

PENCARIAN MAKSIMUM CLIQUE DALAM GRAF DENGAN MENGGUNAKAN ALGORITMA BRANCH AND BOUND

Nur Adi Susliawan Dwi Caksono/13508081
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
lf18081@students.if.itb.ac.id

Abstract

Permasalahan *clique* merupakan sebuah permasalahan graf NP-complete. Clique adalah sebuah upagraf lengkap yang diperoleh dari suatu graf yang terdiri atas tiga atau lebih simpul (vertex). Setiap simpul tersebut saling berdekatan satu sama lain dengan simpul lainnya.

Pencarian maksimum clique telah diterapkan dalam sejumlah bidang keilmuan, keahlian teknik dan bahkan dalam bidang bisnis. Beberapa penerapan clique bahkan mencapai bidang biologi meliputi solusi permasalahan struktur molekul DNA. Clique juga digunakan dalam pemrosesan citra dalam pengaturan jarak jauh, pencocokan titik koordinat dalam system informasi, permasalahan partisi data dalam kepingan memory dan lain-lain.

Permasalahan clique ini merupakan permasalahan pencarian solusi yang dianggap cukup sulit, sama seperti permasalahan knapsack, pewarnaan graf, sirkuit Hamilton, n-puzzle dan Travelling Sales Person. Hingga sekarang belum ditemukan algoritma yang benar-benar mangkus untuk penyelesaian perma-permasalahan tersebut meskipun untuk beberapa kasus telah ditemukan algoritma solusi yang mendekati hasil optimal.

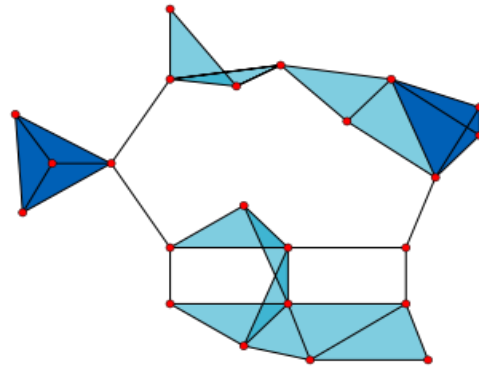
Pencarian maksimum untuk permasalahan clique ini dapat dilakukan dengan menggunakan beberapa metode seperti dengan menggunakan beberapa metode seperti dengan penggunaan algoritma brute force, algoritma backtracking dan algoritma branch and bound. Walaupun sekarang banyak variasi dari penyelesaian dari algoritma pencarian, konsep dasar dari variasi-variasi ini berdasar pada ketiga algoritma diatas.

Kata kunci : clique, branch dan bound, graf, pewarnaan

I. PENDAHULUAN

Clique adalah himpunan bagian dari graf atau biasa disebut upagraf yang terdiri atas 3 atau lebih simpul yang saling terhubung. Misalkan terdapat graf tidak berarah

$G = (V, E)$, clique adalah kumpulan simpul C yang merupakan bagian dari C dimana untuk setiap 2 simpul di C , terdapat sebuah sisi yang menghubungkannya. Clique merupakan salah satu konsep dasar dari teori graf dan digunakan di banyak masalah dan konstruksi matematika dalam graf.



Gambar 1. Graf berarah

Untuk lebih memahami apa itu clique, Anda dapat melihat gambar1 diatas. Dari gambar 1, kita dapat melihat sebuah graf tidak berarah dengan jumlah simpul 23. Seperti yang telah dijelaskan clique adalah himpunan bagian dari sebuah graf yang terdiri dari 2 simpul atau lebih. N-simpul clique adalah kumpulan himpunan bagian dari sebuah graf tidak berarah yang memiliki jumlah N simpul yang saling berhubungan. Jadi pada gambar 1, graf tidak berarah tersebut memiliki 23 buah 1-simpul clique (simpul-simpulnya sendiri), 42 buah 2-simpul clique (2 buah simpul yang dihubungkan dengan sebuah sisi), 19 buah 3-simpul clique (segitiga yang berwarna biru muda). Dan 2 4-simpul clique (bagan yang berwarna biru tua). Perlu diketahui, syarat sebuah upagraf menjadi clique dalam sebuah graf tidak berarah ialah setiap simpul pada sebuah N-simpul clique memiliki N-1 jumlah sisi yang saling terhubung dengan simpul tetangga clique-nya. Misalkan pada gambar 1, setiap simpul pada 4-simpul clique memiliki 3 buah sisi yang saling terhubung dengan simpul tetangga clique-nya.

