

Penerapan Algoritma Greedy untuk Sistem Pertahanan Udara Jarak Dekat

Evan Budianto (13511041)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganessa 10 Bandung 40132, Indonesia
13511041@std.stei.itb.ac.id

Abstrak—Sistem pertahanan udara, khususnya pertahanan udara jarak dekat, saat ini telah berkembang canggih. Hampir seluruh fungsinya diatur oleh komputer. Akibatnya pergerakan dan kerja dari senjata pertahanan ini dapat berjalan secara otomatis.

Kemampuan senjata yang baik serta perangkat keras yang canggih haruslah didukung dengan program yang baik agar sistem ini dapat bekerja semangkus mungkin. Salah satu bagian yang paling penting adalah algoritma pemilihan ancaman yang perlu dimusnahkan terlebih dahulu.

Algoritma yang akan dicoba untuk diterapkan adalah algoritma greedy dengan fungsi seleksi yaitu nilai layer terkecil dan waktu pergerakan senjata minimum dari himpunan kandidat yang tersisa.

Kata kunci—greedy, pertahanan udara, senjata otomatis

I. PENDAHULUAN

Sistem pertahanan udara jarak dekat atau yang dalam bahasa Inggris disebut Short Range Air Defense (SHORAD) merupakan kelompok senjata dan taktik anti kendaraan udara yang bertujuan untuk melindungi dari ancaman udara dengan ketinggian rendah (low-altitude), seperti helicopter dan pesawat suport yang terbang rendah (low-flying close air support) seperti jenis A-10 atau MiG-27. Kelompok senjata ini bisa juga digunakan untuk melawan target yang berukuran kecil, seperti drone/UAV (pesawat tak berawak) dan misil atau yang biasa dikelompokkan menjadi RAM targets (Rocket, Artillery, and Mortar). Karenanya senjata untuk sistem pertahanan udara ini sering juga disebut C-RAM (Counter Rocket, Artillery, and Mortar). C-RAM menjadi singkatan resmi internasional untuk senjata pertahanan udara jenis ini.

SHORAD dan sistem pertahanan sejenisnya, yaitu HIMAD (High to Medium Air Defense) yang menjangkau jarak menengah hingga jauh, dan THAAD (Terminal High Altitude Area Defense) yang menjangkau ketinggian yang lebih lagi, membagi area pertahanan udara menjadi bentuk kubah-kubah. Area tanggung jawab sistem-sistem tersebut dibagi berdasarkan ketinggian dan jarak jangkauan dari senjata.

Salah satu sistem pertahanan udara jarak dekat yang terkenal adalah Nächstbereichschutzsystem (NBS) MANTIS milik Jerman. Nama MANTIS disusun dari huruf depan Modular, Automatic and Network Capable Targeting and Interception System. NBS MANTIS

dikembangkan untuk melindungi operasi basis depan tentara Jerman di Afghanistan. Sebelumnya MANTIS dikenal sebagai NBS C-RAM. MANTIS yang memiliki senjata kaliber 35 mm otomatis penuh ini dikembangkan oleh Rheinmetall Air Defence untuk German Bundeswehr.

Area pertahanan NBS MANTIS dapat diatur secara fleksibel menyesuaikan lingkungan dan luas daerah yang perlu dilindungi. Secara pokok MANTIS terdiri dari pusat kontrol dan kontrol serangan (control and fire control center) atau yang disebut BFZ, dua radar sensor, serta maksimal 8 senjata GDF-020 (maksimal empat senjata untuk satu sensor).



Gambar 1 NBS MANTIS yang terdiri dari satu radar dan dua senjata GDF-020

sumber: www.army-technology.com

Senjata GDF-020 ini digerakkan oleh pusat kontrol (BFZ) dengan mengikuti algoritma tertentu. Apabila terdapat lebih dari satu ancaman dalam waktu yang sama, pusat kontrol harus menentukan ancaman mana yang harus diserang terlebih dahulu. Ada beberapa algoritma yang digunakan yang bisa digunakan oleh MANTIS dan SHORAD lainnya, namun tidak menutup kemungkinan untuk ditemukannya algoritma baru yang lebih efektif untuk keadaan tertentu.

