

Penerapan Algoritma Greedy pada AEGIS Combat System Arleigh Burke-Class Destroyer dalam Skenario Pertempuran Maritim US Navy

Eduardus Alvito Kristiadi - 13522004

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13522004@std.stei.itb.ac.id

Sistem tempur AEGIS pada kapal perusak kelas Arleigh Burke merupakan sistem pertahanan canggih yang mengintegrasikan radar dan senjata untuk menghadapi ancaman maritim. Makalah ini membahas penerapan algoritma greedy dalam AEGIS Combat System untuk optimasi deteksi dan intersepsi target dalam skenario pertempuran maritim. Algoritma greedy digunakan untuk memilih target berdasarkan prioritas ancaman tertinggi, yang ditentukan dari jarak, kecepatan, dan potensi kerusakan target. Selanjutnya, senjata yang paling sesuai dipilih untuk mengintersepsi target tersebut, memastikan respons yang cepat dan efisien. Hasilnya menunjukkan bahwa penggunaan algoritma greedy mampu meningkatkan efektivitas operasional dalam menghadapi berbagai ancaman maritim.

Kata Kunci: AEGIS, Algoritma Greedy, Sistem Tempur, US Navy, Arleigh Burke-Class Destroyer, Pertahanan Maritim.

I. PENDAHULUAN



Gambar 1. United States Navy

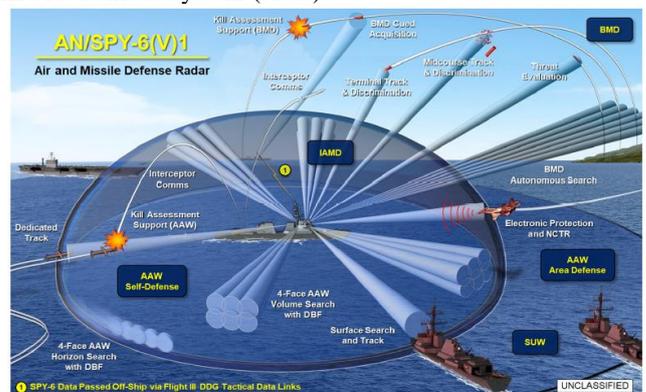
Sumber: https://en.wikipedia.org/wiki/United_States_Navy

Pada era modern ini, kekuatan militer suatu negara sering kali menjadi cerminan dari kemajuan teknologi dan kemampuan strategisnya. Saat ini, Amerika Serikat memiliki kekuatan militer terkuat di dunia. Angkatan Laut Amerika Serikat (US Navy) merupakan salah satu komponen penting dalam sistem pertahanan nasional Amerika Serikat, yang memiliki peran krusial dalam menjaga keamanan maritim global. Tidak hanya berperan dalam menjaga keamanan nasional, US Navy memainkan peran penting dalam menjaga stabilitas dan keamanan di lautan global. Dengan kehadiran militernya di berbagai wilayah strategis, US Navy mampu

melakukan proyeksi kekuatan yang signifikan, mendukung operasi tempur, melakukan pengawasan dan intelijen, serta memberikan bantuan kemanusiaan dalam situasi darurat. Misi global US Navy meliputi perlindungan jalur perdagangan internasional, pencegahan konflik, serta penanggulangan ancaman keamanan maritim seperti pembajakan dan terorisme.

US Navy saat ini memiliki struktur yang kompleks dan canggih, terdiri dari berbagai jenis kapal perang, kapal selam, pesawat tempur, dan sistem senjata canggih. Armada US Navy mencakup kapal induk, kapal penjelajah, kapal perusak, kapal fregat, kapal amfibi, dan kapal selam nuklir, yang semuanya dilengkapi dengan teknologi mutakhir. Selain itu, US Navy juga memiliki sejumlah besar personel yang terlatih dan siap untuk melaksanakan berbagai jenis operasi militer.

Perkembangan teknologi menjadi salah satu pilar utama dalam memperkuat kapabilitas US Navy. Penggunaan teknologi informasi dan komunikasi, sistem senjata presisi tinggi, serta kapal dan pesawat tak berawak merupakan beberapa inovasi yang diimplementasikan oleh US Navy untuk meningkatkan efisiensi dan efektivitas operasional. Selain itu, penelitian dan pengembangan terus dilakukan untuk memastikan bahwa US Navy tetap berada di garis depan dalam hal teknologi militer. Salah satu wujud dari perkembangan teknologi militer Amerika Serikat adalah AEGIS Combat System (ACS).



Gambar 2. AEGIS Baseline 10 and SPY-6 Integration

Sumber: https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.navy.mil%2FPortals%2F103%2FDocuments%2FExhibits%2FASAS2021%2FASAS2021-AEGIS_and_Forge.pdf&psig=AOvVaw3pPDdNNFjyU-H5WVvNvLSv&ust=1718289572137000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCNjkeq11oYDFQAAAAAdAAAAABBJ

AEGIS Combat System (ACS) merupakan sistem persenjataan anti-udara kapal perang terancang Angkatan Laut AS. AEGIS memainkan peran penting dalam pertahanan nasional dengan mengintegrasikan berbagai sensor dan sistem persenjataan untuk melindungi armada kapal tempur dari berbagai ancaman udara berupa misil, drone, pesawat tempur, helikopter atau bentuk ancaman udara lainnya, serta menyediakan pertahanan misil balistik maritim atau yang dikenal dengan *AEGIS Ballistic Missile Defense System* (Aegis BMD). Mengintegrasikan sensor modern dan perangkat keras sistem persenjataan yang sangat canggih ini adalah tugas yang sangat kompleks yang membutuhkan sistem dukungan perangkat lunak multifaset. Sistem dukungan perangkat lunak ini mungkin merupakan bagian paling kritis dalam menjaga keunggulan kompetitif atas lawan-lawan kita.

