

***Dogfight* dari Sudut Pandang Teori Permainan**

Rafi Prayoga Dhenanta¹, Aditya Purwa Santika²

rafibinpepe@gmail.com¹

^{1,2}Institut Teknologi Bandung

Informasi Artikel

Diterima : 10 Jun 2025

Direvisi : 21 Jul 2025

Disetujui : 3 Agu 2025

Kata Kunci

dogfight, teori permainan, permainan berbentuk ekstensif, komputasi

Abstrak

Salah satu skenario dalam konflik yang melibatkan komponen udara adalah pertarungan udara-ke-udara jarak dekat atau yang biasa disebut dengan *dogfight*. Dalam penelitian ini *dogfight* dikaji dari sudut pandang teori permainan. Tujuan dari penelitian ini adalah untuk mencari strategi dalam *dogfight*. Penelitian dilakukan dengan memodelkan *dogfight* ke dalam permainan berbentuk ekstensif lalu mensimulasikan model yang dibuat secara komputasi. Berdasarkan simulasi, model yang dikembangkan dalam penelitian ini meningkatkan jumlah kemenangan seorang pemain dengan signifikan.

Keywords

dogfight, game theory, extensive-form-games, computation

Abstract

Dogfight is one of many scenarios happening in a battle for air-superiority. This research delves deeper into dogfight, using the perspectives of game theory. The purpose of this research is to model a strategy that can be used in a dogfight. This research models dogfight into game theory's extensive-form-games and then simulates the model computationally. From the simulation, the model developed in this research increases the winning rate of a certain player significantly.

A. Pendahuluan

Umat manusia dan perang selalu hidup berdampingan, baik di zaman pra-aksara maupun di zaman modern. Sebuah pertanyaan yang sering muncul mengenai perang adalah pihak manakah yang akan memenangkannya. Berdasarkan penelitian Saunders dan Souva (2020)[1], faktor yang berkorelasi paling signifikan terhadap hasil akhir dari suatu peperangan adalah kekuatan udara. Pihak yang memiliki kekuatan udara lebih besar (*air superiority*) akan bebas meluncurkan serangan efektif dan rendah risiko jauh di dalam jantung pertahanan lawan. Contoh skenario kekuatan udara yang mempengaruhi hasil perang adalah bom atom yang dijatuhkan oleh pesawat sekutu membuat Jepang menyerah dalam Perang Dunia II dan pemboman udara (*aerial bombing*) yang digunakan oleh pasukan koalisi dalam Perang Teluk mempermudah jalan kemenangan pasukan koalisi.

Dalam memperebutkan superioritas udara suatu wilayah, terdapat dua pola pertempuran yang terjadi, yakni pertempuran udara-ke-udara dan pertempuran udara-ke-darat. Salah satu contoh pertempuran udara-ke-udara adalah pertempuran antar pesawat tempur sementara pertempuran udara-ke-darat biasanya terjadi antara pesawat tempur melawan sistem pertahanan udara. Dalam artikel ini peneliti hanya akan membahas mengenai pertempuran udara-ke-udara karena secara pemodelan matematis, pertempuran udara-ke-darat adalah kasus khusus dari pertempuran udara-ke-udara, yakni dengan memodelkan kecepatan salah satu pihak (pihak darat) bernilai nol.

Skenario ideal dari pertempuran udara-ke-udara saat ini adalah model pertempuran BVR (*beyond visual range*), yakni kedua pesawat saling bertempur dari jarak jauh mengandalkan peluru kendali dan sistem elektronika. Meskipun demikian, pelatihan pertempuran udara-ke-udara jarak dekat (*dogfight*) tetap dijalani oleh para pilot. Terlebih lagi, skenario *dogfight* masih muncul di konflik terkini, seperti dalam konflik India-Pakistan yang melibatkan *J-10* dan *Rafale* maupun konflik Rusia-Ukraina. Di masa depan, bukan tidak mungkin skenario *dogfight* antar *drone* akan terjadi melihat perkembangan *AI* dan *drone* yang cukup signifikan semenjak konflik Rusia-Ukraina.