Algoritma greedy merupakan salah satu algoritma yang cocok untuk diimplementasikan pada sistem seperti ini. Salah satu alasannya karena state yang harus dicek oleh sistem bisa berubah-ubah seiring berjalannya proses

algoritma greedy. Dalam kasus nyata bisa dicontohkan dengan munculnya ancaman atau serangan yang bertubi-tubi dan dengan arah serta kecepatan yang berbeda-beda. Algoritma greedy tidak akan dengan baik diimplementasikan pada sistem ini karena algoritma ini hanya mengambil kemungkinan terbaik yang dapat diambil saat itu. Algoritma yang akan dicoba adalah algoritma greedy.

II. DASAR TEORI

A. Algoritma Greedy

Algoritma greedy merupakan algoritma yang mengikuti penyelesaian masalah secara heuristic dengan menentukan pilihan optimum lokal pada setiap tahapan dan berharap akan mencapai optimum global. Pada beberapa permasalahan, strategi greedy tidak menghasilkan solusi yang optimum global, tetapi pendekatan secara greedy yang mencari optimum lokal biasanya dapat mendekati solusi optimum global.

Pada umumnya, algoritma greedy memiliki lima komponen:

1. Himpunan kandidat (C), merupakan himpunan dimana solusi diciptakan.
2. Himpunan solusi (S), merupakan himpunan yang menunjukkan solusi lengkap dari persoalan.
3. Fungsi seleksi (selection function), fungsi untuk memilih kandidat terbaik untuk dijadikan solusi.
4. Fungsi kelayakan, fungsi yang menentukan apakah suatu kandidat layak digunakan sebagai bagian dari solusi.
5. Fungsi objektif, fungsi yang menetapkan suatu nilai menjadi solusi atau bagian dari solusi

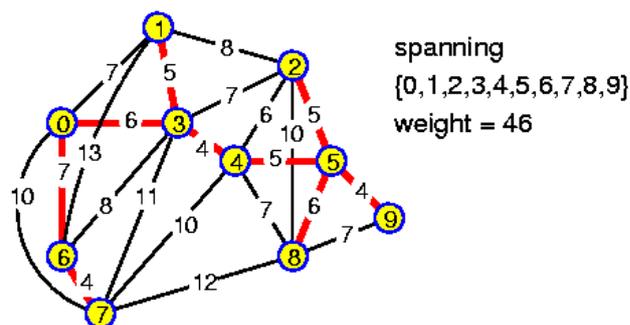
Algoritma greedy menghasilkan solusi yang baik pada beberapa persoalan matematis, tetapi terkadang tidak untuk beberapa persoalan lainnya. Sebagian besar persoalan yang menggunakan algoritma greedy akan memiliki dua property. Properti yang pertama yaitu properti pilihan greedy (greedy choice). Kita dapat membuat pilihan apapun yang tampaknya terbaik saat ini baru kemudian memecahkan submasalah yang timbul kemudian. Pilihan yang dibuat oleh algoritma greedy bergantung pada pilihan-pilihan sebelumnya tetapi tidak bergantung pada pilihan-pilihan yang akan datang maupun pada solusi dari subproblem. Secara iterative, greedy menentukan suatu pilihan setelah pilihan sebelumnya dan mengubah setiap persoalan menjadi persoalan yang lebih kecil. Dengan kata lain, algoritma greedy tidak pernah mempertimbangkan kembali pilihan yang telah diambil sebelumnya. Hal ini yang menjadi perbedaan utama algoritma greedy dengan algoritma pemrograman dinamis (dynamic programming) yang melakukan pengecekan secara exhaustive dan menjamin untuk ditemukannya solusi. Setelah melalui setiap tahap, algoritma pemrograman dinamis menentukan keputusan berdasarkan semua keputusan yang telah dilakukan pada tahap sebelumnya dan melakukan peninjauan ulang terhadap jalur menuju solusi dari algoritma pada tahap sebelumnya.

Propert yang kedua yaitu substruktur optimal. Sebuah masalah menunjukkan substruktur optimal jika solusi untuk masalah tersebut memuat solusi optimal untuk setiap sub masalah.