Perpindahan dari Angkatan Laut AS yang memiliki keunggulan persenjataan dan suite sensor terbaik menjadi lebih seimbang telah terdokumentasi dengan baik selama beberapa dekade terakhir. "A Design for Maintaining Maritime Superiority versi 2.0" dari Kepala Operasi Angkatan Laut menyatakan bahwa dalam lingkungan keamanan baru saat ini, "Keunggulan kompetitif kita telah mengecil dan di beberapa area, sama sekali hilang". Jumlah data yang ditransfer dalam sistem tempur hanya memperburuk pentingnya integrasi dan kualitas perangkat lunak yang menggerakkan perangkat keras.

Sistem tempur saat ini, seperti yang ada di kapal tempur Arleigh Burke-Class Destroyer yaitu AEGIS, menggunakan algoritma yang mengandalkan data kinematik (misalnya, jangkauan, arah, kecepatan, & ketinggian) dan taktis (misalnya, jenis radar & riwayat jalur terbaru) untuk menganalisis dan mengidentifikasi (ID) kontak pesawat. Sistem ini memberikan ID bersama dengan operator manusia yang mengandalkan sejumlah kecil faktor (biasanya ditentukan sebelum kapal memulai penempatannya). Informasi relevan lainnya, seperti geografi, laporan intelijen, dan negara asal, harus secara sengaja diintegrasikan oleh personel Air Defense untuk mencapai penilaian tingkat ancaman dan prioritas. Integrasi ini memerlukan tingkat keahlian taktis yang tinggi yang tidak terdokumentasi dengan baik atau dipahami. Selain itu, ada asumsi implisit bahwa tingkat ancaman dari sebuah pesawat terkait dengan identifikasinya (ID), seperti Pesawat Komersial atau Pesawat Taktis, dan oleh karena itu, tingkat ancaman yang ditimbulkan oleh sebuah pesawat dapat dijelaskan secara metodis oleh satu set faktor yang ditentukan sebelumnya.



Gambar 3. Arleigh Burke-Class Destroyer

Sumber: https://en.wikipedia.org/wiki/Arleigh_Burke-class_destroyer

Arleigh Burke-Class Destroyer adalah salah satu kapal perang US Navy yang dilengkapi dengan AEGIS Combat System, sebuah sistem manajemen tempur canggih yang mengintegrasikan radar, senjata, dan sistem kontrol api untuk deteksi, pelacakan, dan penghancuran ancaman maritim. Dalam skenario pertempuran maritim, kecepatan dan efisiensi dalam menentukan target sangat krusial untuk keberhasilan misi dan perlindungan kapal.

II. LANDASAN TEORI

A. AEGIS Combat System



Gambar 4. Aegis Combat System Block Diagram.

Sumber: Threston (2009).

Dirancang untuk dibangun pada Sistem Senjata Aegis atau *Aegis Weapon System* (AWS), di mana seluruh kapal menjadi senjata perang terpadu sepenuhnya, sistem tempur Aegis pertama kali ditempatkan di kapal penjelajah USS Ticonderoga (CG-47) pada tahun 1981 dan "terdiri dari lebih dari 850 elemen peralatan individu dan memiliki berat lebih dari 600 ton. Itu juga termasuk semua petugas dan pelaut yang diperlukan untuk mengoperasikan sistem". Komponen kritis dari Sistem Senjata Aegis ditunjukkan dalam diagram blok pada Gambar 4, yang menyoroti sistem kritis dan arsitektur terhubungnya. Gambar 4. Diagram Blok Sistem Tempur Aegis. Sumber: Threston (2009). Disusun dari kiri ke kanan, Gambar 3 mengikuti tiga fungsi tradisional perang angkatan laut yaitu deteksi, kontrol, dan serangan. Threston mendefinisikan masing-masing sebagai: Deteksi—Kehadiran potensi ancaman harus dideteksi dan lokasi, jalur, serta kecepatan mereka harus ditentukan, Kontrol—Ancaman yang terdeteksi harus

dan potensi kerusakan. Algoritma greedy kemudian memilih ancaman dengan nilai ancaman tertinggi untuk ditangani terlebih dahulu.

3. Penugasan Senjata: Senjata pertahanan yang tersedia segera dialokasikan untuk menargetkan ancaman yang diprioritaskan. Setelah satu ancaman diatasi, sistem akan beralih ke ancaman berikutnya dengan nilai ancaman tertinggi.
4. Adaptasi Real-Time: Sistem terus memperbarui prioritas ancaman secara real-time berdasarkan informasi terbaru dari radar, memastikan bahwa setiap ancaman baru yang lebih berbahaya segera ditangani.

