

ALGORITMA GREEDY DALAM PERMAINAN DOTS AND BOXES

Danang Tri Massandy

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

danangstei08161@students.itb.ac.id

ABSTRAK

Permainan *Dots and Boxes* adalah permainan sederhana yang dilakukan oleh dua pemain atau lebih. Permainan ini termasuk jenis permainan yang menggunakan pensil dan kertas. Namun, sekarang permainan ini dapat disimulasikan antara pemain manusia dengan pemain komputer. Permainan ini jarang ditemukan dalam bentuk aplikasi, akan tetapi permainan ini sungguh menarik dan menantang karena banyak strategi yang bisa digunakan di dalamnya. Strategi yang digunakan seperti strategi pengorbanan, ataupun *double-cross strategy*. Pemain komputer tentunya menggunakan suatu algoritma yang mengatur bagaimana komputer bermain. Dalam tulisan ini akan dibahas bagaimana algoritma greedy diterapkan dalam menggantikan peran manusia menjadi komputer pada permainan *Dots and Boxes*. Algoritma greedy dipilih karena merupakan algoritma yang sederhana dan lempeng (*straightforward*) sehingga mudah untuk diimplementasikan.

Kata kunci: *Dots and Boxes*, Algoritma Greedy, *double-box moves*, himpunan kandidat, fungsi seleksi, himpunan solusi, fungsi kelayakan, fungsi objektif.

1. PENDAHULUAN

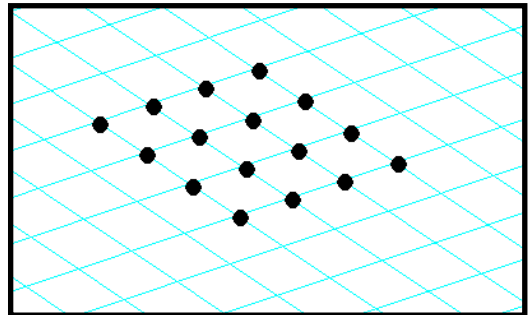
Dots and Boxes adalah permainan yang diciptakan oleh Edouard Lucas pada tahun 1889. Permainan ini dimainkan oleh dua orang atau bisa lebih dan pada awalnya berbentuk permainan pensil dan kertas. Pada zaman sekarang, permainan *Dots and Boxes* dibuat sebagai aplikasi *game* PC yang bisa dimainkan baik melawan komputer (AI) ataupun antara 2 pemain.

Perkembangan permainan *Dots and Boxes* cukup besar pada saat awal diciptakannya. Akan tetapi, sekarang ini permainan ini kurang dikenal karena perkembangan *game-game* lain yang lebih menantang. Sebenarnya permainan ini cukup sederhana, hanya dengan menggunakan pensil dan kertas saja sudah dapat dimainkan. Dan cara bermainnya hanyalah dengan menghubungkan dua titik yang tersedia sampai

terbentuk suatu kotak. Permainan ini pun sangat menantang karena strategi-strategi yang digunakan dapat berkembang sehingga bisa mengasah otak. Strategi yang digunakan dapat melibatkan analisis perhitungan matematis atau hanya dengan asal menebak saja.

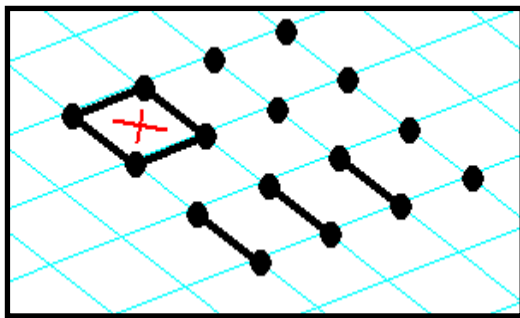
Selain karena sederhana dan menantang, permainan ini juga menarik dalam hal bagaimana kita berpikir cerdas dalam mencoret antara kedua titik (*dots*) yang ada sehingga kita bisa mendapatkan sebuah kotak (*boxes*). Sehingga jika ingin menang, tidak hanya bisa dengan asal mencoret saja. Saat menghubungkan dua titik, biasanya pemain bisa mempunyai suatu alasan tertentu untuk menuju kondisi dia mendapatkan suatu kotak. Seperti bermain catur, memainkan permainan ini harus dengan memikirkan beberapa langkah ke depan juga. Selain itu, pemain harus benar-benar memperhitungkan jumlah kotak yang akan dia dapat atau yang lawan dapatkan saat dia mengambil satu coretan antar dua titik.

