

Penggunaan Algoritma Brute Force dan Greedy dalam Permainan Atomas

Feryandi Nurdiantoro - 13513042
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13513042@std.stei.itb.ac.id - feryandi.n@gmail.com

Abstrak—Makalah ini menjelaskan cara penyelesaian *game mobile* berjenis *puzzle* bernama Atomas dengan menggunakan pendekatan algoritma greedy dan brute force. Tujuan dari makalah ini adalah agar pengaplikasian algoritma didalam ilmu informatika tidak hanya berada di dalam perkuliahan, pekerjaan maupun didalam kode-kode program tetapi juga dapat diaplikasikan dalam kehidupan keseharian bahkan pada saat sengang dan bermain *game*.

Kata kunci— Permainan, Game, Brute Force, Greedy, Algoritma, Solusi, Atomas, Puzzle

I. PENDAHULUAN

Permainan *mobile* merupakan hal yang tidak bisa dilepaskan lagi dari kehidupan kita sehari-hari. Sangat sering kita dalam waktu luang memainkan berbagai permainan yang tersedia di Internet dan memainkan permainan tersebut untuk menghabiskan waktu luang. Tetapi selain dapat bermain, sebuah permainan juga dapat memberikan pelajaran bagi pemainnya dan mengaplikasikan pengetahuan yang dimiliki oleh pemainnya untuk memenangkan permainan serta mengetahui alur dan cara kerja sebuah permainan.

Salah satu permainan yang ingin diangkat disini adalah permainan berjenis *puzzle* bernama Atomas. Permainan ini memiliki tujuan untuk membuat atom-atom yang berada dilintasan menjadi sesimetris mungkin dan jika diberikan *energy ion* atom yang sejenis dan bersebelahan dapat bergabung menjadi atom yang massanya lebih besar, dan jika disebelahnya memiliki kesimetrisan, maka akan terjadi reaksi berantai hingga kesimetrisan dengan titik acuan tersebut tidak ada lagi di lintasan itu.

Pada permainan ini juga pemain dapat belajar nama-nama unsur dasar yang ada di tabel periodik dari atom dengan massa 1 yakni Hidrogen hingga atom dengan massa terbesar yang pernah dibuat dilaboratorium. Dengan bermain permainan ini, pemain secara tidak langsung menghafalkan unsur-unsur yang ada didalam tabel periodik.

Pada makalah ini akan dibahas cara penyelesaian *puzzle* permainan ini menggunakan algoritma greedy dan brute force sehingga mendapatkan skor yang maksimum dan atau mencapai atom dengan massa tertinggi.

II. DASAR TEORI

A. Brute Force

Algoritma brute force adalah algoritma yang polos, terang-terang, sudah jelas atau *straightforward*. Algoritma ini mencoba segala kemungkinan solusi yang mungkin dari suatu permasalahan, membandingkan setiap solusi yang ada dan mendapatkan jawabannya dari hasil perbandingan itu.

Contoh sederhana pemakaian brute force didalam perhitungan pangkat suatu bilangan misalnya,

$$a^n = a * a * a * \dots * a$$

Algoritma brute force akan membutuhkan n-1 kali proses perkalian dalam menyelesaikan masalah ini.

Contoh lain adalah penggunaannya didalam mencari suatu nilai terbesar didalam suatu *array*. Untuk mencari nilai terbesar, algoritma brute force akan melakukan pengecekan satu persatu disetiap elemen *array* yang ada dan untuk setiap kali pengecekan dilakukan perbandingan nilai terbesar, hingga seluruh elemen *array* dicek. Brute force merupakan algoritma yang apa adanya dan tidak membutuhkan hal-hal khusus untuk mengerti dan mempelajarinya.

Keuntungan dari penggunaan brute force ini adalah semua masalah dapat dipecahkan dengan algoritma ini, algoritma yang digunakan dan ditulis sangat mudah dimengerti dan simpel, serta berguna untuk memecahkan masalah yang kecil.