B. Penentuan Prioritas

Dalam rangka memberikan bobot untuk algoritma Greedy, diperlukan adanya penentuan bobot berdasarkan jenis dan potensi kerusakan yang datang dari pihak musuh. Jenis-jenis target dibagi menjadi rudal balistik, rudal jelajah, pesawat tempur, kapal musuh, drone pengintai, dan helikopter serang dengan potensi kerusakan yang berbeda-beda.

Rudal balistik memiliki potensi kerusakan sangat tinggi dan kecepatan sebesar 2000 m/s. Rudal jelajah memiliki potensi kerusakan tinggi dan kecepatan sebesar 3550 m/s. Pesawat tempur memiliki potensi kerusakan tinggi sebesar 2778 m/s. Kapal musuh memiliki potensi kerusakan tinggi sebesar 13 m/s. Drone pengintai memiliki potensi kerusakan tinggi sebesar 56 m/s. Helikopter serang memiliki potensi kerusakan tinggi sebesar 84 m/s.

TABEL 1. Klasifikasi penentuan prioritas target ancaman

Jenis Target	Potensi Kerusakan	Velocity
Rudal Balistik	Sangat tinggi	7200 km/h (2000 m/s)
Rudal Jelajah	Tinggi	3,550 km/h (986 m/s)
Pesawat Tempur	Tinggi	2.778 km/h (772 m/s)
Kapal Musuh	Sedang	48 km/h (13 m/s)
Drone Pengintai	Rendah	200 km/h (56 m/s)
Helikopter Serang	Rendah	304 km/h (84 m/s)

C. Pemilihan Senjata

Selain menggunakan klasifikasi untuk penentuan prioritas target, diperlukan pula pemilihan jenis senjata untuk melakukan intercept terhadap target yang akan datang. Jenis-jenis senjata yang dimiliki oleh Arleigh Burke-Class Destroyer dibagi menjadi Meriam Mark 45 Mod 4, Phalanx CIWS, Mark 41 Vertical Launching System using RIM-67 Standard Missile-2 (SM-2), dan M242 Bushmaster dengan spesifikasi yang berbeda-beda.

Meriam Mark 45 Mod 4 memiliki rate of fire sebesar 16–20 rounds per minute *automatic*, dengan klasifikasi jarak menengah yang memiliki jarak jangkauan minimum sebesar 10000 m dan jarak jangkauan maksimum sebesar 40000 m, serta memiliki velocity sebesar 760 m/s.

Phalanx CIWS memiliki rate of fire sebesar 4500 rounds per minute dengan klasifikasi jarak menengah yang memiliki jarak jangkauan minimum sebesar 6900 m dan jarak jangkauan maksimum sebesar 9000 m, serta memiliki velocity sebesar 1100 m/s.

Arleigh Burke-Class Destroyer juga memiliki Vertical Launch System yang dilengkapi rudal jelajah (*cruise missile*) RIM-67 Standard Missile-2 (SM-2) yang memiliki rate of fire sebesar 1 round per second dengan klasifikasi jarak jauh yang memiliki jarak jangkauan minimum sebesar 41000 m dan jarak jangkauan maksimum sebesar 166680 m, serta memiliki velocity sebesar 1200 m/s.

M242 Bushmaster memiliki rate of fire sebesar 200 rounds per minute dengan klasifikasi jarak dekat yang memiliki jarak jangkauan minimum sebesar 0 m dan jarak jangkauan maksimum sebesar 6800 m, serta memiliki velocity sebesar 1100 m/s.

TABEL 2. Klasifikasi pemilihan senjata

Nama	Rate of Fire	Jarak Jangkauan Minimum	Jarak Jangkauan Maksimum	Velocity
Meriam Mark 45 Mod 4	16–20 rounds per minute <i>automatic</i>	10 km (10000 m)	40 km (40000 m)	760 m/s
Phalanx CIWS	4,500 rounds per minute (75 rounds per second)	6,9 km (6900 m)	9 km (9000 m)	1100 m/s
Mark 41 Vertical Launching System using RIM-67 Standard Missile-2 (SM-2)	1 round per second	41 km (41000 m)	166.68 km (166680 m)	1200 m/s
M242 Bushmaster	200 rounds per minute	0 km	6,8 km (6800 m)	1100 m/s