Fokus dari penelitian ini adalah memodelkan dan mensimulasikan *dogfight* berdasarkan teori permainan. Teori permainan adalah cabang dari ilmu matematika yang membahas interaksi antar pembuat keputusan. Dalam *dogfight*, setiap pilot diharuskan untuk membuat keputusan. Dalam proses pengambilan keputusannya, seorang pilot pasti melihat keputusan/langkah yang diambil oleh lawan untuk menentukan langkah terbaik yang harus diambilnya. Akibatnya, muncullah suatu interaksi antar seorang pilot dengan lawannya. Keberadaan interaksi inilah yang menjadi syarat minimal untuk memodelkan suatu kejadian menggunakan teori permainan.

Pemodelan *dogfight* secara matematis telah beberapa kali dilakukan. Beberapa contoh di antaranya adalah pemodelan proses pengambilan keputusan menggunakan *approximate dynamic programming*^[2] maupun menggunakan *machine learning*^[3]. Model-model yang telah ada membahas proses pengambilan keputusan dengan detail, seperti mengembangkan sebuah *reward function* dan *scheme* komputasi untuk mencari dan memilih keputusan terbaik. Berbeda dari penelitian yang telah ada, penelitian ini lebih terfokus pada jalannya permainan.

Artinya, artikel ini tidak membahas apakah suatu fungsi *reward* yang dibentuk merupakan fungsi yang paling efektif atau apakah metode komputasi yang dilakukan telah optimal, tetapi artikel ini mengembangkan ‘fungsi *reward*’ sederhana yang kemudian disimulasikan (menggunakan prinsip-prinsip teori permainan) dalam berbagai skenario sehingga didapatkan seorang ‘pemenang’ dari setiap skenarionya.

Karena fokus dari artikel ini adalah jalannya permainan, maka digunakan model permainan dinamis yaitu model permainan yang berjalan seiring waktu. Terdapat dua jenis permainan dinamis, yaitu permainan berbentuk ekstensif (*extensive-form games*) dan permainan multi-tahap (*multi-stage games*). Perbedaan antara keduanya terletak pada waktu pengambilan aksinya dengan aksi dalam permainan berbentuk ekstensif diambil secara tidak simultan. Contoh permainan berbentuk ekstensif adalah catur sementara contoh permainan multi-tahap adalah permainan batu-gunting-kertas yang diulang sebanyak lima kali. Dalam penelitian ini hanya akan dibahas model permainan berbentuk ekstensif dari *dogfight*.

Permainan yang dibentuk berdasarkan model permainan berbentuk ekstensif selanjutnya disimulasikan secara komputasi. Hasil dari simulasi selanjutnya dijabarkan dan dianalisis.

Dari latar belakang di atas didapat tujuan dari penelitian ini adalah untuk memodelkan *dogfight* secara teori permainan serta merancang strategi optimal untuk setiap pemain. Adapun kegunaan dari hasil penelitian ini adalah untuk dijadikan referensi awal dalam automasi komponen-komponen kekuatan udara (seperti *drone* dan *UAV*). Secara akademis, hasil penelitian ini dapat digunakan untuk memperluas pembahasan mengenai teori permainan, terutama dalam ranah terapan.

B. Metode Penelitian

Penelitian ini terdiri dari dua tahap, yakni tahap pemodelan dan tahap simulasi – analisis. Tahap pemodelan berisi tentang proses abstraksi *dogfight* menjadi persamaan-persamaan matematika sementara tahap simulasi – analisis berisi simulasi model yang didapatkan dalam tahap pemodelan lalu hasil simulasinya akan dianalisis secara konsep-konsep teori permainan.

Dalam tahap pemodelan, objek yang digunakan sebagai model adalah *dogfight*. Mengingat data dari *dogfight* ‘nyata’ tidak tersedia secara umum maka dalam penelitian ini digunakan data *dogfight* berdasarkan observasi dalam gim *Modern Air Combat: Team Match*. Adapun objek observasi ini karena gim yang dimaksud telah dirasa cukup untuk merepresentasikan *dogfight*, data-data dari gim tersebut mudah diambil, dan gim tersebut sangat terjangkau serta ringan untuk dimainkan.