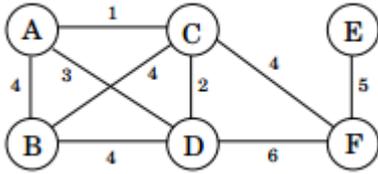
Kekurangan algoritma ini di bandingkan dengan algoritma lain adalah kebutuhan waktu pemrosesan yang akan semakin bertambah seiring dengan bertambahnya data yang diproses atau dapat dikatakan bahwa kompleksitas algoritma brute force sangat tinggi.

B. Greedy

Algoritma greedy merupakan pendekatan terhadap suatu solusi dengan cara yang maruk (*greedy*). Algoritma ini bekerja dengan cara membangun satu persatu solusi yang muncul oleh suatu masalah disuatu waktu dengan cara mengambil solusi yang paling baik saat itu tanpa memikirkan konsekuensi yang akan didapatkan dimasa

depan.

Contoh penggunaan algoritma greedy dapat dilihat pada pencarian jarak terpendek pada sebuah pohon merentang, pada pohon merentang terdapat masing-masing jarak untuk masing-masing titik yang ada,



Gambar 1. Pohon Merentang[3]

Dari gambar pohon merentang diatas, dalam mencari jarak terdekat, algoritma greedy (misalkan pencarian dimulai dari A ke E) akan mendapatkan bahwa jarak terdekat dari A ke titik selanjutnya adalah 1 yakni ke C, dilanjutkan dengan D dengan jarak 2 lalu menuju ke F dengan jarak 6 dan E dengan jarak 5 dan total jaraknya adalah 14.

Solusi greedy = (A, C, D, F, E)

Padahal, jarak terdekat A ke E adalah 10 dengan jalur (A, C, F, E). Namun dalam hal ini, solusi dari algoritma greedy telah cukup baik dan cukup dapat diterima.

Algoritma greedy memiliki kemungkinan untuk mendapatkan solusi yang optimal, namun jika terdapat beberapa solusi dan salahsatunya merupakan solusi lokal dan solusi global, maka algoritma greedy dapat terjebak dengan solusi lokal sebagai jawabannya, seperti pada contoh pohon merentang diatas.

III. PERMAINAN ATOMAS

A. Aturan Permainan

Permainan dimulai dengan beberapa atom yang telah ada di lintasan dan atom yang dapat dikendalikan berada di tengah lintasan. Terdapat tiga jenis atom yang dapat dikendalikan oleh pemain, yakni:

1. Atom biasa
Atom ini merupakan atom biasa yang dapat digabungkan dengan atom sejenis menggunakan *energy ion* yang berada diantara kedua atom sejenis bersebelahan.
2. Atom *energy ion* (berwarna merah dengan +)
Atom ini dapat menggabungkan atom sejenis yang berada disamping kanan-kirinya dan juga dapat membuat reaksi berantai
3. Atom *de-energy ion* (berwarna biru dengan -)
Atom ini memiliki kemampuan mengambil kembali atom yang telah diletakkan pada lintasan dan mengubahnya menjadi *energy-ion* atau tetap menjadi atom semula.

Setiap atom hanya dapat dimasukkan kedalam sela-sela diantara 2 atom dilintasan dan bukan menggantikannya.

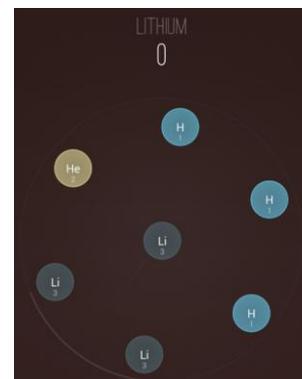
Atom akan bergabung dengan atom lainnya jika diantara atom sejenis tersebut terdapat *energy ion*, selain itu atom sejenis juga dapat bergabung melalui reaksi berantai yang dihasilkan dari 2 atom sejenis yang bersebelahan digabungkan oleh *energy ion* lalu disisi kedua atom yang telah bergabung terdapat atom yang sejenis satu dengan yang lain. Hal ini disebut dengan kesimetrisan didalam makalah ini.