D. Source Code

1. main.py:

```

1 import time
2 from target_list import TargetList
3 from weapon import Weapon
4
5 def select_weapon(target, weapons):
6     suitable_weapons = [weapon for weapon in weapons if weapon.is_suitable(target)]
7     if suitable_weapons:
8         selected_weapon = min(suitable_weapons, key=lambda w: w.calculate_time_to_intercept(target))
9         selected_weapon.assign_target(target)
10    return selected_weapon
11    return None
12
13 weapons = [
14     Weapon("Meriam Mark 45 Mod 4", "30-20 rounds per minute", 10000, 40000, 750),
15     Weapon("Phalanx CIWS", "4500 rounds per minute", 6000, 5000, 1100),
16     Weapon("Rim-67 Standard Missile-2", "2 rounds per second", 41000, 100000, 1200),
17     Weapon("Mk41 Buckmaster", "200 rounds per minute", 0, 6000, 1100)
18 ]
19
20 target_list = TargetList()
21
22 def add_target_from_input():
23     target_types = ("Muda Balistik", "Muda Selajuh", "Pesawat Tempur", "Kapal Mula", "Omaha Pangkalan", "Helikopter Serang")
24     while True:
25         for i, target_type in enumerate(target_types, 1):
26             print(f"{i}. {target_type}")
27
28         choice = int(input("Pilih jenis target ancaman (masukkan nomor): "))
29         if choice < 1 or choice > 6:
30             print("Pilihan tidak valid. Silakan masukkan nomor antara 1 dan 6.")
31             continue
32
33         distance = float(input("Masukkan jarak target (meter): "))
34         if distance <= 0:
35             print("Jarak tidak valid. Silakan masukkan jarak yang positif.")
36             continue
37
38         target_type = target_types[choice - 1]
39         target_list.add_target(target_type, distance)
40
41     except ValueError:
42         print("Input tidak valid. Silakan masukkan nomor yang benar dan jarak yang benar.")
43
44 while True:
45     add_more = input("Apakah Anda ingin menambahkan target baru (ya/tidak): ").lower()
46     if add_more == "ya":
47         add_target_from_input()
48     elif add_more == "tidak":
49         break
50     else:
51         print("Input tidak valid. Silakan masukkan 'ya' atau 'tidak'")
52
53 target_list.sort_targets()
54
55 detik = 0
56 while target_list.targets:
57     detik += 1
58     print(f"Detik ke {detik}")
59     assignments = []
60     all_intercepted = True
61
62     for weapon in weapons:
63         current_distance = weapon.update_interception()
64         if current_distance is None:
65             print(f"Senjata: {weapon.name} mendakki target, jarak ke target: {current_distance:.2f} meter")
66
67     for target in target_list:
68         if target.intercepted:
69             assignments.append((target.name, target.intercept_weapon, max(target.distance, 0), True))
70         else:
71             weapon_choice = select_weapon(target, weapons)
72             if weapon_choice:
73                 time_to_intercept = weapon_choice.calculate_time_to_intercept(target)
74                 assignments.append((target.name, weapon_choice.name, max(target.distance, 0), time_to_intercept + 1))
75             if time_to_intercept < 1:
76                 all_intercepted = False
77
78     if not all_intercepted:
79         assignments.append((target.name, "Tidak ada senjata yang cocok", max(target.distance, 0), False))
80     all_intercepted = False
81
82 for assignment in assignments:
83     print(f"Target: {assignment[0]} - Senjata: {assignment[1]} (jarak ke kapal: {assignment[2]:.2f} meter) Intercept: {assignment[3]}")
84
85 if all_intercepted:
86     print("Semua target telah ditang.")
87     break
88
89 target_list.move_targets()
90 time.sleep(1)
91

```

Gambar 6. Source code main.py
Sumber: dokumentasi pribadi

Pada kode tersebut, terdapat function `select_weapon`. Fungsi ini memilih senjata yang paling cocok dari daftar senjata untuk diberikan target berdasarkan kriteria kesesuaian dan waktu yang diperlukan untuk mencapai target. Parameter adalah berupa target yang spesifik, daftar senjata. Dalam function tersebut, dilakukan proses mencari senjata yang sesuai dengan target, memilih senjata dengan waktu intercept terpendek, menugaskan senjata ke target jika ada yang sesuai. Function tersebut akan mengembalikan senjata yang dipilih atau None jika tidak ada yang cocok.

Terdapat pula function `add_target_from_input`. Function ini mengumpulkan informasi dari pengguna melalui input untuk menambahkan target baru ke dalam `target_list`. Dalam function ini, dilakukan proses menampilkan jenis target yang tersedia, menerima input jenis dan jarak target, menambahkan target yang valid ke `target_list`.

Terdapat pula input/output exception handling yang bertugas try catch user input dan akan meminta user untuk menginput sesuai format yang benar. Selama inputnya True, akan melakukan looping untuk meminta user menambahkan target sesuai kebutuhan. Kemudian targets tersebut akan di

sorting sesuai pembobotan algoritma greedy yang terdapat dalam Class target.

Pada main.py juga akan dicetak target yang datang menuju ke kapal Arleigh Burke-Class Destroyer dan menampilkan pula senjata pada kapal yang akan meng-*intercept*. Suatu senjata akan di-assign ke target tertentu berdasarkan algoritma Greedy pada class Senjata dan Target yang telah ditentukan.

2. weapon.py:

```

1 class Weapon:
2     def __init__(self, name, rate_of_fire, range_min, range_max, velocity):
3         self.name = name
4         self.rate_of_fire = rate_of_fire
5         self.range_min = range_min
6         self.range_max = range_max
7         self.velocity = velocity
8         self.target_assigned = None
9         self.time_to_intercept = None
10        self.distance_traveled = 0
11
12    def is_suitable(self, target):
13        return self.range_min <= target.distance <= self.range_max
14
15    def assign_target(self, target):
16        self.target_assigned = target
17        self.time_to_intercept = self.calculate_time_to_intercept(target)
18
19    def calculate_time_to_intercept(self, target):
20        return target.distance / self.velocity
21
22    def update_interception(self, time_passed):
23        if self.target_assigned:
24            self.distance_traveled += self.velocity * time_passed
25            self.time_to_intercept -= time_passed
26            current_distance = self.target_assigned.distance - self.distance_traveled
27            if self.time_to_intercept <= 0:
28                self.target_assigned.intercepted = True
29                self.target_assigned.intercept_weapon = self.name
30                self.target_assigned = None
31                self.time_to_intercept = None
32                self.distance_traveled = 0
33            return max(current_distance, 0)
34        return None
35

