga simbol/kumpulan simbol terakhir dengan kekerapan paling besar maka akan didapatkan pohon Huffman yang diinginkan.

Dari pohon Huffman yang telah dibuat, dapat didaftarkan pula semua simbol yang ada beserta kode Huffman yang dimiliki oleh tiap simbol. Secara langsung dapat dilihat kode Huffman untuk tiap simbol tertulis di bawah setiap daun yang merepresentasikan simbol tersebut. Secara lengkap kode Huffman untuk masing-masing simbol dapat dilihat pada Tabel 2 di bawah ini.

Simbol	Kekerapan	Peluang	Kode Huffman
i	3	3/24	000
n	4	4/24	10
s	1	1/24	001100
t	4	4/24	01
u	2	2/24	11111
e	1	1/24	001101
k	1	1/24	001111
o	2	2/24	0010
l	1	1/24	11001
g	2	2/24	110
b	1	1/24	11100
a	1	1/24	11110
d	1	1/24	001110

Tabel2. Tabel kekerapan dan kode Huffman untuk string “institutteknologibandung”



Gambar 2. Pohon Huffman untuk teks "institutteknologibandung"

Dengan menggunakan kode Huffman yang telah dirancang, maka teks "institutteknologibandung" dapat direpresentasikan menjadi deretan bit:

0001000110001000011111101010011010011111000101

10010010110000111001111010001110111110110

Jika dijumlahkan, maka total bit yang digunakan dalam data yang telah dimampatkan ini adalah 88 bit.

Sekarang akan diperbandingkan ukuran dari data setelah dimampatkan dengan sebelum dimampatkan. Untuk mengetahui ukuran asli dari data yang berupa teks "institutteknologibandung", maka setiap simbol/karakter dari teks ini akan dilihat deretan bitnya berdasarkan kode ASCII (setiap karakter dikodekan ke dalam 8 bit biner). Di bawah ini adalah Tabel 3 yang mendaftarkan simbol-simbol yang ada pada data beserta kode ASCII-nya.

Simbol	Kode ASCII (8 bit biner)
i	01101001
n	01101110
s	01110011
t	01110100
u	01110101
e	01100101
k	01101011
o	01101111
l	01101100
g	01101111
b	01100010
a	01100001
d	01100100

Tabel 3. Tabel kode Huffman untuk string "institutteknologibandung"

Berdasarkan kode ASCII tiap simbol didapatkan deretan bit sebagai berikut:

0110100101101110011100110111010001101001011101
0001110101011101000111010001100101011010110110
1110011011110110110001101111011001110110100101
1000100110000101101110011001000111010101101110
01100111

Jika dijumlahkan, maka total bit yang digunakan oleh data asli jika menggunakan pengkodean ASCII adalah $24 \times 8 = 192$ bit (24 byte).

Dapat dilihat dari total bit yang digunakan oleh data yang telah dimampatkan dengan metode pengkodean Huffman dengan data asli dengan pengkodean ASCII memiliki perbedaan yang cukup "jauh". Selisih antara keduanya adalah 104 bit. Berarti metode pemampatan dengan kode Huffman ini telah menghemat memori sekitar 54,17% dari ukuran data yang asli. Dalam skala kecil seperti data yang berisi teks "institutteknologibandung" ini memang tidak begitu terasa efek dari pemampatannya. Tetapi dalam skala yang lebih besar dengan penggunaan memori yang besar, metode pemampatan data, terutama dengan metode Huffman ini tentunya akan memberi efek yang cukup besar dan juga tentunya menguntungkan dari pihak pengirim dan penerima.

III.3 Aplikasi dan Peran Kode Huffman dalam Teknik Pemampatan Data

Aplikasi kode Huffman sangat banyak dalam bidang ilmu komputer. Algoritma dari pengkodean Huffman tidak hanya terbatas pada pemampatan data yang berbentuk teks saja. Pada implementasinya, kode Huffman dapat digunakan untuk memampatkan data yang berbentuk gambar, suara dan jenis data lainnya, seperti MP3, data