Permainan ini dimulai dengan dibuat grid-grid kosong yang terdiri dari titik. Setiap pemain saat gilirannya harus menambahkan satu garis horizontal atau vertikal yang dapat menghubungkan titik-titik tersebut.



Gambar 1. Permainan *Dots and Boxes* saat awal dimulai

Pemain menghubungkan titik-titik yang ada sampai keempat titik ada yang membentuk sebuah kotak. Pemain yang melengkapi 4 titik menjadi sebuah kotak mendapatkan satu poin dan harus mengambil giliran mencoret lagi. Poin yang ditandai dengan sebuah tanda yang diletakkan dalam kotak. Salah satu yang harus dipertimbangkan dalam permainan ini yaitu kewajiban untuk mengambil giliran setelah membentuk suatu kotak atau mendapat poin. Karena jika tidak berhati-hati, pemain dapat masuk ke jebakan lawan yang dapat menyebabkan dia kalah.



Gambar 2. Pemain mendapat poin dengan memasang tanda dalam kotak

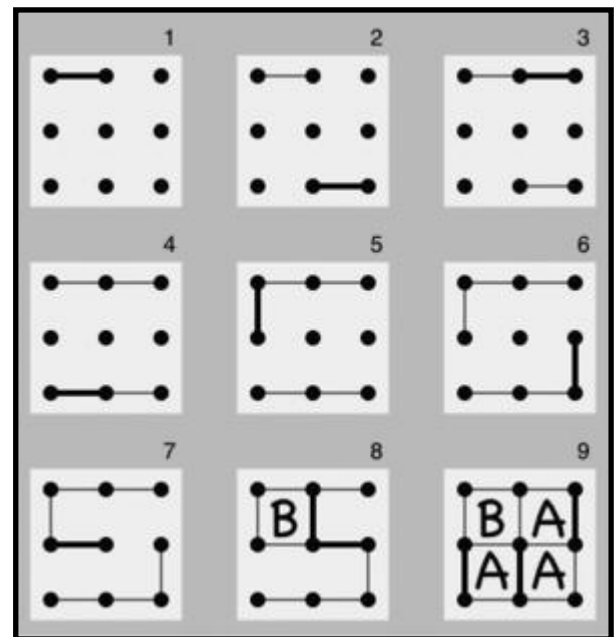
Coret mencoret dilakukan terus hingga permainan berakhir saat tidak ada garis yang bisa dicoret lagi. Pemenang dari permainan adalah pemain dengan poin yang paling banyak.

Dots and Boxes bisa terdiri dari beberapa ukuran. Pada gambar di atas digunakan 3 x 3 kotak. Selain itu, bisa terdapat 2 x 2 kotak dengan 9 titik yang biasanya untuk dimainkan pemula. Atau pun ukuran 6 x 6 kotak dengan untuk permainan yang pemainnya sudah jago/ahli. Ukuran lainnya dapat ditentukan sendiri misalnya, 3 x 5, 5 x 5, 7 x 7, 9 x 9, 11 x 11, atau 13 x 13. Semakin besar ukuran semakin banyak pula titik yang ada. Semakin besar ukuran juga semakin membutuhkan waktu yang lama untuk menyelesaikan permainan.

2. STRATEGI PERMAINAN DOTS AND BOXES

Strategi permainan dari *Dots and Boxes* yang paling umum digunakan adalah dengan memasang suatu jebakan. Yaitu dia harus merelakan atau dengan kata lain mengorbankan satu poin/atau beberapa poin untuk mendapatkan poin yang lebih banyak dari yang dikorbankan.