Clique dikatakan maksimum ketika clique tersebut tidak dapat diperpanjang dengan memasukkan satu simpul yang lebih berdekatan dengannya. Pada gambar 1, clique maksimal dari graf tidak berarah tersebut ialah bernilai 4.

Masalah clique maksimum adalah masalah untuk menemukan clique dengan jumlah simpul terbesar dari sebuah graf tidak berarah. Algoritma untuk menemukan clique maksimum digunakan di bidang kimia, bioinformatics, dan aplikasi komputasi biologi dimana aplikasi tersebut untuk mencari kesamaan antar molekul. Algoritma ini digunakan untuk menyaring molekul dalam suatu senyawa yang memiliki kesamaan dengan molekul aktif (molekul yang mudah bersenyawa dengan molekul lain). Algoritma ini juga digunakan untuk membandingkan struktur protein untuk menyediakan informasi tentang fungsi protein.

Mencari clique maksimum dalam aplikasi ini seringkali bersifat bottle-neck (query yang masuk banyak, sedangkan pintu masuk query tersebut kecil sehingga mengharuskan query untuk mengantri). Masalah maksimum clique merupakan masalah NP-hard. Algoritma exact yang dapat menjamin untuk menemukan clique maksimum ialah algoritma branch and bound yang akan mencari secara sistematis melalui solusi-solusi yang mungkin dan menggunakan bound untuk membatasi ruang pencarian. Bound ini dihasilkan melalui metode pewarnaan simpul. Metode pewarnaan simpul ini akan memberikan warna kepada simpul-simpul sehingga tidak ada simpul yang bertetangga yang memiliki warna yang sama. Jumlah warna yang dihasilkan akan menjadi upper bound untuk ukuran dari clique maksimum dari suatu graf.

II. ALGORITMA DASAR

2.1 ALGORITMA BRANCH DAN BOUND

Algoritma dasar branch and bound yang akan digunakan ialah algoritma MCR. Algoritma ini dimulai dari clique yang kecil dan berlanjut untuk menemukan clique-clique yang lebih besar sampai salah satu dari clique ini diverifikasi bahwa clique ini lah yang memiliki ukuran terbesar. Algoritma ini dimulai dengan membuat variable global Q dan Qmax, dimana Q akan berisi simpul-simpul dari clique sebelumnya dan Qmax berisi simpul-simpul yang memiliki ukuran terbesar yang ditemukan sementara.

Pseudocode dari algoritma ini dapat Anda lihat dibawah ini

```

Procedure MaxClique(R,C)
  while R tidak sama dengan  $\emptyset$  do
    pilih a vertex p dengan warna maksimum C(p) dari set R
    R = R/{p} // simpul p dihapus di R
    if  $|Q| + C(p) > |Q_{max}|$  then
      Q = Q  $\cup$  {p}
  
```

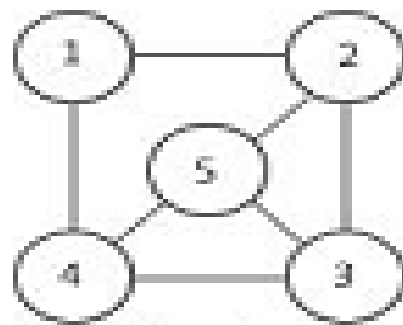
```

if  $R \cap \Gamma(p)$  tidak sama dengan  $\emptyset$  then
  MaxClique( $R \cap \Gamma(p)$ , C)
else if  $|Q| > |Q_{max}|$  then
  Qmax = Q;
  Q = Q/{p} // simpul p dihapus di Q
else return
end while
  