ZIP, JPEG, dll. Memang pada pemampatan tipe data JPEG metode utama pemampatannya menggunakan transformasi cosinus diskrit, tetapi pengkodean Huffman tetap mengambil peran kecil dalam pembentukan data dengan pemampatan tipe JPEG. Selain kode Huffman, masih banyak lagi teknik pemampatan data yang lainnya. Walaupun begitu, kode Huffman adalah salah satu teknik pemampatan data yang menawarkan kesederhanaan bentuk dari pemampatan data. Beberapa teknik lainnya bahkan butuh penguasaan kode Huffman yang baik untuk dapat mengaplikasikannya, seperti pengkodean aritmatik dan pengkodean Lempel-Ziv-Welch. Pada intinya, kode Huffman dapat digunakan secara efektif untuk mendapatkan kode yang singkat dan padat untuk merepresentasikan deretan yang panjang dari beberapa byte-byte (8 bit, tiap simbol/karakter memiliki kode byte yang berbeda) yang berbeda.

Pada aplikasinya, pengkodean Huffman dengan membuat program yang menggunakan metode pengkodean yang baku berdasarkan kekerapan muncul tiap simbol (misalkan dalam suatu teks) cukup jarang digunakan. Satu alternatif dari kode Huffman yang lebih sering digunakan adalah *adaptive Huffman coding*. *Adaptive Huffman coding* ini merupakan salah satu *adaptive coding* yang berbasis kode Huffman. *Adaptive coding* merupakan salah satu metode pengkodean untuk pemampatan data tipe *lossless*. Teknik ini cocok untuk data yang dinamis karena dia dapat menyesuaikan pengkodean dengan perubahan karakteristik dari suatu data dan data tersebut tidak perlu dialirkan terlebih dahulu untuk dilihat dan dihitung model peluangnya. Biaya untuk keuntungan dari tipe ini adalah *encoder* dan *decoder* harus lebih kompleks dan tetap menjaga *state* mereka masing-masing tersinkronisasi, dan dibutuhkan kemampuan komputasi yang lebih baik untuk menjaga *state* dari *encoder* dan *decoder* tersebut. Salah satu implementasi dari *adaptive Huffman coding* ini adalah *Faller-Gallager-Knuth and Vitter algorithm*.

Pada masa kini, seiring berkembangnya teknologi dan kemampuan manusia dalam merancang algoritma-algoritma yang lebih efektif dan efisien, kode Huffman pada sebagian besar metode pemampatan data tidak lagi menjadi pemeran utama. Kode Huffman lebih menjadi dasar dari pengembangan algoritma-algoritma lain yang lebih efisien untuk teknik pemampatan data. Seperti misalnya, metode DEFLATE (Algoritma PKZIP) yang berperan sebagai front-end untuk mengkodekan rangkaian simbol yang kembar, lalu kemudia kode Huffman digunakan sebagai back-end setelah data diproses untuk mengkodekan sisa simbol yang tidak kembar.

IV. KESIMPULAN

Pengkodean Huffman merupakan salah satu algoritma “tertua” dalam bidang pemampatan data. Kode Huffman sendiri dapat memampatkan data secara efektif untuk mendapatkan deretan bit yang lebih pendek dari yang asli. Seiring berkembangnya teknologi, pengkodean Huffman secara tradisional tidak lagi secara murni diterapkan. Muncul banyak metode-metode lain yang mendasarkan

tekniknya pada pengkodean Huffman untuk menghasilkan proses pemampatan data yang lebih efektif dan efisien.

V. UCAPAN TERIMA KASIH

Terima kasih penulis ucapkan terutama kepada Bpk. Rinaldi Munir dan Ibu Harlili atas jasa pengajarannya selama satu semester ini. Pengajaran Bapak dan Ibu membukakan mata saya akan bidang teknologi informasi dan menyadarkan saya bahwa masih banyak hal “di dalam sana” yang perlu dikaji dan dikuasai untuk dapat berhasil di bidang teknologi informasi.

DAFTAR PUSTAKA

- [1] Rinaldi, Munir. *Diktat Kuliah IF2091 Struktur Diskrit*. Bandung: Institut Teknologi Bandung. 2008.
- [2] http://en.wikipedia.org/wiki/Adaptive_Huffman_coding waktu akses: 11 Desember 2011
- [3] http://en.wikipedia.org/wiki/Data_compression waktu akses: 9 Desember 2011
- [4] http://en.wikipedia.org/wiki/Huffman_coding waktu akses: 9 Desember 2011
- [5] <http://michael.dipperstein.com/huffman/> waktu akses: 10 Desember 2011

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, 11 Desember 2011



Patrick Lumban Tobing
NIM 13510013