Contoh dari strategi di atas dapat dilihat pada gambar berikut,

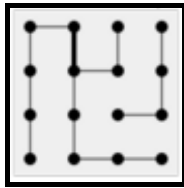


Gambar 3. Strategi pengorbanan yang dilakukan untuk mendapat poin yang lebih banyak

Gambar di atas memperlihatkan bagaimana permainan berjalan dari awal hingga akhir. Permainan ini dimainkan dalam ukuran papan 2 x 2 dengan menggunakan 9 titik. Permainan dimulai dengan langkah pertama yang dilakukan pemain A dengan mencoret garis sebelah kiri atas (posisi 1). Kemudian pemain B mencoret garis sebelah kanan bawah (posisi 2) dengan maksud agar papan dapat terbagi menjadi dua bagian dan permainan berakhir seri. Permainan berjalan dengan langkah-langkah biasa yang dilakukan hingga pada posisi 7 pemain A melakukan suatu pengorbanan untuk menjebak pemain B. Di posisi 7 sudah terdapat 3 garis yang akan membentuk kotak dan kurang 1 garis saja. Pemain B pun melengkapinya dengan senang hati karena dia akan mendapat satu poin dan kemudian dia harus mencoret garis satu kali lagi setelah mendapat poin yaitu dengan mencoret garis antara titik tengah menuju titik tengah sebelah kanan. Akan tetapi, hal ini menyebabkan kotak yang tersisa terhubung seperti rantai yang terlihat pada posisi 8. Sehingga giliran selanjutnya pemain A mengambil semua kotak yang tersisa dan A menang dengan poin 3-1. Secara umum, strategi pengorbanan terlihat seperti contoh ini. Akan tetapi, bisa saja strategi ini lebih kompleks dengan 2 atau lebih langkah ke depan yang sudah diperhitungkan.

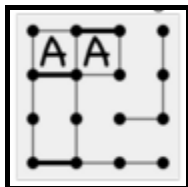
Strategi lain yang digunakan adalah strategi *double-cross strategy*. Strategi ini pada awal permainan hanya mengambil langkah random tapi diusahakan jangan sampai mengambil langkah yang mencoret sisi ketiga dari kotak. Hal ini dilakukan terus menerus sampai menemui suatu kondisi dimana semua kotak yang akan terbentuk telah tersusun menjadi sebuah grup rantai dari satu atau

lebih urutan kotak. Rantai ini terlihat seperti gambar di bawah,



Gambar 4. Sebuah rantai panjang dari kotak yang akan terbentuk

Dari sini, strategi *double-cross* digunakan untuk menjebak lawan agar pemain bisa mendapatkan kotak yang berantai ini. Caranya adalah pemain hanya boleh mengambil dua kotak dari kotak-kotak yang ada pada rantai dan mencoret lagi yang mengharuskan lawan harus membuka rantai berikutnya. Kondisi ini ditunjukkan oleh gambar berikut,



Gambar 5. Pemain yang meninggalkan dua kotak untuk lawan agar lawan membuka rantai panjang berikutnya

Dalam kondisi ini, mau tidak mau lawan harus mencoret salah satu pembuka rantai panjang terbentuknya kotak. Dengan demikian pemain yang menggunakan strategi *double-cross* ini pun menang telak.

Strategi *double-cross* ini berlaku berapapun banyak rantai panjang. Intinya, ambil semua kecuali dua kotak setiap rantai, tetapi ambil semua kotak di rantai paling akhir. Permainan *Dots and Boxes* ini menjadi permainan yang mengandalkan kendali dalam permainan, artinya pemain yang profesional akan memaksa lawannya untuk membuka rantai panjang tersebut.

Jika bermain dengan pemula, yang tidak mengetahui bagaimana konsep pengorbanan, lawan hanya harus membuat beberapa pengorbanan kotak yang menyebabkan sang pemula membuka rantai kotak panjang pertama yang cukup untuknya untuk menang. Jika pemain lawan juga mengetahui bagaimana cara pengorbanan tersebut, maka pemain harus dapat memanipulasi/memperhitungkan jumlah pengorbanan yang harus dia lakukan sejak awal permainan.

Pengorbanan sebenarnya tidaklah harus diambil oleh lawan, akan tetapi jika memang pengorbanan yang dibuat benar-benar merupakan satu-satunya langkah yang tersedia, maka lawan haruslah mengambilnya. Dengan demikian, saat membuat suatu pengorbanan haruslah dalam waktu yang benar-benar tepat yaitu memaksa lawan harus mengambilnya. Jika lawan tidak mengambil pengorbanan yang dibuat, maka pemain yang membuat pengorbanan dapat mengambil pengorbanan dia sendiri di langkah berikutnya. Sebenarnya, kasus dimana lawan

tidak mengambil pengorbanan yang dibuat bukan merupakan suatu hal yang harus dipertimbangkan dalam strategi ini.