Algoritma greedy sering kali gagal dalam mencari solusi optimum global karena algoritma ini tidak melakukan pengecekan exhaustive secara menyeluruh untuk setiap data. Algoritma ini menentukan pilihan terlalu dini sehingga mengakibatkan tertutupnya jalur untuk mencapai solusi global yang optimum. Sebagai contoh, semua algoritma pewarnaan greedy yang diketahui untuk menyelesaikan problem grafik pewarnaan serta seluruh problem NP-complete lainnya tidak selalu mendapatkan solusi optimum. Meskipun begitu, algoritma ini sangat berguna karena cepat untuk berpikir dan sering memberikan pendekatan yang baik menuju optimum global.

Apabila suatu algoritma greedy telah dibuktikan dapat memberikan solusi optimum global untuk suatu persoalan, algoritma ini akan menjadi metode yang menjadi pilihan utama karena algoritma ini lebih cepat dari metode lainnya seperti dynamic programming. Contoh dari penggunaan algoritma greedy antara lain algoritma Kruskal dan algoritma Prim yang digunakan untuk mencari pohon rentang minimum, algoritma Dijkstra yang digunakan untuk mencari rute terpendek, dan algoritma untuk mencari pohon Huffman maksimum.

Algoritma greedy juga digunakan pada pengarahan jaringan (network routing). Dengan menggunakan greedy routing, suatu pesan dikirimkan/dilanjutkan kepada node yang terdekat dengan node tujuan. Jarak yang dimaksud pada hal ini adalah jarak secara geografis. Penerapan algoritma greedy untuk hal-hal lain juga sangat banyak.



Gambar 2 Penerapan algoritma greedy untuk kasus pohon merentang minimum

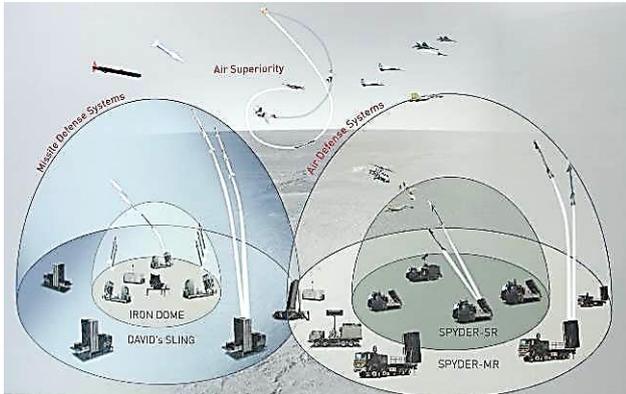
sumber: www8.cs.umu.se

B. Pertahanan Udara Jarak Dekat atau Short Range Air Defense (SHORAD)

Sistem pertahanan udara jarak dekat atau yang dalam bahasa inggris disebut Short Range Air Defense (SHORAD) merupakan kelompok senjata dan taktik anti kendaraan udara yang bertujuan untuk melindungi dari ancaman udara dengan ketinggian rendah (low-altitude), seperti helicopter dan pesawat suport yang terbang rendah (low-flying close air support). Kelompok senjata ini bisa juga digunakan untuk melawan target yang berukuran kecil, seperti drone/UAV (pesawat tak berawak) dan misil

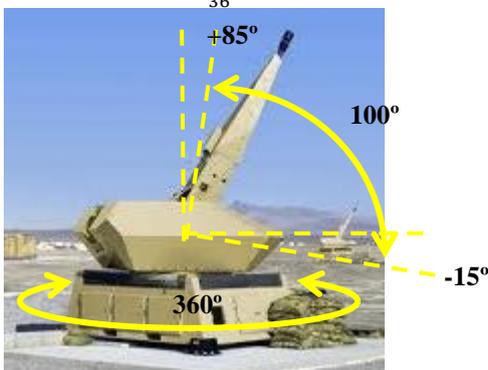
atau yang biasa dikelompokkan menjadi RAM targets (Rocket, Artillery, and Mortar).

Pada umumnya SHORAD memiliki tiga komponen utama. Komponen tersebut yaitu, pusat kendali, radar, dan senjata. Senjata dapat berupa misil maupun senapan. Senapan lebih sering digunakan karena lebih cepat dan dan memiliki fire rate yang lebih tinggi.