Dalam tahap simulasi – analisis, model yang didapat terlebih dahulu disimulasikan menggunakan program komputer. Dalam setiap langkah dalam setiap skenario simulasinya, aksi seorang pemain dipilih menggunakan algoritma *minimax*. Karena algoritma *minimax* menggunakan struktur pohon maka program untuk simulasi dibangun menggunakan bahasa pemrograman *C++* untuk efisiensi memori dan waktu komputasinya. Simulasi dilakukan dalam beberapa kasus besar, yakni

1. seluruh pemain memilih aksinya tidak berdasarkan model yang terbentuk di tahap pemodelan,
2. salah satu pemain memilih aksinya berdasarkan model sementara pemain lainnya tidak, dan
3. seluruh pemain memilih aksinya berdasarkan model.

Simulasi dari ketiga kasus besar di atas bertujuan untuk dianalisis apakah model yang didapatkan sudah cukup baik. Analisis kemudian dilanjutkan secara matematis untuk mengetahui ada atau tidaknya strategi khusus yang dapat diambil dalam *dogfight* sehingga kemungkinan menang seorang pemain akan meningkat.

C. Hasil dan Pembahasan

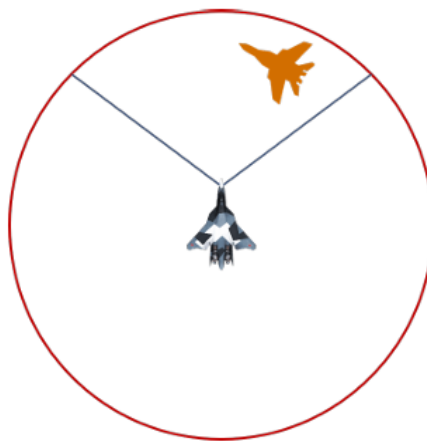
Bab ini terbagi menjadi dua bagian, yaitu bagian pemodelan yang berisi pembentukan model dari objek observasi, yaitu *dogfight* dalam gim *Modern Air Combat: Team Match*, serta bagian simulasi dan analisisnya.

C1. Pemodelan

Tahapan pertama yang dilakukan dalam pemodelan adalah simplifikasi model. Hal ini dikarenakan *dogfight* yang diamati dari gim masih terlalu kompleks karena berada dalam 'ruang kontinu'. Selain itu proses simplifikasi juga berfungsi untuk mengurangi pengaruh *nature* (kejadian acak) dalam permainan yang dikembangkan.

Simplifikasi yang dipilih adalah,

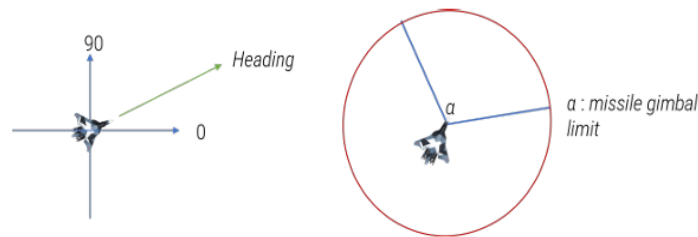
1. *dogfight* dilakukan dalam 2 dimensi, yakni di bidang xy
2. aksi yang dilakukan dibatasi. Sebagai ilustrasi misalkan suatu pesawat belok kanan selama 0.312 detik maka aksinya hanya dianggap belok kanan,
3. *dogfight* yang ditinjau hanya terdiri dari 2 pemain,
4. permainan berakhir ketika minimal salah satu pesawat (pemain) mengunci pesawat lain.