```

Gambar 2 Pencarian Clique Maksimum dengan Algoritma Branch and Bound

Algoritma diatas dimulai dari himpunan kosong dari himpunan Q dan secara rekursif akan menambah dan menghapus simpul-simpul dari himpunan Q, sampai tidak ditemukan clique dengan simpul-simpul yang lebih besar ditemukan. Simpul selanjutnya yang akan ditambah ke Q dipilih dari himpunan simpul-simpul kandidat R yang merupakan himpunan bagian dari V. Pertama kali himpunan R akan diinisialisasi dengan himpunan V. Di setiap langkah, algoritma akan mengambil simpul p yang merupakan anggota dari R dengan jumlah warna yang maksimal dari C(p) diantara simpul-simpul di R. Ketika simpul p tersebut diambil simpul p tersebut di R akan dihapus. C(p) adalah upper bound untuk ukuran clique maksimum di himpunan R. Jika jumlah dari $|Q| + C(p)$ lebih besar dari ukuran dari Qmax, maka simpul p akan ditambahkan ke himpunan Q. $\Gamma(p)$ merupakan himpunan semua simpul-simpul yang bertetangga dengan simpul p. Himpunan kandidat $R \cap \Gamma(p)$ dengan simpul-simpul yang berkorespondensi dengan simpul-simpul C dan akan dijadikan parameter di fungsi rekursif ke prosedur MaxClique. Jika $R_p = \emptyset$ dan $|Q| > |Q_{max}|$ (Clique sementara lebih besar dibandingkan clique terbesar yang ditemukan) maka simpul-simpul dari Q akan disalin ke Qmax. Algoritma selanjutnya akan melakukan backtrack dengan menghapus p dari Q dan kemudian memilih simpul selanjutnya dari R. Prosedur ini akan terus dijalankan sampai $R = \emptyset$.

Mungkin Anda sedikit bingung dengan algoritma ini. Alangkah baiknya apabila kita menelusuri algoritma ini dengan contoh. Misalkan terdapat graf G seperti gambar dibawah ini.



Gambar 3 Contoh Graf

Dari graf diatas maka kita dapat mendefinisikan graf $G = (\{1,2,3,4,5\}, E)$. Langkah pertama yang dilakukan algoritma ini ialah dengan membuat variable $Q = \{\}$ dan

$Q_{max} = \{ \}$. Inisialisasi awal dari R akan bernilai $R = \{1,2,3,4,5\}$. Selanjutnya kita akan mendefinisikan himpunan C yang akan berisi nilai $C(p)$ dimana p merupakan salah satu simpul dari graf G. Himpunan C diisi dengan metode pewarnaan dimana tidak ada satu simpul memiliki warna yang sama dengan warna yang lain. Dengan teknik pewarnaan maka nilai C akan bernilai (berurutan dari simpul 1 sampai simpul 5), $C = \{1,2,3,4,1\}$.

Setelah inisialisasi dilakukan, kita akan memasuki algoritma MaxClique ini, Karena R tidak sama dengan himpunan kosong maka simpul p dari R yang memiliki warna maksimum akan diambil, dalam hal ini simpul yang akan diambil adalah simpul 4 karena $C(4) = 4$. Simpul 4 akan dihapus dari himpunan R. R sekarang akan bernilai $R = \{1,2,3,5\}$. Kemudian simpul 4 akan ditambahkan ke himpunan Q sehingga $Q = \{4\}$. Selanjutnya kita perlu mendefinisikan $\Gamma(4)$. $\Gamma(4) = \{1,3,5\}$. Karena $R \cap \Gamma(4) = \{1,3,5\}$ dan tidak sama dengan himpunan kosong maka selanjutnya program akan memasuki proses rekursif prosedur MaxClique dengan parameter masukan MaxClique($R \cap \Gamma(4)$, C).

Selanjutnya, karena $R \cap \Gamma(4) = \{1,3,5\}$ tidak sama dengan himpunan kosong maka simpul 3 akan diambil karena memiliki warna terbesar jika dibandingkan dengan simpul 1 dan simpul 5. Simpul 3 akan dihapus dari $R \cap \Gamma(4)$. $R \cap \Gamma(4)$ sekarang akan bernilai $\{1,5\}$. Kemudian simpul 3 akan ditambahkan ke himpunan Q sehingga $Q = \{3,4\}$. Selanjutnya kita akan mendefinisikan $\Gamma(3) = \{2,4,5\}$. Karena $(R \cap \Gamma(4)) \cap \Gamma(3) = \{5\}$ tidak sama dengan himpunan kosong maka program akan memasuki proses rekursif kembali dengan parameter masukan MaxClique(X, C) dengan $X = (R \cap \Gamma(4)) \cap \Gamma(3)$.