Jika yang bermain adalah orang yang sudah profesional, maka dia akan mencoba untuk membuat lawan untuk tidak menggunakan strategi ini, yaitu pada saat awal permainan dia membuat langkah yang dapat membagi papan. Pembagian papan ini bertujuan untuk membuat permainan menjadi seri. Dalam papan yang terdapat jumlah kotak yang belum terbuat genap akan menghasilkan permainan yang seri, tetapi jumlah kotak yang belum terbuat ganjil akan membuat permainan berakhir di salah satu pemain dengan beda satu kotak saja.

Strategi-strategi di atas dapat dijadikan sebuah aturan, yaitu *The Long Chain Rule* yang berbunyi :

Anggap papan bermain adalah sebuah pesergi panjang dengan m baris dan n kolom sehingga mempunyai mn kotak. Jika m dan n adalah bilangan genap, maka pemain pertama seharusnya bermain untuk membuat jumlah rantai panjang ganjil. Jika salah satu m atau n adalah ganjil, maka pemain pertama seharusnya bermain untuk membuat jumlah rantai panjang genap.[1]

Aturan di atas jika diperlakukan untuk pemain kedua, maka pemain kedua seharusnya membuat jumlah rantai panjang genap jika baik m dan n adalah bilangan genap. Sebaliknya, jika m atau n adalah bilangan ganjil, maka pemain kedua seharusnya membuat jumlah rantai yang ganjil.

Bagaimana aturan ini bekerja? Dalam kotak permainan, ada $(m+1)n$ langkah yang dapat diambil secara horizontal dan $m(n+1)$ langkah yang dapat diambil secara vertikal. Asumsi kotak permainan berukuran $m \times n$. Sehingga total langkah yang ada adalah

$$2mn + m + n \quad (1)$$

Jika pemain bebas mencoret tanpa harus ada aturan bahwa pemain yang telah melengkapi kotak harus mencoret lagi, maka dapat diambil kesimpulan bahwa pemain yang melakukan langkah pertama juga akan melakukan langkah terakhir jika dan hanya jika $2mn + m + n$ adalah ganjil.

Akan tetapi, dengan aturan bahwa pemain yang sudah melengkapi satu kotak harus mencoret lagi, hipotesis di atas tidak dapat diambil karena harus diambil satu langkah setiap minimal satu kotak terisi kecuali untuk kotak terakhir. Jadi, setiap minimal satu kotak yang terisi, $2mn+m+n$ ini harus dikurangi dengan 1. Dalam permainan, beberapa langkah dapat secara langsung melengkapi dua kotak sekaligus, langkah seperti ini bisa disebut sebagai langkah *double-box*. Jika tidak ada langkah *double-box*, maka semenjak ada mn kotak dan karena melengkapi kotak terakhir tidak mengubah apa-apa, harus dilakukan pengurangan $mn-1$ dari jumlah total langkah untuk mendapatkan jumlah langkah yang menyebabkan perubahan dalam permainan. Dengan ini didapat,

$$\begin{aligned}
 2mn + m + n - (mn - 1) \\
 = mn + m + n + 1 \\
 = (m + 1)(n + 1)
 \end{aligned}
 \tag{2}$$

Dengan demikian, jika tidak ada langkah *double-box*, maka pemain yang melangkah pertama juga akan melangkah terakhir jika $(m+1)(n+1)$ adalah bilangan ganjil. Hal ini berlaku juga ketika misalkan ada jumlah *double-box* yang genap dalam permainan.