Gambar 1. Pesawat abu-abu mengunci pesawat jingga

Dalam teori permainan, sebuah model permainan memiliki 3 buah komponen, yaitu pemain, aksi, dan fungsi bayaran (*payoff function*). Pertama akan dilakukan abstraksi terhadap pemain. Dalam penelitian ini, istilah pemain melambangkan pesawat tempur di dunia nyata. Berdasarkan simplifikasi, pemain yang akan ditinjau ada dua, yaitu pemain 1 dan pemain 2 yang dilambangkan dengan P_1 dan P_2 . Dari simplifikasi, permainan akan berakhir ketika salah satu pemain mengunci pemain lain. Akibatnya, hal-hal yang berpengaruh terhadap proses penguncian akan diabstraksi sebagai *state* dari seorang pemain. Didapatkan model dari pemain untuk $i \in \{1, 2\}$ adalah,

$P_i = (x_i, y_i, heading_i, speed_i, turn_rate_i, firing_range_i, missile_gimbal_limit_i, lock_i)$ dengan x, y menyatakan posisi pesawat dalam koordinat kartesian-xy, *heading* menyatakan arah pesawat dengan 0° adalah arah yang sejajar dengan sumbu-x positif yang diukur dari posisi pesawat di koordinat kartesian, *speed* menyatakan kecepatan pesawat, *turn_rate* menyatakan besarnya perubahan sudut pesawat ketika berbelok, *firing_range* menyatakan jarak maksimal agar suatu pesawat dapat dikunci oleh pesawat lain, *missile_gimbal_limit* menyatakan posisi sudut maksimal agar suatu pemain dapat mengunci pemain lain, dan *lock* adalah variabel yang menyatakan kondisi akhir permainan.



Gambar 2. Ilustrasi *heading* dan *missile_gimbal_limit*

Selanjutnya akan dimodelkan aksi pemain. Dalam sebuah *dogfight*, kata aksi mengacu pada manuver-maneuver yang mungkin dilakukan oleh seorang pilot. Inti dari manuver *dogfight* adalah menggerakkan pesawat, yang jika ditinjau dari teori permainan berarti mengubah *state* pemain untuk *timestep* berikutnya. Perlu diingat bahwa aksi-aksi seperti menembakkan rudal, mengeluarkan *flare*, *ejection* tidak dibahas dalam penelitian ini mengingat kondisi akhir permainan hanya dengan mengunci pemain lain.

Gerakan atau perubahan posisi pesawat di bidang kartesian-xy dipengaruhi oleh dua faktor, yaitu kecepatan pesawat dan *heading* pesawat. Diskritisasi aksi dalam penelitian ini dipilih berdasarkan kombinasi dua faktor tersebut, yaitu dari faktor kecepatan dipilih tiga aksi yaitu *normal*, *accelerating* yang berarti penambahan kecepatan, dan *braking* yang berarti pengurangan kecepatan sementara dari *heading* dipilih tiga aksi yaitu *straight*, *right*, dan *left*.

Berikut data kecepatan yang diambil dari gim,

Tabel 1. Data kecepatan pesawat

Kondisi	FA-33	FA-Y26
<i>Normal</i>	1220	1150
<i>Accelerating</i>	2440	2290
<i>Braking</i>	580	590

Dari data kecepatan di atas dapat diaproksimasi kecepatan pesawat ketika *accelerating* adalah 2 kali kecepatan pesawat ketika dalam posisi *normal* sementara dalam kondisi *braking* kecepatan pesawat menjadi setengah dari kecepatan pesawat dalam kondisi *normal*. Selanjutnya diberikan data mengenai pola belok pesawat, yaitu berapa lama (detik) pesawat menyelesaikan satu kali putaran 360 derajat.