Kemudian karena $X = \{5\}$ tidak sama dengan himpunan kosong maka simpul 5 yang merupakan simpul satu-satunya akan dihapus dari X. X sekarang akan bernilai $\{ \}$. Kemudian simpul 5 akan ditambahkan ke himpunan Q sehingga $Q = \{3,4,5\}$. Selanjutnya kita definisikan $\Gamma(5) = \{2,3,4\}$. Karena $X \cap \Gamma(5)$ sama dengan himpunan kosong maka program akan melanjutkan ke instruksi

```
else if  $|Q| > |Q_{max}|$  then
     $Q_{max} = Q;$ 
```

Karena jumlah $|Q| = 3 > |Q_{max}| = 0$, maka $Q_{max} = Q$, sehingga nilai Q_{max} sementara sekarang bernilai $Q_{max} = \{3,4,5\}$.

Karena $X = \{ \}$, maka prosedur MaxClique(X, C) berakhir dan beralih kembali ke prosedur MaxClique($R \cap \Gamma(4)$, C). Di akhir prosedur simpul 3 akan dihapus dari Q, sehingga Q sekarang bernilai $\{4,5\}$. Karena $R \cap \Gamma(4)$ belum sama dengan himpunan kosong maka simpul 1 akan diambil dari $R \cap \Gamma(4)$ walaupun memiliki nilai warna yang sama dengan simpul 5, sehingga nilai dari $R \cap \Gamma(4)$ sekarang ialah $R \cap \Gamma(4) = \{5\}$. $\Gamma(1) = \{2,4\}$. Karena $(R \cap \Gamma(4)) \cap \Gamma(1)$ sama dengan himpunan kosong maka

simpul 1 akan dihapus dari himpunan Q. Begitupula dengan simpul 5 untuk loop selanjutnya. Dan pada titik ini program akan kembali ke prosedur MaxClique (R, C). Begitu seterusnya sampai R mencapai himpunan kosong dimana semua himpunan R telah diperiksa.

Dari graf pada gambar 2 diatas akan didapat nilai maksimum dari clique adalah 2 dengan jumlah clique juga 3 yaitu $\{3,4,5\}$ dan $\{2,3,5\}$

2.2 ALGORITMA PEWARNAAN DENGAN PENDEKATAN PERKIRAAN

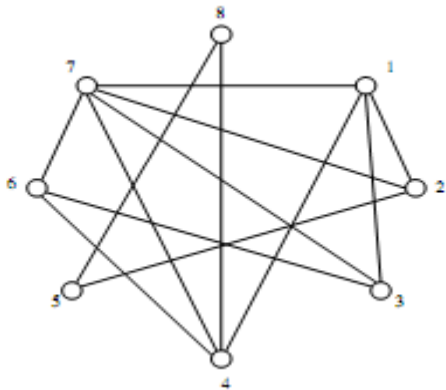
Algoritma pewarnaan dengan pendekatan perkiraan pertama kali diperkenalkan oleh Tomita dan Seki yang menyediakan pewarnaan simpul pada algoritma MaxClique. Semua simpul-simpul diwarnai di himpunan kandidat satu persatu sesuai dengan urutan yang terdapat di himpunan. Algoritma ini akan memasukkan setiap simpul v anggota dari R ke dalam warna pertama C_k yang mungkin, sehingga v tidak bertetangga dengan semua simpul-simpul yang memiliki warna yang sama. Jika simpul v memiliki setidaknya satu simpul yang memiliki warna yang sama dengannya maka warna baru akan C_{k+1} akan diberikan kepada simpul v. Keluaran dari algoritma adalah himpunan R yang baru dan himpunan warna C, dimana warna-warna pada himpunan C berkorespondensi satu-satu dengansimpul R.