```

Gambar 7. Source code weapon.py
Sumber: dokumentasi pribadi

Pada class `weapon`, pada inisiasi kelas, terdapat beberapa atribut yaitu `name`, `rate_of_fire`, `range_min`, `range_max`, dan `velocity` dari senjata tersebut. Terdapat beberapa method pada class tersebut antara lain, `__init__`, `is_suitable`, `assign_target`, `calculate_time_to_intercept`, dan `update_interception`. Method `__init__` merupakan konstruktor yang menginisialisasi senjata dengan atribut dasar seperti nama, laju tembak, jangkauan minimum dan maksimum, kecepatan, dan status target yang ditugaskan. Method `is_suitable` merupakan metode yang menentukan apakah senjata cocok untuk menyerang target berdasarkan jarak target. Target dianggap cocok jika jaraknya berada dalam jangkauan operasional senjata (antara `range_min` dan `range_max`). Method `assign_target` merupakan metode untuk menugaskan target ke senjata dan menghitung waktu yang diperlukan untuk intersepsi berdasarkan kecepatan senjata dan jarak target. Method `calculate_time_to_intercept` bertugas menghitung waktu yang diperlukan untuk senjata mencapai target, yang didasarkan pada jarak pembagian kecepatan senjata. Ini memberikan estimasi waktu dalam detik untuk senjata mencapai jarak target. Method `update_interception` memperbarui status intersepsi senjata terhadap target yang ditugaskan setiap kali dipanggil. Metode ini menghitung jarak yang telah ditempuh oleh senjata dalam

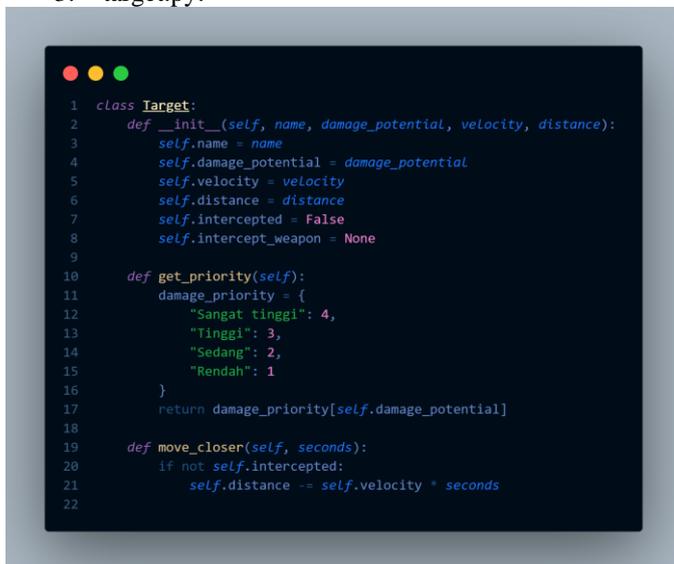
waktu yang ditentukan dan mengurangi waktu yang tersisa untuk intersepsi. Jika waktu intersepsi kurang dari atau sama dengan nol, target dianggap telah dicegat, dan status target diupdate.

Algoritma Greedy dalam konteks pemilihan senjata mengacu pada strategi memilih senjata yang meminimalkan waktu untuk intersepsi dari target yang tersedia. Algoritma Greedy diimplementasikan dalam fungsi `select_weapon` yang tidak disediakan di sini, tetapi berdasarkan penjelasan sebelumnya, saat target muncul atau perlu dicegat, sistem memilih senjata yang tidak hanya cocok berdasarkan jarak tetapi juga yang menawarkan waktu intersepsi terpendek. Ini memastikan bahwa target dicegat secepat mungkin, mengoptimalkan respons terhadap ancaman yang mendesak.

Secara efisiensi, algoritma ini efektif dalam situasi di mana penting untuk segera mengatasi ancaman dengan sumber daya yang tersedia, meskipun mungkin tidak selalu memberikan solusi paling optimal jika dilihat dari perspektif penggunaan sumber daya secara keseluruhan atau strategi jangka panjang.

Secara keseluruhan, kelas `Weapon` dan algoritma pemilihan senjata menggunakan pendekatan pragmatis dan langsung untuk mengelola pertahanan terhadap berbagai jenis ancaman, dengan menggunakan metode yang serakah untuk mengoptimalkan waktu respons dalam situasi krisis.

3. target.py:



```
1 class Target:
2     def __init__(self, name, damage_potential, velocity, distance):
3         self.name = name
4         self.damage_potential = damage_potential
5         self.velocity = velocity
6         self.distance = distance
7         self.intercepted = False
8         self.intercept_weapon = None
9
10    def get_priority(self):
11        damage_priority = {
12            "Sangat tinggi": 4,
13            "Tinggi": 3,
14            "Sedang": 2,
15            "Rendah": 1
16        }
17        return damage_priority[self.damage_potential]
18
19    def move_closer(self, seconds):
20        if not self.intercepted:
21            self.distance -= self.velocity * seconds
22
```

Gambar 8. Source code target.py
Sumber: dokumentasi pribadi

Pada class `Target`, terdapat inisiasi class yang memiliki atribut `name`, `damage_potential`, `velocity`, dan `distance`. Terdapat pula beberapa method dalam class tersebut seperti `__init__`, `get_priority`, dan `move_closer`. Method `__init__` menerima nama target, potensi kerusakan (`damage_potential`), kecepatan (`velocity`), dan jarak (`distance`). Dalam `__init__` dilakukan inisialisasi atribut target, seperti nama, potensi kerusakan, kecepatan, jarak, status intersepsi, dan senjata yang

digunakan untuk intersepsi (jika ada). Akan direturn objek target dengan atribut yang telah diinisialisasi.