Gambar 3 Ilustrasi area perlindungan dari sistem pertahanan udara jarak dekat
sumber: www.rpdefense.over-blog.com

Pada makalah ini penulis akan mengambil contoh SHORAD yaitu NBS MANTIS. MANTIS memiliki senjata kaliber 35 mm otomatis penuh bernama GDF-020. GDF-020 memiliki traversing range sebesar 360° dan elevation range 100°. Elevation range menjangkau sudut -15° hingga +85°. Jangkauan radar maksimum adalah 20 km dan jangkauan senjata maksimum adalah 5 km untuk target besar dan 3 km untuk RAM (Rocket, Artillery, and Mortar). GDF-020 mampu menembakkan 1000 peluru per menit. Kecepatan tembakan (muzzle velocity) mencapai 1050 meter per detik dan burst rate 36 per tembakan. Artinya dalam satu menit GDF-020 dapat melakukan tembakan kepada $\frac{1050}{36} \approx 29$ target dalam satu menit.



Gambar 3 Traversing range dan elevation range NBS MANTIS

sumber: www.forum.nationstates.net dengan perubahan

Setiap pergerakan traverse dan elevation membutuhkan waktu tertentu hingga mengarah ke target. Selain itu dibutuhkan juga waktu untuk melakukan tembakan ke arah target. Apabila terdapat lebih dari satu target pada waktu yang sama dan datang dari berbagai arah yang

berbeda, maka senjata ini harus memilih target mana yang harus dihancurkan terlebih dahulu sehingga dapat menghancurkan sejumlah target semangkas mungkin.

III. ANALISIS PENERAPAN ALGORITMA GREEDY UNTUK SISTEM PERTAHANAN UDARA JARAK DEKAT

Deklarasi variabel:

sudut transversal = θ_T

sudut elevasi = θ_E

kecepatan sudut transversal = ω_T

kecepatan sudut elevasi = ω_E

waktu pergerakan transversal = t_T

waktu pergerakan elevasi = t_E

waktu yang dibutuhkan untuk membidik target dari kedudukan awal = t_{aim}

waktu yang dibutuhkan untuk menembak satu target hingga selesai = t_{shot}

waktu total yang dibutuhkan dari kedudukan awal hingga selesai menembak target = t_{tot}

jarak target akan dikelompokkan menjadi sebanyak n lapisan dan memiliki variabel = $layer[i]$

Semakin dekat dengan senjata, nilai layer akan semakin kecil.

Apabila sudut transversal dan kecepatan sudut transversal atau sudut elevasi dan kecepatan sudut elevasi dapat diketahui, maka kita bisa menghitung waktu pergerakan transversal dan waktu pergerakan elevasi.

$$t_T = \frac{\theta_T}{\omega_T}$$

$$t_E = \frac{\theta_E}{\omega_E}$$

waktu yang dibutuhkan oleh senjata untuk mengarah ke target dari kedudukan awal adalah

$$t_{aim} = \max(t_T, t_E)$$

waktu yang dibutuhkan untuk bergerak dari kedudukan awal hingga selesai menembak adalah

$$t_{tot} = t_{aim} + t_{shot}$$

t_{aim} mengambil nilai maksimum dari t_T dan t_E karena pergerakan secara traversal dan elevasi dapat dilakukan secara bersama, jadi waktu yang menjadi tolak ukur pergerakan senjata adalah maksimum dari keduanya. Selanjutnya t_{aim} ini akan menjadi fungsi seleksi dari algoritma greedy yang akan dibuat.

Komponen-komponen algoritma greedy yang akan digunakan yaitu:

1. Himpunan kandidat (C) : himpunan target yang akan ditembak.
2. Himpunan solusi (S) : target berhasil ditembak.
3. Fungsi seleksi : pilih target dengan nilai layer terkecil dan t_{aim} minimum dari himpunan kandidat yang tersisa.
4. Fungsi kelayakan : tidak ada target yang telah mengenai sistem pertahanan.
5. Fungsi objektif : menembak target sebanyak mungkin.