III. PERBAIKAN ALGORITMA PEWARNAAN

Pada bab 2 telah dijelaskan algoritma branch dan bound dan pewarnaan untuk mencari maksimum clique dalam suatu graf. Pada bab ini saya akan mencoba melakukan improvisasi terhadap algoritma pewarnaan pada bab 2.2. Pada bab 2.2 pewarnaan akan dilakukan dengan urutan yang tidak menaik. Hal ini berarti urutan simpul-simpul di R setelah aplikasi dari algoritma pewarnaan untuk himpunan ini akan sama dengan urutan simpul-simpul di R sebelum algoritma ini dijalankan. Hal ini bukanlah kasus untuk algoritma perkiraan pada bab 2.2, dimana urutan simpul-simpul di R sesuai dengan warnanya, sehingga algoritma MaxClique di setiap langkahnya akan memilih simpul dengan warna yang maksimal dari himpunan R dimana warna yang maksimal akan berada di simpul terakhir dari himpunan. Kita akan mengobservasi simpul-simpul v dimana simpul v ini merupakan anggota himpunan R dengan warna $C(v) < |Q_{max}| - |Q| + 1$, tidak diperlukan terurut dengan warnanya seperti algoritma MaxClique tidak akan pernah menambah simpul simpul ini ke himpunan clique Q. Properti inheren dari simpul-simpul ini adalah warna dari simpul lebih rendah dari warna yang seharusnya, dimana kita namakan variable kmin. Kita menggunakan sebuah counter j dari simpul-simpul.

Permulaan dari algoritma pewarnaan ini kita akan menghitung k_{min} dimana $k_{min} = |Q_{max}| - |Q| + 1$ dan kita mengeset $j = 0$. Jika $k_{min} \leq 0$, maka kita set k_{min} menjadi 1.. Ketika berada di loop pada prosedur MaxClique, simpul v yang berada di posisi i -th di himpunan R akan diassign dengan kelas warna C_k , selanjutnya kita akan menguji jika $k < k_{min}$ untuk simpul ini maka kita akan melakukan shift simpul v ini dari posisi i -th ke posisi j -th di R dan kita increment j dengan 1. Ketika assignment simpul-simpul ini ke kelas warna telah dilakukan, maka simpul dengan warna $k < k_{min}$ akan berada di posisi paing depan himpunan R . Simpul-simpul sisanya akan disalin dari kelas warna C_k , dimana $k \geq k_{min}$, kembali ke himpunan R seperti yang terlihat di setiap warna di kelas C_k .

Contoh dari pewarnaan ini dapat dilihat pada gambar dibawah ini



Gambar 4 Contoh graf dengan clique maximum = 3

Gambar 4 memperlihatkan gambar dari graf tidak berarah. Himpunan kandidat dari simpul-simpul ini di urutan yang tidak menaik ialah $R = \{7^{(5)}, 1^{(4)}, 4^{(4)}, 2^{(3)}, 3^{(3)}, 6^{(3)}, 5^{(2)}, 8^{(2)}\}$. Dapat Anda lihat himpunan kandidat ini akan diurutkan berdasarkan jumlah sisi yang berhubungan dengan simpul tersebut dari yang terbesar sampai yang terkecil. Arti dari $7^{(5)}$ ini ialah simpul 7 dengan jumlah sisi yang terhubung dengan simpul 7 ialah 5. Himpunan inilah yang akan menjadi input dari algoritma pewarnaan perkiraan. Berikut adalah prosedur pewarnaan dengan algoritma ini.

Procedure ColorSort(R,C)

```

max no := 1;
kmin := |Qmax|-|Q| +1;
if kmin ≤ 0 then kmin := 1;
j := 0;
C1 := ∅;
C2 := ∅;
for i := 0 to |R|- 1 do
    p := R[i]; {the i-th vertex in R}

```

```

k := 1;
while Ck ∩ Γ(p) = ∅ do
    k := k+1;
if k > maxno then
    maxno := k;
    Cmaxno+1 := ∅;
end if
Ck := Ck ∪ {p};
if k < kmin then
    R[j]:= R[i];
    j := j +1;
end if
end for
C[j - 1] := 0;
for k := kmin to max no do
    for i := 1 to |Ck| do
        R[j]:= Ck[i];
        C[j]:= k;
        j := j +1;
    end for
end for