Gambar 2. Tampilan awal permainan

B. Tujuan Permainan

Permainan ini memiliki tujuan utama yaitu mendapatkan skor setinggi-tingginya dan mendapatkan massa atom sebesar-besarnya. Skor yang tinggi didapatkan ketika menggabungkan 1 pasang atau lebih atom menjadi atom dengan massa yang lebih besar dan atom dengan massa yang tinggi didapatkan dengan menggabungkan atom tersebut dengan atom yang lainnya secara terus-menerus sesuai dengan aturan permainan yang telah dijelaskan.



Gambar 3. Tampilan permainan saat baru dimulai

Permainan ini termasuk kedalam permainan yang *endless* karena tingkat kesulitan yang semakin tinggi setiap kenaikan massa atom menyebabkan sulitnya mencapai nilai massa atom terbesar.

IV. PENCARIAN SOLUSI

Didalam makalah ini pencarian solusi terhadap permainan Atomas akan dilakukan dengan cara algoritma Greedy dan Brute Force. Menggabungkan keduanya diharapkan solusi yang dicapai bisa mendapatkan solusi global namun tidak membutuhkan waktu evaluasi yang lama dan terlalu banyak. Digunakan algoritma greedy karena didalam permainan tidak dapat mengetahui atom selanjutnya yang muncul sehingga tidak bisa memperhitungkan akibat di kemudian waktu (*future consequences*).

Terdapat beberapa kondisi pada permainan ini yang harus diperhatikan untuk dapat menggunakan algoritma yang sesuai. Kondisi-kondisi tersebut dibagi menjadi 2 bagian, yakni

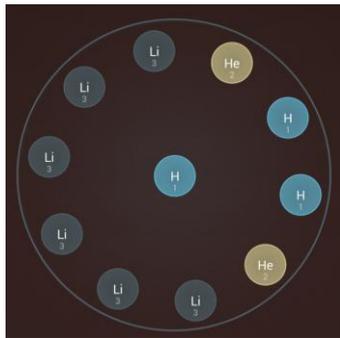
1. Terdapat 2 Atom sejenis pada lintasan yang saling bersebelahan

Kondisi ini dibagi lagi menjadi 3 kondisi yang lebih mendasar yakni, himpunan masalah yang muncul berupa atom, berupa *energy ion* atau berupa *de-energy ion*.

A. Himpunan masalah berupa atom

a. Algoritma Greedy

Algoritma ini digunakan sebagai penentuan awal titik yang akan dijadikan acuan awal untuk melakukan *brute force*. Titik yang dicari merupakan titik dengan nilai tertinggi jika titik tersebut diberikan *energy ion*.



Gambar 4. Himpunan masalah berupa atom

Pada contoh kondisi permainan diatas, terdapat 10 atom yang telah terpasang pada lintasan. Menggunakan algoritma Greedy dibuat tabel yang merepresentasikan nilai setiap tempat diantara kedua atom jika *energy ion* ditaruh pada titik tersebut. Untuk setiap 2 atom yang bergabung, akan mendapatkan 1 poin.

He-Li	Li-Li	Li-Li	Li-Li	Li-Li	Li-Li	Li-He	He-H	H-H	H-He
0	1	2	5	2	1	0	0	1	0

Tabel 1. Perhitungan kesimetrisan suatu titik

Dari tabel diatas, didapatkan nilai terbesar berada di titik

Li-Li (kolom ke 4). Kolom ke 4 ini dipilih sebagai patokan untuk algoritma Brute Force yang akan dilakukan.

b. Algoritma Brute Force

Untuk kasus pada Gambar (4) didapatkan bahwa nilai terbesar saat *energy ion* dilepaskan dititik pada kolom ke 4. Algoritma Brute Force menggunakan acuan tersebut untuk meletakkan atom yang berada ditengah. Dicoba setiap kemungkinan titik tempat peletakan atom sehingga mencapai nilai gabungan 2 atom yang besar jika *energy ion* diletakkan pada kolom 4.