Terdapat pula method `get_priority` yang bertugas menentukan prioritas target berdasarkan potensi kerusakannya. Method ini menggunakan kamus (dictionary) untuk memetakan potensi kerusakan ke nilai prioritas numerik: "Sangat tinggi" = 4, "Tinggi" = 3, "Sedang" = 2, "Rendah" = 1. Pada method ini akan di return nilai prioritas numerik target.

Terdapat pula method `move_closer` yang menerima argumen waktu dalam detik (`seconds`). Dalam method ini terjadi penghitungan jarak target berdasarkan kecepatan dan waktu yang berlalu, kecuali target telah dicegat. Akan direturn jarak target yang diperbarui.

Pada class ini, digunakan Algoritma Greedy dalam penentuan prioritas target dengan mengubah potensi kerusakan menjadi numerik. Algoritma "greedy" dalam konteks pemilihan dan penanganan target merujuk pada strategi memilih target yang harus dicegat terlebih dahulu berdasarkan prioritas tertinggi. Setiap target diberi prioritas berdasarkan potensi kerusakannya menggunakan metode `get_priority`. Target dengan potensi kerusakan yang lebih tinggi diberi nilai prioritas yang lebih tinggi. Dalam skenario di mana beberapa target ada, sistem memilih target dengan prioritas tertinggi terlebih dahulu untuk dicegat. Hal ini memastikan bahwa ancaman yang paling berbahaya ditangani terlebih dahulu, mengoptimalkan penggunaan sumber daya pertahanan untuk meminimalkan risiko kerusakan besar.

Algoritma greedy ini sangat efisien dalam situasi di mana perlu merespons dengan cepat terhadap ancaman yang berpotensi tinggi. Meskipun algoritma ini bersifat lokal optimal (memilih solusi terbaik pada setiap langkah), ia tidak selalu menghasilkan solusi global optimal (solusi terbaik secara keseluruhan).

4. target_list.py:

```

1 from target import Target
2
3 class TargetList:
4     def __init__(self):
5         self.targets = []
6         self.target_count = {
7             "Rudal Balistik": 0,
8             "Rudal Jelajah": 0,
9             "Pesawat Tempur": 0,
10            "Kapal Musuh": 0,
11            "Drone Pengintai": 0,
12            "Helikopter Serang": 0
13        }
14
15     def add_target(self, target_type, distance):
16         self.target_count[target_type] += 1
17         name_map = {
18             "Rudal Balistik": "RB",
19             "Rudal Jelajah": "RJ",
20             "Pesawat Tempur": "PT",
21             "Kapal Musuh": "KM",
22             "Drone Pengintai": "DP",
23             "Helikopter Serang": "HS"
24         }
25         velocity_map = {
26             "Rudal Balistik": 2000,
27             "Rudal Jelajah": 986,
28             "Pesawat Tempur": 772,
29             "Kapal Musuh": 13,
30             "Drone Pengintai": 56,
31             "Helikopter Serang": 84
32         }
33         damage_map = {
34             "Rudal Balistik": "Sangat tinggi",
35             "Rudal Jelajah": "Tinggi",
36             "Pesawat Tempur": "Tinggi",
37             "Kapal Musuh": "Sedang",
38             "Drone Pengintai": "Rendah",
39             "Helikopter Serang": "Rendah"
40         }
41         name = f"{name_map[target_type]}{self.target_count[target_type]}"
42         velocity = velocity_map[target_type]
43         damage_potential = damage_map[target_type]
44         target = Target(name, damage_potential, velocity, distance)
45         self.targets.append(target)
46
47     def sort_targets(self):
48         self.targets.sort(key=lambda target: target.get_priority(), reverse=True)
49
50     def move_targets(self, seconds):
51         for target in self.targets:
52             target.move_closer(seconds)
53
54     def __iter__(self):
55         return iter(self.targets)
56
57
58

```

Gambar 9. Source code target_list.py
Sumber: dokumentasi pribadi

Pada kelas TargetList terdapat beberapa method dan atribut. Terdapat method init yang menginisialisasi objek TargetList dengan dua atribut, yaitu targets dan target_count. Target merupakan daftar kosong untuk menyimpan objek target. target_count merupakan dictionary yang melacak jumlah setiap jenis target yang ditambahkan. Akan direturn objek TargetList yang diinisialisasi.

Terdapat pula method add_target yang menerima parameter jenis target (target_type) dan jarak (distance). Method ini menambah jumlah target dari jenis yang ditentukan dalam target_count, menggunakan kamus (name_map, velocity_map, damage_map) untuk memetakan jenis target ke atribut spesifik (nama singkatan, kecepatan, dan potensi kerusakan), membuat objek Target baru dengan nama yang dihasilkan, potensi kerusakan, kecepatan, dan jarak, menambahkan objek target baru ke dalam daftar targets. Method ini memasukkan target ke dalam list.