Jika ada n buah target, maka waktu yang dibutuhkan untuk menembak seluruh target adalah:

$$\sum_i^n t_{tot} [n]$$

Algoritma greedy dalam pseudocode dijelaskan sebagai berikut:

```

target T
layer minLayer L
time minTaim

minLayer = min(L) { layer minimum
dari target-target yang ada }
minTaim = min(T.aimtime) { t aim
minimum dari kedudukan senjata
sekarang }

function searchTarget(layer i, time
t) { mencari target dengan lokasi
pada layer i dan aimtime t }

function greedyTarget(target X,
layer Y, time Z)
    current_target CT =
searchTarget(Y, Z)
    layer L = layer L -
current_target CT { tembak target }
    minlayer = min(layer L) {
minimum layer yang baru }

while (T.jumlahtarget != 0) do
    greedyTarget(T)
    
```

Apabila dibandingkan dengan exhaustive search,

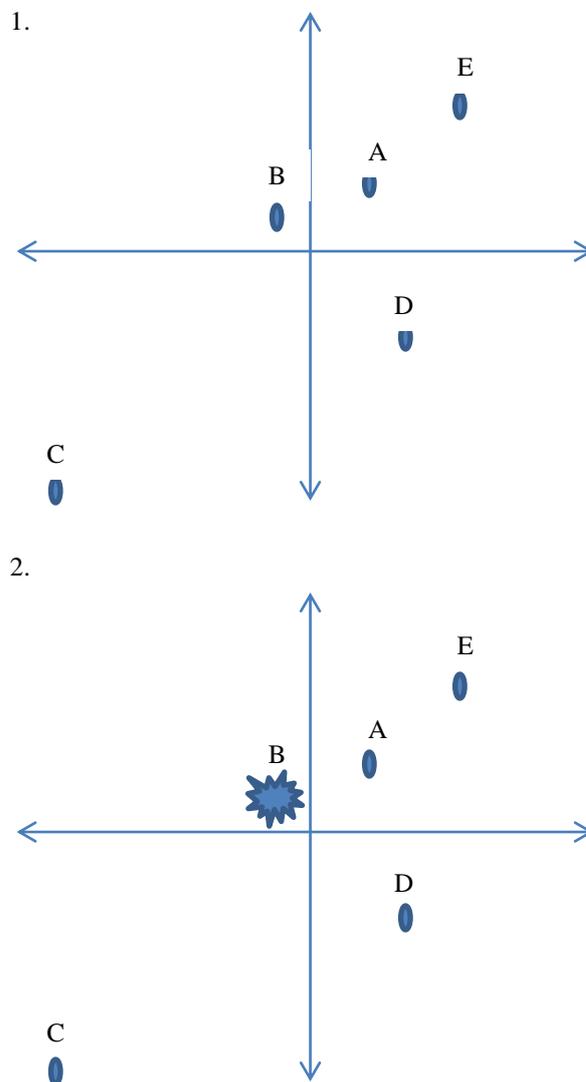
algoritma ini jauh lebih mangkus karena tidak semua kemungkinan urutan penembakan didaftar terlebih dahulu. Kompleksitas algoritma untuk exhaustive search adalah $O(n \cdot 2^n)$, sedangkan kompleksitas algoritma untuk greedy adalah $O(n)$.

Akan diberikan kasus dengan lima target yang harus ditembak. Target diberi nama A, B, C, D, dan E. Untuk memudahkan perhitungan, koordinat target didefinisikan pada diagram kartesius.

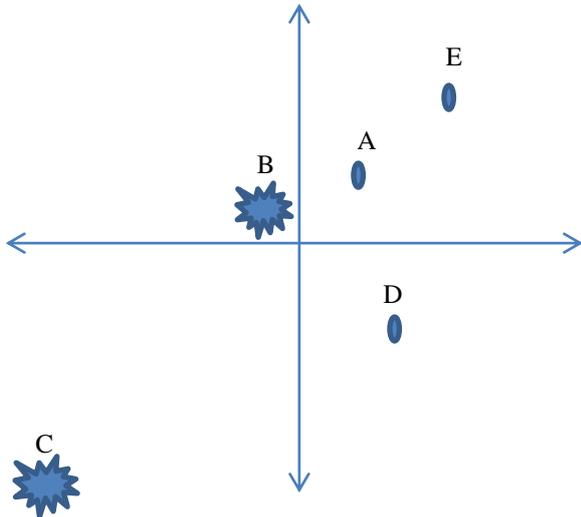
Target	Koordinat	Kecepatan (layer/session)
A	(2,2)	0,5
B	(-1,1)	1
C	(-10,10)	2
D	(3,-3)	0,5
E	(4,4)	1