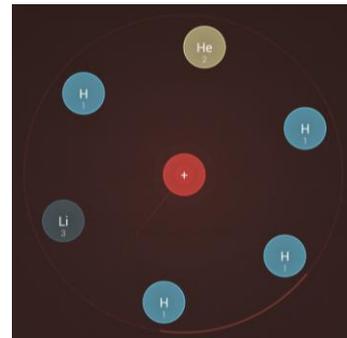
He-Li	Li-Li	Li-Li	Li-Li	Li-Li	Li-Li	Li-He	He-H	H-H	H-He
3	2	1	x	1	2	3	5	5	5

Tabel 2. Perhitungan kesimetrisan dengan titik *energy ion* yang telah ditentukan

Dari perhitungan diatas, didapatkan bahwa pada titik dikolom 8, 9, dan 10 nilai gabungan 2 atom akan mencapai maksimum. Pada pemilihan keputusan ini digunakan lagi algoritma Greedy untuk menyelesaikan permasalahan. Maka, akan diambil solusi yakni salah satu diantara ketiga kolom tersebut.

B. Himpunan masalah berupa *energy ion*

Algoritma yang digunakan pada kasus ini adalah algoritma Greedy. Penentuan titik langsung ditujukan untuk *energy ion* karena himpunan permasalahan yang dicari solusinya merupakan *energy ion* itu sendiri.



Gambar 5. Himpunan masalah berupa *energy ion*

Pada kondisi permainan diatas, terdapat 6 atom yang berada di lintasan, dan dapat dibuat tabel representasi titik diantara kedua atom dan poin gabungan 2 atom sebagai berikut.

He-H	H-Li	Li-H	H-H	H-H	H-He
0	0	0	1	1	0

Tabel 3. Tabel kesimetrisan atom

Menggunakan algoritma Greedy dan perhitungan diatas, maka solusi yang dapat diambil adalah titik pada kolom ke 4 dan ke 5 sehingga *energy ion* akan diletakkan di

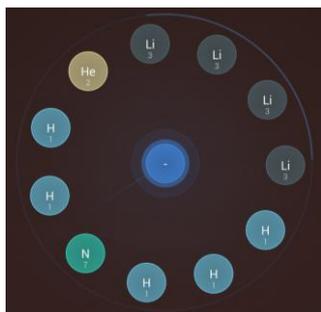
salah satu antara kedua titik tersebut. Solusi ini adalah solusi final karena tidak membutuhkan algoritma Brute Force untuk mendapatkan solusi yang diinginkan.

C. Himpunan masalah berupa de-energy ion

Atom jenis *de-energy ion* cukup sulit karena memiliki dua sifat, karena dapat menjadi atom lain yang diambil dari lintasan, lalu dapat tetap menjadi atom tersebut atau berubah menjadi *energy ion*.

a. Algoritma Greedy

Untuk menentukan atom yang akan diambil / ditarik oleh *de-energy ion* ini, dihitung jumlah setiap jenis atom dan akan diambil jenis atom yang tidak memiliki jumlah lebih dari 1 (namun lebih besar dari nol).