```

Gambar 5 Algoritma Pewarnaan pada Clique

Setelah melakukan assigning simpul-simpul kepada kelas-kelas warna, algoritma pewarnaan pada bab 2.2 akan menyalin simpul-simpul dari kelas-kelas ini ke himpunan kandidat sehingga $R = \{7^{(5)}, 1^{(4)}, 4^{(4)}, 2^{(3)}, 3^{(3)}, 6^{(3)}, 5^{(2)}, 8^{(2)}\}$ dengan himpunan $C = \{1, 1, 2, 2, 3, 3, 3\}$. Jika dibandingkan dengan himpunan R di permulaan dari algoritma pewarnaan di himpunan simpul-simpul R yang baru tidak terurut sesuai dengan derajatnya. Kita berharap ketika algoritma pada bab 2.2 digunakan di algoritma MaxClique, ketidakteraturan dari simpul-simpul di himpunan kandidat akan terakumulasi di level berikutnya dari rekursif. Di algoritma ColorSort pada gambar 5, misalkan $|Q_{max}| = 2$ dan $|Q| = 0$. k_{min} akan menghasilkan nilai $k_{min} = 2 - 0 + 1 = 3$. Di loop dimana assignment simpul-simpul ke kelas-kelas warna akan dilakukan. Algoritma akan melakukan perpindahan simpul-simpul dengan warna $k < 3$ ke bagian depan dari himpunan R . Setelah semua simpul-simpul diassign ke kelas-kelas warnanya, himpunan kandidat parsial dari R akan dibentuk. $R = \{7, 1, 6, 5, 8\}$. Sisa dari simpul simpul berada di kelas warna dengan $k \geq 3$. Di dalam eksperimen komputasional di banyak graf, jumlah dari simpul-simpul di himpunan kandidat dengan $k < k_{min}$ secara rata-rata lebih besar. Oleh karena itu anggota di dalam himpunan kandidat warna ini tidak diurutkan membesar.

IV. SIMPULAN

Pencarian maksimum clique diterapkan dalam sejumlah bidang keilmuan, keahlian teknik dan bahkan dalam bidang bisnis. Beberapa penerapan clique bahkan mencapai bidang biologi meliputi solusi permasalahan struktur molekul DNA. Clique juga digunakan dalam pemrosesan citra dalam pengaturan jarak jauh, pencocokan titik koordinat dalam system informasi, permasalahan partisi data dalam kepingan memory dan lain-lain.

Pencarian clique dari suatu graf dapat dilakukan dengan banyak metode baik itu brute force, backtracking dan branch dan bound. Sekarang ini banyak sekali variasi-variasi dari algoritma untuk mencari maksimum clique dalam suatu graf. Akan tetapi semua variasi tersebut berdasar pada ketiga algoritma ini. Metode yang dipakai pada makalah ini ialah metode branch dan bound yang memanfaatkan metode BFS (Breadth First Search).

Algoritma ini bukanlah algoritma yang benar-benar mangkus, karena sampai sekarang belum ada algoritma yang benar-benar mangkus untuk menyelesaikan masalah ini. Akan tetapi telah banyak algoritma yang optimal dalam menyelesaikannya walaupun belum dapat dikatakan sempurna.

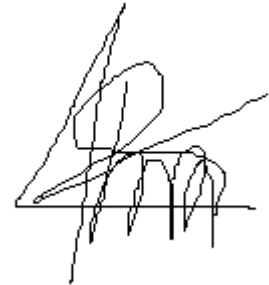
REFERENSI

- [1] E. Tomita, T. Seki, An efficient branch-and-bound algorithm for finding a maximum clique, Lecture Notes in Computer Science 2631 (2003) 278-289
- [2] Konc. Janz, Janezic Dusanka, An improved branch and bound algorithm for the maximum clique problem
- [3] D.R. Wood, An algorithm for finding a maximum clique in Graph, Operations Research Letters 21
- [4] Strickland M Dawn, Using maximum clique problem to motivate branch and bound
- [5] Munir, Rinaldi, Diktat Kuliah IF 2251 Strategi Algoritmik, ITB, 2007
- [6] http://en.wikipedia.org/wiki/Clique_problem Tanggal 5 Desember 2010 Pukul 22.00 WIB
- [7] [http://www.sicmm.org/~konc/%C4%8CLANKI/MATCH58\(3\)569-590.pdf](http://www.sicmm.org/~konc/%C4%8CLANKI/MATCH58(3)569-590.pdf) Tanggal 5 Desember 2010 Pukul 22.30 WIB
- [8] <http://www.swdsi.org/swdsi06/Proceedings06/Papers/QM01.pdf> Tanggal 5 Desember 2010 Pukul 23.00 WIB

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, 29 April 2010



Nur Adi Susliawan D C
13508081