Tabel 1 Target senjata beserta koordinat dan kecepatannya

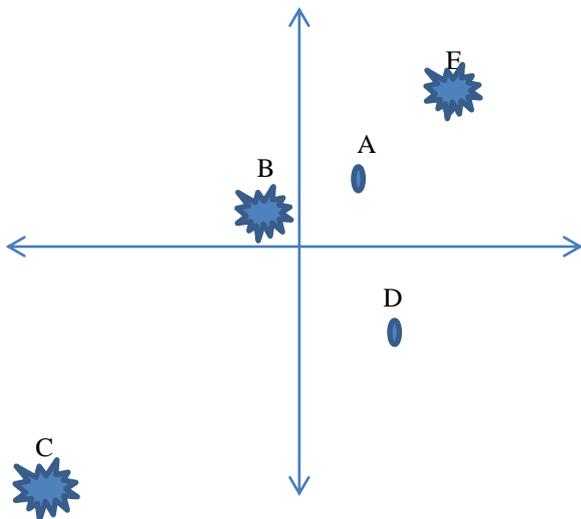
Posisi awal senjata mengarah ke titik (0,0) dan kedudukan layer seluruh target ada pada layer 10 (jumlah layer 10). Dengan menggunakan algoritma greedy yang telah dibuat, tahapan-tahapan yang dilakukan adalah sebagai berikut:



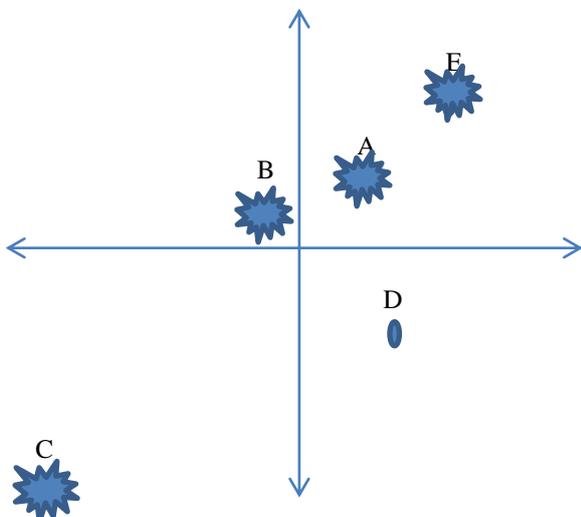
3.



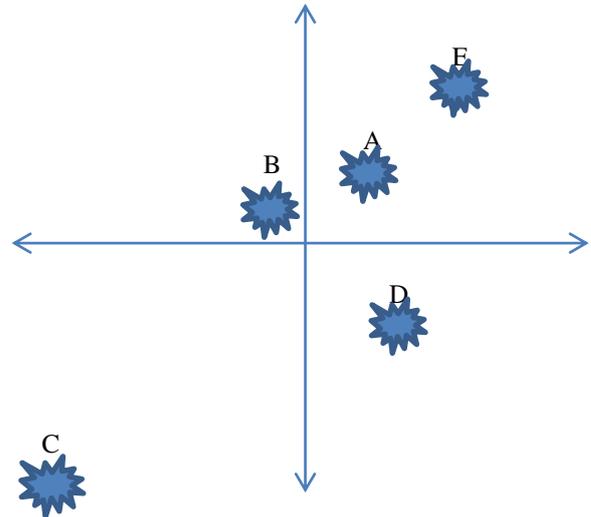
4.



5.



6.



Jadi urutan target yang ditembak apabila menggunakan algoritma ini adalah B-C-E-A-D.