Tabel 2. Data pola belok pesawat

FA-33			FA-Y26		
Normal	Accelerating	Braking	Normal	Accelerating	Braking
6.73	7.03	5.68	7.33	7.35	6.04
6.76	6.64	5.74	7.30	7.15	6.00
6.78	6.74	5.32	7.16	7.27	6.03
6.80	6.69	5.41	7.33	7.19	5.93
6.73	6.76	5.74	7.24	7.33	5.93

Dari data di atas didapatkan *turn rate* pesawat dalam mode *normal* dan *accelerating* dapat dianggap sama sementara dalam kondisi *braking*, *turn rate* pesawat berada di angka sekitar 1.25 kali kondisi *normal* atau dengan kata lain waktu yang diperlukan pesawat untuk menyelesaikan satu putaran dalam kondisi *braking* adalah 0.8 kali lebih cepat daripada dalam kondisi *normal* dan *accelerating*. Didapatkan *scheme* berikut untuk meng-*update state* dari para pemain,

1. *Normal straight*
 $x_i^{t+1} = x_i^t + speed_i \cos(heading_i)$
 $y_i^{t+1} = y_i^t + speed_i \sin(heading_i)$
2. *Accelerating straining*
 $x_i^{t+1} = x_i^t + 2 speed_i \cos(heading_i)$
 $y_i^{t+1} = y_i^t + 2 speed_i \sin(heading_i)$
3. *Braking straight*
 $x_i^{t+1} = x_i^t + \frac{1}{2} speed_i \cos(heading_i)$
 $y_i^{t+1} = y_i^t + \frac{1}{2} speed_i \sin(heading_i)$
4. *Normal right*
 $x_i^{t+1} = x_i^t + speed_i \cos(heading_i - turn_rate_i)$
 $y_i^{t+1} = y_i^t + speed_i \sin(heading_i - turn_rate_i)$
 $heading_i = heading_i - turn_rate_i \pmod{360}$
5. *Accelerating right*
 $x_i^{t+1} = x_i^t + 2 speed_i \cos(heading_i - turn_rate_i)$
 $y_i^{t+1} = y_i^t + 2 speed_i \sin(heading_i - turn_rate_i)$
 $heading_i = heading_i - turn_rate_i \pmod{360}$
6. *Braking right*
 $x_i^{t+1} = x_i^t + \frac{1}{2} speed_i \cos(heading_i - turn_rate_i)$
 $y_i^{t+1} = y_i^t + \frac{1}{2} speed_i \sin(heading_i - turn_rate_i)$
 $heading_i = heading_i - 1.25 turn_rate_i \pmod{360}$
7. *Normal left*
 $x_i^{t+1} = x_i^t + speed_i \cos(heading_i + turn_rate_i)$
 $y_i^{t+1} = y_i^t + speed_i \sin(heading_i + turn_rate_i)$
 $heading_i = heading_i + turn_rate_i \pmod{360}$

8. *Accelerating left*

$$x_i^{t+1} = x_i^t + 2 \text{ speed}_i \cos(\text{heading}_i + \text{turn_rate}_i)$$

$$y_i^{t+1} = y_i^t + 2 \text{ speed}_i \sin(\text{heading}_i + \text{turn_rate}_i)$$

$$\text{heading}_i = \text{heading}_i + \text{turn_rate}_i \pmod{360}$$

9. *Braking left*

$$x_i^{t+1} = x_i^t + \frac{1}{2} \text{ speed}_i \cos(\text{heading}_i + \text{turn_rate}_i)$$

$$y_i^{t+1} = y_i^t + \frac{1}{2} \text{ speed}_i \sin(\text{heading}_i + \text{turn_rate}_i)$$

$$\text{heading}_i = \text{heading}_i + 1.25 \text{ turn_rate}_i \pmod{360}$$

Komponen permainan terakhir yang akan dimodelkan adalah fungsi bayaran untuk semua pemain. Perlu diingat bahwa pemenang adalah pemain yang mengunci pemain lain, sehingga fungsi bayaran dapat dibentuk berdasarkan jarak dan posisi relatif kedua pemain^[2].