Gambar 6. Himpunan masalah berupa de-energy ion

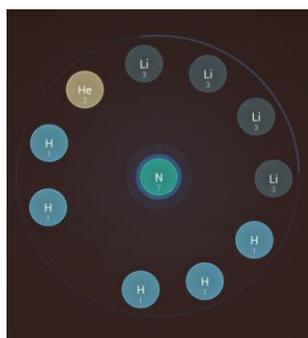
Jenis	H	He	Li	N
Jumlah	5	1	4	1

Tabel 4. Jumlah setiap jenis atom

Himpunan solusi dari permasalahan ini adalah atom {He, N} yang akan diuji selanjutnya. Karena sifatnya yang mendua, maka kedua kemungkinan juga harus dicoba.

b. Algoritma Brute Force

Dari hasil algoritma greedy pada contoh gambar (6) didapatkan 2 solusi yang mungkin, masing-masing solusi tersebut akan dicari nilai poin kesimetrisan terbesar dengan algoritma yang digunakan pada himpunan permasalahan berupa atom serta berupa *energy ion* karena sifatnya yang mendua.



Gambar 7. Jika atom N yang dipilih



Gambar 8. Jika atom He yang dipilih

Untuk kasus pada gambar (7) dilakukan perhitungan sebagai berikut,

(i) Dengan menganggap Atom yang diambil tetap

Didapatkan tabel yang merepresentasikan nilai setiap tempat diantara kedua atom jika *energy ion* ditaruh pada titik tersebut seperti dibawah ini,

Li-He	He-H	H-H	H-H	H-H	H-H	H-Li	Li-Li	Li-Li	Li-Li
0	0	1	2	2	1	0	1	2	1

Tabel 5. Tabel kesimetrisan tiap titik

Dari tabel, didapatkan himpunan solusinya adalah { H-H(4), H-H(5), Li-Li(9) }, salah satu diantaranya diambil menjadi solusi dan dilanjutkan langkah selanjutnya.

Dianggap bahwa solusi yang diambil adalah Li-Li kolom ke 9. Maka untuk mendapatkan solusi selanjutnya dibuat tabel kesimetrisan jika ditambahkan atom masalah N pada lintasan.

Li-He	He-H	H-H	H-H	H-H	H-H	H-Li	Li-Li	Li-Li	Li-Li
2	2	2	2	2	2	2	1	X	1

Tabel 6. Kesimetrisan pada titik Li-Li yang ditentukan

Didapatkan bahwa himpunan solusi merupakan kolom 1 hingga kolom ke 8. Dan dicatat bahwa kesimetrisan paling tinggi yang didapat adalah 2.

(ii) Dengan menganggap Atom yang diambil menjadi *energy ion*

Penentuan titik langsung ditujukan untuk *energy ion* karena himpunan permasalahan yang dicari solusinya merupakan *energy ion* itu sendiri.

Li-He	He-H	H-H	H-H	H-H	H-H	H-Li	Li-Li	Li-Li	Li-Li
0	0	1	2	2	1	0	1	2	1

Tabel 7. Kesimetrisan masing-masing titik

Didapatkan bahwa himpunan solusi berada di titik yang ditunjuk oleh kolom 4, 5 dan 9. Kesimetrisan paling tinggi yang didapat adalah 2.

(iii) Pemutusan langkah

Melalui dua proses sebelumnya, didapatkan bahwa kesimetrisan paling tinggi untuk kedua proses adalah 2. Sehingga keduanya merupakan himpunan solusi dan kedua solusi dapat diterima dan dijalankan.

Ketiga langkah tersebut juga akan dilaksanakan pada kasus untuk gambar (8) dan didapatkan juga himpunan solusi yang mungkin. Dari kedua himpunan solusi tersebut, diambil yang nilai kesimetrisannya paling tinggi dan titik tersebut menjadi titik solusi yang akan dijalankan dan dijadikan jawaban.

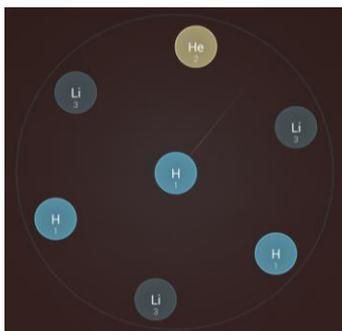
2. Tidak ada 2 Atom sejenis yang saling bersebelahan

Sama seperti kondisi sebelumnya, kondisi juga ini dibagi lagi menjadi 3 kondisi yang lebih mendasar yakni, himpunan masalah yang muncul berupa atom, berupa *energy ion* atau berupa *de-energy ion*.