Terdapat pula method sort_targets yang mengurutkan daftar targets berdasarkan prioritas (diperoleh dari metode

get_priority pada objek Target), dengan prioritas tertinggi di urutan pertama.

Terdapat method move_targets yang menerima parameter waktu dalam detik (seconds). Method ini memindahkan setiap target dalam daftar targets lebih dekat berdasarkan kecepatan target dan waktu yang berlalu.

IV. ANALISIS

```

Apakah Anda ingin menambahkan target baru? (ya/tidak): ya
1. Rudal Balistik
2. Rudal Jelajah
3. Pesawat Tempur
4. Kapal Musuh
5. Drone Pengintai
6. Helikopter Serang
Pilih jenis target ancaman (masukkan nomor): 1
Masukkan jarak target (meter): 20
Apakah Anda ingin menambahkan target baru? (ya/tidak): ya
1. Rudal Balistik
2. Rudal Jelajah
3. Pesawat Tempur
4. Kapal Musuh
5. Drone Pengintai
6. Helikopter Serang
Pilih jenis target ancaman (masukkan nomor): 2
Masukkan jarak target (meter): 100
Apakah Anda ingin menambahkan target baru? (ya/tidak): ya
1. Rudal Balistik
2. Rudal Jelajah
3. Pesawat Tempur
4. Kapal Musuh
5. Drone Pengintai
6. Helikopter Serang
Pilih jenis target ancaman (masukkan nomor): 3
Masukkan jarak target (meter): 200
Apakah Anda ingin menambahkan target baru? (ya/tidak): ya
1. Rudal Balistik
2. Rudal Jelajah
3. Pesawat Tempur
4. Kapal Musuh
5. Drone Pengintai
6. Helikopter Serang
Pilih jenis target ancaman (masukkan nomor): 4
Masukkan jarak target (meter): 80
Apakah Anda ingin menambahkan target baru? (ya/tidak): ya
1. Rudal Balistik
2. Rudal Jelajah
3. Pesawat Tempur
4. Kapal Musuh
5. Drone Pengintai
6. Helikopter Serang
Pilih jenis target ancaman (masukkan nomor): 5
Masukkan jarak target (meter): 250
Apakah Anda ingin menambahkan target baru? (ya/tidak): ya
1. Rudal Balistik
2. Rudal Jelajah
3. Pesawat Tempur
4. Kapal Musuh
5. Drone Pengintai
6. Helikopter Serang
Pilih jenis target ancaman (masukkan nomor): 6
Masukkan jarak target (meter): 190
Apakah Anda ingin menambahkan target baru? (ya/tidak): tidak
Pilih jenis target ancaman (masukkan nomor): 4
Masukkan jarak target (meter): 80
Apakah Anda ingin menambahkan target baru? (ya/tidak): ya
1. Rudal Balistik
2. Rudal Jelajah
3. Pesawat Tempur
4. Kapal Musuh
5. Drone Pengintai
6. Helikopter Serang
Pilih jenis target ancaman (masukkan nomor): 5
Masukkan jarak target (meter): 250
Apakah Anda ingin menambahkan target baru? (ya/tidak): ya
1. Rudal Balistik
2. Rudal Jelajah
3. Pesawat Tempur
4. Kapal Musuh
5. Drone Pengintai
6. Helikopter Serang
Pilih jenis target ancaman (masukkan nomor): 6
Masukkan jarak target (meter): 190
Apakah Anda ingin menambahkan target baru? (ya/tidak): tidak
Detik ke-1
Target: RB1 -> Senjata: M242 Bushmaster (Jarak ke kapal: 20.00 meter) Intercept: True
Target: RJ1 -> Senjata: M242 Bushmaster (Jarak ke kapal: 100.00 meter) Intercept: True
Target: PT1 -> Senjata: M242 Bushmaster (Jarak ke kapal: 200.00 meter) Intercept: True
Target: KM1 -> Senjata: M242 Bushmaster (Jarak ke kapal: 80.00 meter) Intercept: True
Target: DP1 -> Senjata: M242 Bushmaster (Jarak ke kapal: 250.00 meter) Intercept: True
Target: HS1 -> Senjata: M242 Bushmaster (Jarak ke kapal: 190.00 meter) Intercept: True
Semua target telah diogot.
PS E:\Vondrive - Institut Teknologi Bandung\COLLEGE\SEKESTER 4\Strategi Algoritma\Ukalah\code\

```

Uji kasus ini menunjukkan bagaimana sistem simulasi AEGIS Combat System bekerja dalam memilih dan mengintersepsi target berdasarkan input pengguna. Berikut adalah langkah-langkah dan penjelasan dari setiap bagian dalam uji kasus tersebut: 1. Input dan Penambahan Target
Langkah 1: Menambahkan target pertama

- Input:
 - Jenis target: Kapal Musuh (nomor 4)
 - Jarak: 80 meter
- Output: Target "KM1" ditambahkan dengan jarak 80 meter.

Langkah 2: Menambahkan target kedua

- Input:
 - Jenis target: Drone Pengintai (nomor 5)
 - Jarak: 250 meter

- Output: Target "DP1" ditambahkan dengan jarak 250 meter.

Langkah 3: Menambahkan target ketiga

- Input:
 - Jenis target: Helikopter Serang (nomor 6)
 - Jarak: 190 meter
- Output: Target "HS1" ditambahkan dengan jarak 190 meter.