Misalkan f_i adalah fungsi bayaran pemain i , g adalah fungsi berparameter jarak dan h adalah fungsi berparameter posisi relatif, didapat formulasi fungsi bayarannya,

$$f_i(P_i, P_{-i}) = g(P_i, P_{-i}) \times h(P_i, P_{-i})$$

Definisikan d sebagai jarak kedua pemain, s_1 dan s_2 masing-masing adalah posisi relatif P_2 dari sudut pandang P_1 dan posisi relatif P_1 dari sudut pandang P_2 . Sehingga di ruang kartesian-xy didapat,

$$d = ((x_1 - x_2)^2 + (y_1 - y_2)^2)^{1/2}$$

$$s_1 = \cos^{-1}((\cos(\text{heading}_1) \times (x_2 - x_1) + \sin(\text{heading}_1) \times (y_2 - y_1)) / d)$$

$$s_2 = \cos^{-1}((\cos(\text{heading}_2) \times (x_1 - x_2) + \sin(\text{heading}_2) \times (y_1 - y_2)) / d)$$

Selanjutnya adalah pemodelan fungsi berparameter jarak. Terdapat tiga hasil basis pemodelan untuk fungsi berparameter jarak, yakni faktor jarak tidak berpengaruh secara signifikan terhadap pemilihan aksi ketika jarak kedua pemain masih di luar *firing range*, di dalam *firing range* terdapat *optimum range*, r_0 , yang diasumsikan setengah dari *firing range*, dan nilai fungsi berparameter jarak akan menurun ketika jarak kedua pemain semakin berdekatan.

Misalkan r_1, r_2 adalah akar dari persamaan $-d^2 + 2r_0d - k/d = 0$ dengan k adalah sebuah konstanta bernilai riil, maka dapat dipilih fungsi berparameter jarak,

$$g(d) = -d^2 + 2r_0d \text{ jika } 0 \leq d < \max(r_1, r_2)$$

atau

$$g(d) = k/d \text{ untuk kasus lainnya}$$

Selanjutnya adalah pemodelan fungsi berparameter posisi relatif pesawat. Beberapa istilah yang umum digunakan untuk menyatakan posisi relatif pesawat dalam *dofight* adalah *hot*, *flank*, dan *cold*. Di Gambar 3, misalkan P_1 adalah pesawat abu-abu dan P_2 adalah pesawat biru. Dari kiri didapat s_1 *flank* sementara s_2 *hot*, s_1 *flank* sementara s_2 *flank*, dan s_1 *hot* sementara s_2 *cold*.



Gambar 3. Posisi relatif pesawat

Untuk membangun fungsi bayarannya diperlukan preferensi pemain. Berikut tabel yang menyatakan urutan preferensi para pemain,

Tabel 3. Preferensi pemain

s_1	s_2	Preferensi P_1	Preferensi P_2
hot	hot	6	4
hot	flank	2	8
hot	cold	1	9
cold	hot	9	1
cold	flank	7	3
cold	cold	4	6
flank	hot	8	2
flank	flank	5	5
flank	cold	3	7

Dalam penelitian ini dipilih sebuah fungsi yang dapat mewakili preferensi di atas yakni,

$$h(s_i, s_{-i}) = (-s_i + s_{-i})/180$$

Lema 1 Fungsi h membuat permainan memiliki bentuk *zero-sum games*

Bukti $h(s_i, s_{-i}) + h(s_{-i}, s_i) = (-s_i + s_{-i})/180 + (-s_{-i} + s_i)/180 = 0$

Akibat dari *Lema 1* adalah permainan bersifat *zero-sum games*, yakni jika salah satu pihak mendapatkan keuntungan maka pihak lain mendapatkan kerugian yang 'sepadan' dengan keuntungan pihak lain. Hal ini sangatlah relevan dalam *dogfight*.

Didapat bentuk umum dari fungsi bayarannya adalah,

$$f_i(P_i, P_{-i}) = (-d^2 + 2r_0d) \times (-s_i + s_{-i})/180 \text{ jika } 0 \leq d < \max(r_1, r_2)$$

atau

$$f_i(P_i, P_{-i}) = (k/d) \times (-s_i + s_{-i})/180 \text{ untuk kasus lainnya}$$

Lema 2 Fungsi f tidak satu – satu

Bukti Dengan contoh pilih $s_1 = 1$, $s_2 = 3$, dan parameter lain nilainya ditetapkan, maka nilai fungsi f akan sama dengan ketika kita memilih $s_1 = 9$ dan $s_2 = 11$.