A. Himpunan masalah berupa atom

a. Algoritma Greedy

Penentuan titik awal menyerupai kondisi sebelumnya, dimana dicari titik yang memiliki pasangan atom yang dapat bergabung jika titik tersebut diisi dengan *energy ion*. Namun dalam masalah ini, titik tidak lagi berada diantara kedua atom, melainkan digantikan di titik atom-atom yang ada pada lintasan.



Gambar 9. Tidak ada 2 atom sejenis berdekatan dengan himpunan masalah berupa atom

Pada contoh permasalahan yang ditunjukkan pada gambar (9), terlihat bahwa tidak ada atom sejenis yang saling bersebelahan. Untuk melakukan algoritma Greedy, sebelumnya harus dibentuk tabel representasi titik atom jika diganti dengan *energy ion* serta jumlah 2 atom yang dapat bergabung.

Dalam kasus ini, sebenarnya dicari satu titik yang memiliki kesimetrisan paling tinggi diantara titik lainnya,

He	Li	H	Li	H	Li
2	0	1	2	1	0

Tabel 8. Kesimetrisan pada titik tempat atom berada

Dari tabel diatas didapatkan bahwa himpunan solusi yang mungkin diambil oleh algoritma Greedy adalah titik He dan titik Li dimana poin kesimetrisan paling tinggi. Maka solusi yang diambil merupakan salah satu diantara kedua solusi tersebut.

Kasus khusus, jika atom masalah sama dengan atom dengan kesimetrisan paling tinggi, maka titik solusi tidak lagi dicari dengan brute force melainkan titik tersebut langsung menjadi titik solusi.

b. Algoritma Brute Force

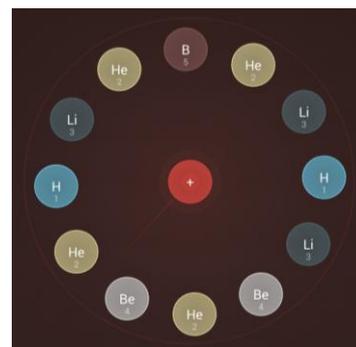
Dari solusi yang didapat menggunakan algoritma Greedy pada permasalahan diatas, digunakan algoritma Brute Force untuk mendapatkan solusi titik peletakan atom masalah. Dengan menganggap terdapat atom tambahan yang sejenis disebelah titik solusi, maka jika atom masalah diletakkan pada titik-titik antara, maka poin gabungan atom yang didapat adalah,

He-Li	Li-H	H-Li	Li-Li	Li-H	H-Li	Li-He
2	1	0	X	0	1	2

Tabel 9. Kesimetrisan terhadap titik yang ditentukan, tertanda X

Didapatkan himpunan solusi yakni titik pada kolom 1 dan kolom 7 yakni disebelah atom He pada permasalahan di Gambar (9). Dari himpunan solusi tersebut, salah satu solusi dapat diambil.

B. Himpunan masalah berupa energy ion



Gambar 10. Tidak ada 2 atom sejenis berdekatan dengan himpunan masalah berupa energy ion

Contoh permasalahan pada gambar (10) tersebut adalah himpunan masalah berupa *energy ion* dan tidak memiliki pasangan atom sejenis yang berdekatan. Untuk mendapatkan pemecahannya dilakukan dengan menggunakan algoritma Greedy yang mirip dengan himpunan masalah atom sebelumnya, hanya saja tidak menggunakan Brute Force setelahnya.