Apabila diselesaikan dengan exhaustive search akan memberikan hasil akhir B-A-D-E-C. Namun akan ada perbedaan waktu yang sangat signifikan. Pada pengerjaan menggunakan exhaustive search, program harus melakukan penjabaran secara menyeluruh dari setiap solusi. Padahal ketika melewati satu tahapan, kondisi atau kedudukan dari setiap target akan berubah. Hal ini disebabkan target yang tidak diam melainkan bergerak dengan kecepatan tertentu ke arah senjata. Akibatnya, bisa saja senjata sudah terkena serangan sebelum semua target berhasil ditembak. Exhaustive search bisa saja memperoleh hasil yang sama dengan algoritma greedy pada kasus ini apabila exhaustive search dilakukan berulang kali setiap melewati satu tahap. Namun hal ini justru akan memakan waktu yang jauh lebih lama.

IV. ALGORITMA LAIN YANG DIGUNAKAN UNTUK SISTEM PERTAHANAN UDARA JARAK DEKAT

Ada beberapa algoritma yang digunakan atau dikembangkan untuk sistem pertahanan udara jarak dekat. Algoritma-algoritma tersebut antara lain Genetic Algorithm unseeded (GA), Genetic Algorithm seeded (GA-seed), Ant Colony Optimization (ACO), Random Search, Exhaustive Search, dan Particle Swarm Optimization (PSO). Setiap algoritma ini memiliki kelebihan masing-masing. Untuk kasus yang berbeda, misalkan jumlah target yang banyak atau sedikit, akan sangat berpengaruh kepada kemangkusan dari algoritma-algoritma ini.

V. KESIMPULAN

Algoritma greedy merupakan salah satu algoritma yang baik untuk diterapkan pada sistem pertahanan udara jarak dekat. Untuk jumlah target yang kecil, algoritma greedy terbukti jauh lebih efektif dan efisien dibanding dengan

exhaustive search. Selain itu, sifat algoritma greedy yang mengambil keputusan terbaik yang bisa didapat saat itu sangat cocok diterapkan untuk kasus-kasus sistem pertahanan udara yang sangat dinamis. Kita tidak tahu berapa jumlah serang, kecepatan, serta arah datang dari setiap ancaman. Karenanya, dengan menembak target yang paling potensial untuk dijatuhkan akan sangat mendukung kerja dari sistem pertahanan ini. Keuntungan lainnya adalah sistem kontrol tidak perlu melakukan terlalu banyak perhitungan seperti yang dilakukan jika menggunakan exhaustive search.

VI. UCAPAN TERIMA KASIH

Penulis berterima kasih kepada Tuhan Yang Maha Esa, yang telah memberi kekuatan dan kesempatan sehingga makalah ini dapat penulis selesaikan. Juga kepada dosen pengajar yang telah memberikan kuliah Strategi Algoritma selama satu semester ini, sehingga penulis memperoleh pengetahuan dan pengalaman untuk mengerjakan makalah ini

REFERENCES

- [1] Munir, Rinaldi. Diktat Kuliah IF2211 Strategi Algoritma. Edisi keempat. Program Studi Teknik Informatika, Institut Teknologi Bandung
- [2] *Introduction to Algorithms* (Cormen, Leiserson, Rivest, and Stein) 2001, Chapter 16 "Greedy Algorithms".
- [3] J. Bang-Jensen, G. Gutin and A. Yeo, When the greedy algorithm fails. *Discrete Optimization 1* (2004)
- [4] [Rheinmetall Air Defence System is designed to protect camp of the German Armed Forces from attack](#) . Rheinmetall press release dated 3 April 2007.
- [5] <http://www.army-technology.com/projects/mantis/>
- [6] <http://www.youtube.com/watch?v=r1jnLu8yXDY>
- [7] Luftwaffe.de
- [8] Helmut Michelis: *New missile defense comes from Dusseldorf* . In: "Rheinische Post" of 19 May 2009.
- [9] [With high-tech against missile attacks: Rheinmetall to supply protection systems to the German Army](#) . Rheinmetall press release of 19 May 2009.
- [10] [Rheinmetall defense system is not used in Kunduz](#) Handelsblatt, 5 September 2011
- [11] G. Bendall and F. Margot, Greedy Type Resistance of Combinatorial Problems, *Discrete Optimization 3* (2006).
- [12] <http://informatika.stei.itb.ac.id>

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, 20 Desember 2013



Evan Budianto - 13511041