Jika dan hanya jika $(m+1)(n+1)$ adalah ganjil, maka untuk menguasai permainan pemain pertama akan mencoba membuat kondisi jumlah *double-box* dalam permainan adalah genap. Akan tetapi, jika ada rantai kotak yang panjangnya hanya satu atau dua, baik kedua pemain tidak boleh membuat langkah *double-box*. Jika terdapat rantai kotak yang panjangnya 3 atau lebih, salah satu pemain dapat mengambil semua kotak kecuali dua kotak untuk menyediakan lawan dengan satu langkah *double-box*. Untuk pengulangan 4 atau lebih kotak, salah satu pemain boleh mengambil semua kecuali 4 kotak untuk menyediakan lawan dengan dua langkah *double-box*. Dengan demikian, dalam permainan yang dimainkan dengan baik, langkah *double-box* sama dengan jumlah rantai panjang, ditambah dua kali jumlah pengulangan dikurangi satu karena pemain yang mengambil langkah pada rantai panjang terakhir akan mengambil semua kotak yang ada. Jadi jika $(m+1)(n+1)$ adalah ganjil, maka pemain pertama akan menginginkan jumlah rantai panjang yang ganjil dalam permainan. Dengan catatan, $(m+1)(n+1)$ adalah ganjil jika dan hanya jika baik m dan n adalah genap.

3. PENGGUNAAN ALGORITMA GREEDY DALAM PERMAINAN DOTS AND BOXES

Algoritma greedy adalah metode yang paling populer dan mudah diimplementasikan untuk memecahkan persoalan-persoalan yang membutuhkan suatu optimasi. Dalam permainan *dots and boxes* optimasi dilakukan pada saat kapan kita harus mengambil langkah mencoret, berapa kotak yang harus diambil, ataupun langkah mana yang harus dilakukan. Karena algoritma greedy sederhana dan lempang, maka algoritma greedy dapat diterapkan sebagai pemikir komputer dalam permainan *dots and boxes* menggantikan pemain manusia.

Algoritma greedy adalah algoritma yang memecahkan masalah langkah per langkah, dan pada setiap langkahnya mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan. Berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global. Algoritma ini mengasumsikan bahwa optimum local adalah optimum global, karena itu jika tiap langkah

didapatkan optimum lokal, maka akan diperoleh optimum global pada akhir penyelesaian masalah.

Dalam penerapan algoritma greedy sebagai pemain komputer pada permainan *dots and square*, algoritma ini memiliki batasan-batasan tertentu. Misalnya, algoritma ini hanya mencari pasangan titik yang tidak membentuk garis ketiga dalam kotak atau pasangan titik yang dapat membentuk sebuah kotak.

Algoritma greedy terdiri dari beberapa elemen, yaitu himpunan kandidat (C) yang berisi elemen pembentuk solusi, himpunan solusi (S) yang berisi kandidat yang terpilih sebagai solusi persoalan, fungsi seleksi yaitu fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan untuk mencapai solusi optimal, fungsi kelayakan (*feasible*) yaitu fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak (kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada), dan fungsi objektif yaitu fungsi yang memaksimumkan atau meminimumkan nilai solusi.

Dalam persoalan permainan *dots and boxes*, himpunan kandidat adalah titik-titik yang belum dihubungkan oleh garis. Banyaknya himpunan ini adalah $(m+1)(n+1)$ dengan m adalah panjang papan dan n adalah lebar papan. Himpunan solusi adalah pasangan titik yang tidak dapat membentuk garis ketiga dalam kotak ataupun pasangan titik yang dapat membentuk sebuah kotak. Jika tidak ditemukan pasangan titik tersebut maka himpunan solusi adalah pasangan titik yang dipilih secara acak yang dapat membentuk garis. Fungsi seleksi dalam algoritma ini adalah fungsi yang mencari pasangan titik yang dapat membentuk garis. Fungsi kelayakan adalah fungsi yang memeriksa pasangan titik yang diambil tidak membentuk garis ketiga dalam kotak. Sedangkan fungsi objektif adalah fungsi yang membentuk sebuah kotak.