Dalam kasus ini, sebenarnya dicari satu titik yang memiliki kesimetrisan paling tinggi diantara titik lainnya namun memiliki nomor atom terendah, karena kita tidak menginginkan *energy ion* tertumpuk di lintasan dan semakin rendah nomor atom, kemungkinan muncul

semakin tinggi,

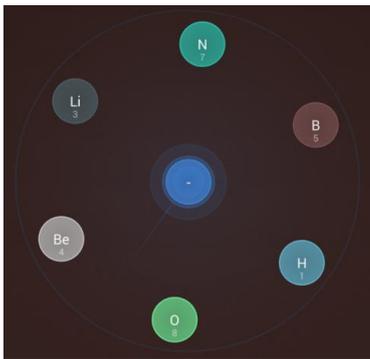
Atom	Poin Kesimetrisan	Nomor Atom	Poin / No. Atom
B	3	5	0.6
He	0	2	0
Li	0	3	0
H	0	1	0
He	0	2	0
Be	1	4	0.2
He	1	2	0.5
Be	0	4	0
Li	0	3	0
H	1	1	1
Li	0	3	0
He	0	2	0

Tabel 10. Penentuan poin tertinggi dengan memperhitungkan kemungkinan kemunculan atom

Didapatkan bahwa atom B memiliki poin kesimetrisan paling tinggi diantara atom-atom lainnya, namun kemungkinan atom B untuk muncul cukup kecil dan *energy ion* sebaiknya tidak berada dilintasan untuk waktu yang lama, sehinggadipilih titik disebelah atom H kanan sebagai himpunan solusinya.

C. Himpunan masalah berupa *de-energy ion*

Atom jenis *de-energy ion* dimana tidak ada pasangan atom berdampingan sangat sulit untuk dicari solusinya, maka solusi yang digunakan pada permasalahan ini menggunakan algoritma brute force. Dimana semua kemungkinan dicoba satu persatu untuk mendapatkan solusi yang paling sesuai.



Gambar 11. Tidak ada 2 atom sejenis yang berdekatan dengan himpunan masalah berupa *de-energy ion*

3. Pengecualian

Pencarian dan algoritma diatas selalu menggunakan *energy ion* sebagai acuan, sehingga pada kasus himpunan masalah berupa atom maupun *de-energy ion* tidak dibutuhkan lagi pengecekan menggunakan algoritma greedy untuk mencari titik penempatan *energy ion* tersimetris, namun langsung menggunakan titik yang telah ada.

V. PENGGUNAAN

Selain dapat digunakan sebagai algoritma pemecahan masalah oleh mesin otomatis atau komputer, algoritma yang telah dibahas pada makalah ini juga dapat digunakan sebagai cara berpikir dalam bermain permainan Atomas sehingga dapat juga digunakan sebagai strategi dalam bermain.

VII. UCAPAN TERIMA KASIH

Pertama-tama penulis mengucapkan terimakasih kepada Tuhan Yang Maha Esa atas berkat dan penyertaan-Nya sehingga makalah ini dapat diselesaikan dengan baik. Selanjutnya penulis ingin mengucapkan terimakasih kepada Ibu Nur Ulfa Maulidevi beserta Bapak Rinaldi Munir, untuk seluruh pengajaran dan pembelajarannya baik mengenai Strategi Algoritma maupun hal-hal lainnya yang dibutuhkan oleh penulis sebagai mahasiswa.

DAFTAR PUSTAKA

- [1] <http://www.framingham.edu/~dkeil/alg-brute.pdf> diakses pada tanggal 3 Mei 2015, pukul 15.40
- [2] http://faculty.simpson.edu/lydia.sinapova/www/cmsc250/LN250_Levitin/L05-BruteForce.htm diakses pada tanggal 3 Mei 2015, pukul 16.02
- [3] <https://www.cs.berkeley.edu/~vazirani/algorithms/chap5.pdf> diakses pada tanggal 3 Mei 2015, pukul 16.25
- [4] Munir, Rinaldi. 2004. Diktat Perkuliahan Strategi Algoritmik.

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, 3 Mei 2015

Feryandi Nurdiantoro
13513042