Akibat dari *Lema 2* adalah fungsi bayaran f tidak memiliki batasan yang jelas antara kondisi menang, kalah, atau seri. Sebagai ilustrasi misalkan kita berada dalam skenario $d < \text{firing_range}$ dan $(s_1, s_2) = (60, 61)$ maka dapat dilihat bahwa P_1 memenangkan permainan. Ketika semua parameter lainnya sama, dapat dilihat ketika $(s_1, s_2) = (4, 5)$ maka nilai fungsi f akan sama dengan nilai fungsi f parameter

$(s_1, s_2) = (60, 61)$ namun kondisi permainan ketika $(s_1, s_2) = (4, 5)$ adalah permainan berakhir dengan seri.

Lema 3 Fungsi bayaran f tidak pada untuk ruang bilangan real.

Bukti Jelas bahwa $f \leq g(r_0)$ yang terjadi ketika jarak kedua pesawat berada di *optimum range* sementara posisi relatif kedua pesawat adalah (*hot, cold*) atau (*cold, hot*).

C2. Simulasi dan Analisis

Dogfight yang disimulasikan dalam penelitian ini adalah *dogfight* satu lawan satu yang disimulasikan menggunakan pesawat yang sama. Adapun parameter-parameter yang digunakan yang nilainya akan selalu konstan adalah *speed* = 0.9072, *turn_rate* = 24, *firing_range* = 5, dan *missile_gimbal_limit* = 60. Simulasi dilakukan dengan memvariasikan nilai awal x , y , dan *heading* berdasarkan parameter jarak awal 10, 15, 20, 25, dan 30 serta posisi relatif s_1 dan s_2 yang masing-masingnya dipilih dari $\{0, 5, 10, \dots, 180\}$.

Simulasi diawali dengan simulasi pada permainan berbentuk ekstensif. Simulasi dilakukan menggunakan pohon dengan kedalaman 0 (pemain memilih langkah secara acak), pohon kedalaman 1 (pemain memilih langkah dengan memaksimalkan nilai fungsi bayarannya), dan pohon kedalaman 2 (pemain memilih langkah dengan meminimalkan nilai fungsi bayaran lawan). Tanpa mengurangi keumuman, asumsikan P_1 adalah pemain yang melangkah terlebih dahulu. Berikut hasil simulasi dengan format (a, b, c) dengan a, b, c yang masing-masing menyatakan banyaknya P_1 memenangkan permainan, banyaknya P_2 memenangkan permainan, dan banyaknya permainan yang berakhir denganimbang.

Tabel 4. Hasil simulasi permainan berbentuk ekstensif

		P_2		
		0	1	2
P_1	0	(547, 497, 5801)	(7, 5423, 1415)	(6, 5449, 1390)
	1	(5418, 7, 1420)	(284, 175, 6368)	(335, 157, 6353)
	2	(5457, 6, 1382)	(260, 212, 6373)	(310, 182, 6353)

Beberapa hal penting yang didapatkan dari hasil pada Tabel 4 adalah,

1. Perhatikan ketika salah satu pemain memilih aksi secara acak (pohon kedalaman 0) dan pemain lainnya memilih aksi yang tidak acak (pohon kedalaman 1 atau 2) lalu bandingkan dengan kasus ketika kedua pemain memilih aksi mereka secara acak. Dapat dilihat bahwa terjadi peningkatan kemenangan bagi pemain yang memilih aksi secara tidak acak jika dibandingkan dengan kasus ketika pemain tersebut masih memilih aksi secara acak (sebagai contoh bandingkan $(P_1, P_2) = (1, 0)$ dengan $(P_1, P_2) = (0, 0)$ maka dapat dilihat kemenangan P_1 meningkat dengan sangat signifikan). Hasil ini telah cukup untuk membuktikan bahwa model yang dibuat sudah baik karena mampu meningkatkan banyaknya kemenangan seorang pemain yang menggunakannya (jika dibandingkan dengan memilih aksi secara acak)