Algoritma greedy dapat diperlihatkan dalam *pseudo-code* berikut,

```

function GetGaris(input C : himpunan titik )->
pasangan_titik
{mengembalikan pasangan titik yang dapat
membentuk garis dengan syarat tertentu}

Deklarasi
S : solusi_pasangan_titik
x : pasangan_titik
temp : himpunan titik

Algoritma
S <- ( )
temp <- C
while ( (S={}) and (C masih ada pasangan titik
yang bisa dibuat garis) ) do
x <- pasangan titik yang bisa membuat garis
C <- C - x
if ( x dapat membentuk kotak ) then
S <- x
endif
endwhile

```

```

C <- temp
while ( (S={}) and (C masih ada pasangan titik
yang bisa dibuat garis) ) do
  x <- pasangan titik yang bisa membuat garis
  C <- C - x
  if ( x tidak membentuk garis ketiga dalam
kotak ) then
    S <- x
  endif
endwhile

if ( S!={} ) then
  return S
else
  return pasangan titik yang bisa membuat
garis

```

Algoritma ini terdiri dari dua bagian yaitu mencari titik-titik yang dapat membentuk kotak terlebih dahulu kemudian jika tidak ditemukan maka dicari pasangan titik yang tidak membentuk garis ketiga dalam kotak. Secara umum, dua bagian ini memiliki bentuk perulangan yang sama, hanya saja berbeda pada fungsi kelayakan yang digunakan. Dalam algoritma secara keseluruhan, diprioritaskan pencarian pasangan titik yang dapat membentuk kotak terlebih dahulu dengan tujuan agar komputer mendapatkan poin sebanyaknya.

Algoritma ini berjalan menelusuri titik-titik yang ada dan memeriksa apakah dua titik yang ditemukan dapat membuat garis atau tidak (dengan fungsi seleksi). Kemudian setelah ditemukan pasangan titik yang dapat membuat garis, himpunan kandidat dikurangi dengan pasangan titik ini agar pasangan titik ini tidak dicek lagi saat pengulangan berikutnya. Fungsi kelayakan digunakan untuk memeriksa apakah pasangan titik ini dapat membentuk kotak atau tidak membentuk garis ketiga dalam kotak.

Tujuan dari algoritma ini adalah untuk mendapatkan poin sebanyak-banyaknya dengan cara memilih pasangan titik yang dapat membuat kotak. Jika tidak ditemukan, maka algoritma ini memilih pasangan titik yang tidak membuat garis ketiga pada kotak agar lawan tidak mengambil poin dari kotak yang hampir jadi tersebut.

4. KESIMPULAN

Algoritma yang diterapkan dalam permainan ini memiliki kelebihan dan kekurangan. Kelebihan dari algoritma ini adalah pencarian yang *straightforward*, selagi bisa mengumpulkan poin maka dipilihlah pasangan titik yang dapat membuat sebuah kotak.

Akan tetapi, algoritma ini jika diterapkan dalam permainan *Dots and Boxes* tidaklah sepintar algoritma lain yang lebih kompleks, misalkan algoritma MiniMax. Algoritma yang diimplementasikan dalam tulisan ini tidak dapat membuat strategi yang dijelaskan sebelumnya, karena prinsip dasar dari algoritma greedy adalah *take all what you can get now* sehingga jika diterapkan pada

permainan ini, algoritma greedy tentunya bertujuan untuk mengambil poin sebanyak-banyaknya agar pemain komputer dapat memenangkan permainan. Kenyataannya, dalam permainan sebenarnya melawan pemain manusia, selalu dapat digunakan strategi-strategi yang optimal dalam memenangkan permainan, misalkan dengan strategi pengorbanan ataupun *double-cross strategy*.

Oleh karena itu, algoritma greedy disini dapat digunakan sebagai komputer yang berlevel mudah (*easy*) untuk melatih para pemula yang baru mengenal permainan *Dots And Boxes*. Akan tetapi, tidak menutup kemungkinan algoritma greedy ini diimplementasikan dengan lebih kompleks sehingga dapat menjadi algoritma yang lebih cerdas.

REFERENSI

- [1] <http://cgi.cae.wisc.edu/%7Edwilson/play.php?moves=&opt=NB>
Waktu akses : 7 Desember 2010 15:00
- [2] <http://www.apples4theteacher.com/java/dots-and-squares/>
Waktu akses : 7 Desember 2010 15:15
- [3] http://en.wikipedia.org/wiki/Dots_and_Boxes
Waktu akses : 7 Desember 2010 15:40
- [4] <http://homepages.cae.wisc.edu/~dwilson/boxes/intro.shtml>
Waktu akses : 7 Desember 2010 16:00
- [5] Rinaldi Munir, "Diktat Kuliah IF3051 Strategi Algoritma", Informatika, 2009

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, 9 Desember 2010



Danang Tri Massandy
13508051