Penentuan Prioritas dan Intersepsi adalah sebagai berikut.

Setelah target ditambahkan, sistem menentukan prioritas berdasarkan potensi kerusakan yang dapat ditimbulkan oleh setiap target. Prioritas dihitung sebagai berikut:

- Rudal Balistik: Sangat tinggi
- Rudal Jelajah: Tinggi
- Pesawat Tempur: Tinggi
- Kapal Musuh: Sedang
- Drone Pengintai: Rendah
- Helikopter Serang: Rendah

Namun, dalam kasus ini, hanya ada jenis target Kapal Musuh, Drone Pengintai, dan Helikopter Serang.

Proses Intersepsi adalah sebagai berikut.

Detik ke-1: Simulasi dimulai dan sistem melakukan intersepsi pada semua target yang ada. Berikut adalah detail intersepsi:

- Target "RB1": Senjata: M242 Bushmaster, Jarak ke kapal: 20.00 meter, Intercept: True
- Target "RJ1": Senjata: M242 Bushmaster, Jarak ke kapal: 100.00 meter, Intercept: True
- Target "PT1": Senjata: M242 Bushmaster, Jarak ke kapal: 200.00 meter, Intercept: True
- Target "KM1": Senjata: M242 Bushmaster, Jarak ke kapal: 80.00 meter, Intercept: True
- Target "DP1": Senjata: M242 Bushmaster, Jarak ke kapal: 250.00 meter, Intercept: True
- Target "HS1": Senjata: M242 Bushmaster, Jarak ke kapal: 190.00 meter, Intercept: True

Semua target berhasil dicegat pada detik pertama simulasi. Hal ini menunjukkan bahwa sistem AEGIS Combat System berhasil mengalokasikan senjata yang tepat untuk setiap target, mengoptimalkan waktu intersepsi menggunakan algoritma greedy. Algoritma ini memastikan bahwa setiap ancaman ditangani dengan prioritas tertinggi secepat mungkin.

Uji kasus ini berhasil menunjukkan bagaimana sistem AEGIS Combat System menggunakan algoritma greedy untuk menentukan prioritas target dan mengalokasikan senjata secara efektif. Semua target berhasil dicegat dalam waktu singkat, membuktikan efisiensi dan efektivitas dari pendekatan yang digunakan. Implementasi ini penting dalam skenario nyata di mana respons cepat dan akurat sangat krusial untuk keberhasilan misi dan keselamatan kapal serta personel.

V. KESIMPULAN

Makalah ini telah mengeksplorasi penerapan algoritma greedy dalam AEGIS Combat System yang digunakan pada kapal perusak kelas Arleigh Burke milik US Navy. Algoritma greedy digunakan untuk mengoptimalkan

proses deteksi dan intersepsi target dengan cara memprioritaskan target berdasarkan tingkat ancaman. Dalam simulasi, pendekatan ini terbukti efektif dalam mengatasi ancaman yang paling berbahaya terlebih dahulu, memastikan perlindungan yang lebih baik untuk kapal dan personel. Implementasi algoritma ini memungkinkan sistem AEGIS untuk melakukan penilaian dan respons terhadap ancaman secara real-time, meningkatkan efisiensi operasional. Meskipun demikian, penting untuk diakui bahwa solusi greedy tidak selalu menjamin solusi global optimal, namun pendekatan ini tetap memberikan hasil yang memadai dalam situasi darurat yang memerlukan respons cepat. Penelitian lebih lanjut disarankan untuk menguji integrasi algoritma ini dengan sistem lain dan mengeksplorasi potensi peningkatan performa melalui kombinasi dengan teknik optimasi lainnya.

VI. UCAPAN TERIMA KASIH

Penulis berterima kasih kepada Tuhan yang Maha Esa karena telah memberikan kemudahan dalam penulisan makalah ini. Penulis juga ingin berterima kasih kepada Dr. Nur Ulva Maulidevi, S.T., M.Sc. selaku dosen dan pembimbing penulis dalam mata kuliah IF2211 Strategi Algoritma. Selain itu, penulis juga ingin berterima kasih kepada keluarga dan kawan-kawan yang sudah mendukung penulis dalam menyelesaikan penulisan makalah ini. Akhir kata, penulis juga ingin berterima kasih kepada kawan dekat penulis, Ahmad Syafiq Rofii yang telah menginspirasi hidup serta memotivasi untuk membuat sebuah makalah penelitian tentang militer.

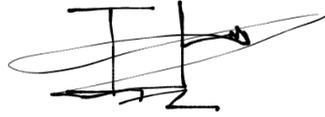
REFERENCES

- [1] <https://informatika.stei.itb.ac.id/>
- [2] <https://www.intechopen.com/chapters/71902>
- [3] Threston, Joseph T. 2009. "The Story of AEGIS: The AEGIS Combat System." American Society of Naval Engineers 121, no. 3(October): 109–132.
- [4] Chance, Kyle A. Naval combat systems product line economics: Extending the constructive product line investment model for the aegis combat system. Diss. Monterey, CA; Naval Postgraduate School, 2019.
- [5] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.
- [6] K. Elissa, "Title of paper if known," unpublished.
- [7] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
- [8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].

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, 12 Juni 2024



Eduardus Alvito Kristiadi (13522004)