2. Perhatikan kasus $(P_1, P_2) = (1, 1)$ dan $(P_1, P_2) = (2, 2)$. Dapat dibentuk sebuah dugaan bahwa pemain yang melangkah pertama akan memperoleh lebih banyak kemenangan ketika kedua pemain menggunakan pohon berkedalaman sama dalam pemilihan aksinya.
3. Perhatikan kasus $(P_1, P_2) = (1, 1)$ dan $(P_1, P_2) = (1, 2)$. Dapat dilihat bahwa banyak kemenangan P_1 meningkat ketika P_2 mengganti cara pemilihan aksinya. Hal sebaliknya juga berlaku pada P_2 . Dari sini dapat diduga bahwa kedalaman pohon yang paling optimal adalah 1 atau metode pemilihan aksi yang paling optimal adalah pemilihan aksi berdasarkan fungsi bayaran di satu langkah ke depan.

Sebagai argumen, metode pemilihan aksi berdasarkan pohon dengan kedalaman 2 yaitu memilih aksi terbaik di dua langkah ke depan. Dengan algoritma *minimax* maka sangat mungkin jika kondisi permainan berakhir di satu langkah ke depan dan algoritma akan melewati langkah tersebut. Akibatnya pemain seolah-olah menghindari kemenangan di depan mata demi menciptakan kondisi terbaik baginya di dua langkah ke depan. Hasil observasi ini mirip dengan akibat dari *Lema 2* yang telah dijabarkan di atas.

4. Perhatikan kasus ketika salah satu pemain memilih aksi secara acak dan pemain lainnya tidak. Dapat dilihat bahwa belum ada *win rate* yang mencapai 100% bahkan ketika pemain lain bermain secara acak. Hal ini terjadi akibat 2 hal, yakni skenario ketika seorang pemain akan menang walau menggunakan langkah acak (akibat kondisi awal) dan adanya langkah yang menghindari kemenangan (akibat *Lema 2*) yang diambil oleh pemain yang memilih aksi secara tidak acak.

Sebagai ilustrasi, misalkan terdapat kondisi P_1 menang di satu langkah ke depan dengan d, s_1, s_2 masing – masing bernilai 4, 60, dan 61. Didapat nilai $f = 1/45$. Selanjutnya perhatikan kondisi imbang dengan d, s_1, s_2 yang masing – masing bernilai 4, 70, 100. Didapat nilai $f = 2/3$. Seorang pemain akan memilih ke skenario kedua karena nilai f yang lebih besar. Akibatnya, pemain tersebut menghindari kemenangan.

D. Simpulan

Dogfight dapat dimodelkan menggunakan teori permainan karena didalamnya mengandung unsur interaksi. Dalam penelitian ini digunakan permainan berbentuk ekstensif untuk mensimulasikan *dogfight* secara dinamis. Adapun fungsi bayaran dibentuk berdasarkan kombinasi dari jarak dan posisi relatif dari para pemain. Berdasarkan simulasi yang dilakukan dapat diketahui bahwa model yang dikembangkan dalam penelitian ini dapat mempengaruhi hasil akhir permainan secara signifikan, yakni dengan meningkatkan jumlah kemenangan pemain jika dibandingkan ketika pemain tersebut memilih aksinya secara acak. Meskipun secara umum model telah baik, namun masih terdapat beberapa kasus yang menyebabkan seorang pemain memilih aksi yang menghindari kemenangan.

E. Referensi

- [1] Saunders, R. dan Souva, M., "Air Superiority and Battlefield Victory", 2020.
- [2] McGrew, J. S., dkk. "Air Combat Strategy using Approximate Dynamic Programming", 2008.
- [3] Gorton, P. R., dkk. "A survey of air combat behavior modeling using machine learning", *IEEE*, 2024.
- [4] Tadelis, S., *Game Theory: An Introduction*, Princeton University Press, 2011.
- [5] Osborne, M. J., *An Introduction to Game Theory*, Oxford University Press, 2000.