Penerapan Kombinatorika dan Teori Graf Berbobot dalam Analisis Komposisi Tim Ideal serta Rute Entry di Map Ascent Valorant

Athallah Nanda Andita - 13525148
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: athallahnanda14@gmail.com , 13525148@std.stei.itb.ac.id

***Abstract*—Permainan berbasis *Tactical First-Person Shooter* (FPS) seperti Valorant menuntut komposisi tim serta strategi yang baik selain akurasi tembakan dan kemampuan penempatan bidikan (*Crosshair Placement*). Keberhasilan dalam memenangkan permainan sangat bergantung pada kemampuan tim untuk menguasai map dengan baik. Makalah ini bertujuan untuk melakukan optimasi terhadap pemilihan komposisi *agent* yang ideal dan rute entry yang paling efisien di Map Ascent.**

Keywords—Valorant; Kombinatorika; Graf; Komposisi; Rute entry

I. PENDAHULUAN

Valorant merupakan permainan berbasis *Tactical First-Person Shooter* (FPS) yang dikembangkan oleh Developer Riot Games yang di rilis pada tahun 2020 untuk Windows dan di tahun 2024 untuk *console*. Valorant merupakan game gratis untuk di mainkan (*Free-to-play*) dengan ada nya *microtransactions* di dalamnya, pengguna bisa membeli berbagai hal kosmetik di dalam game dengan uang asli. Pemain akan bermain dengan format 5 melawan 5 sebagai salah satu agent yang dipilih, dengan satu sisi berperan sebagai penyerang dan yang lain berperan sebagai yang bertahan. Penyerang bertugas untuk mengatur strategi dan memilih *site* yang akan di serang, jika penyerang berhasil memasuki *site* tersebut maka penyerang harus memasang bom (*spike*) di site tersebut. Penyerang juga harus mempertahankan *spike* dari pemain yang bertahan. Untuk pemain yang bertahan bertugas untuk mematikan bom tersebut (*defuse*). Setelah 12 *round*, penyerang akan bertukar giliran dan menjadi pemain yang bertahan begitu juga sebaliknya. Terdapat juga berbagai jenis senjata yang dapat dipilih sesuai gaya bermain pemain dengan tingkat kerusakan yang berbeda-beda. Mode permainan juga bervariasi dengan yang paling popular yaitu *Competitive* dan Unrated, aturan bermain keduanya kurang lebih sama, namun mode *Competitive* menyertakan penambahan *Rank Rating* (RR) jika menang dan dikurangi jika kalah. RR digunakan untuk menaiki tangga Rank yang ada di Valorant. Latar bermain yang disebut *map* oleh Valorant juga bervariasi dan beragam, pada tahun 2025 ini sudah ada 12 *map* dengan 7 di antaranya masuk ke dalam rotasi *map* yang dapat di mainkan di mode *Competitive*.

Gambar 1. Cover Rilis Valorant Console
Sumber:https://www.riotgames.com/ms/news/valorant-console-global-collaboration, diakses pada 18/06/2026

Salah satu tantangan yang ada di Valorant adalah pemilihan *agent* yang tepat dan strategi *entry* yang efektif. *Entry* merupakan istilah yang digunakan untuk memasuki suatu *site* atau area pada *map* tertentu. Setiap *agent* memiliki kemampuan (*abilities*) yang berbeda-beda, dibagi menjadi 2 kemampuan yang dapat di beli pada setiap awal babak baru, 1 kemampuan *signature* yang didapatkan secara gratis setiap awal babak, dan 1 kemampuan *ultimate* yang tidak bisa dibeli namun bisa didapatkan dengan mendapatkan poin *kill* atau melakukan objektif yang lain seperti memasang *spike.* Karena, *abilities* yang beragam ini, tidak semua komposisi *agent* akan efektif untuk mendapatkan kemenangan. Hal ini bergantung kepada sinergi *abilites* antara 5 *agent* di satu tim dan *map* yang akan dimainkan. Selain itu, Rute *entry site* juga bervariasi pada setiap *map* sehingga terdapat kombinasi yang sangat banyak antara *agent* yang dapat digunakan dan strategi pemilihan rute *entry* pada *map* tertentu. Maka dari itu, diperlukan pendekatan sistematis yang efektif untuk mengetahui komposisi *agent* yang harus di pilih dan rute *entry* yang efektif agar dapat memenangkan permaianan.

Dalam konsep Matematika Diskrit, terdapat 2 teori yang akan dapat kaitkan dalam menyelesaikan masalah ini yaitu

Kombinatorika dan Teori Graf. Untuk menyelesaikan masalah komposisi tim yang efektif dan ideal dapat menggunakan Kombinasi dan Rute *entry* dapat digunakan Teori Graf berbobot. Dengan merepresentasikan jalur penyerangan sebagai sisi-sisi dalam graf dan mengaitkan kemampuan setiap agen sebagai faktor pereduksi bobot risiko, kita dapat melakukan optimasi terhadap rute penyerangan.

Makalah ini bertujuan untuk melakukan analisis yang sistematis terhadap kombinasi tim dan rute *entry* yang efektif dengan mengintegrasikan kombinasi dan teori graf berbobot.

II. LANDASAN TEORI

A. Kombinatorika

A.1. Apa itu Kombinatorika ?

Kombinatorika merupakan cabang ilmu di dalam Matematika Diskrit yang mempelajari tentang penyusunan, pengaturan, pemilihan, dan pengelompokan objek dalam suatu himpunan berdasarkan aturan tertentu tanpa harus mengenumerasi semua kemungkinannya. Kombinatorika di dasarkan pada dua kaidah dasar menghitung, yaitu:

1. Rule of product: Jika percobaan 1 memiliki p hasil yang mungkin dan percobaan 2 memiliki q hasil yang mungkin, maka total kemungkinan kedua percobaan tersebut terjadi secara bersamaan adalah p x q.

   $p_1 \times p_2 \times \ldots \times p_n$ hasil

2. Rule of sum: Jika percobaan 1 memiliki p hasil yang mungkin dan percobaan 2 memiliki q hasil yang mungkin, maka total kemungkinan yang terjadi apabila kedua percobaan tersebut saling lepas adalah p + q.

   $p_1 + p_2 + \ldots + p_n$ hasil

Ada 2 cara yang dapat di gunakan jika berhubungan dengan kombinatorika yaitu permutasi dan kombinasi.

A.2. Permutasi

Permutasi merupakan semua susunan atau pengaturan dari himpunan objek dengan memperhatikan urutan kemunculan. Sebagai contoh, {A, B, C} dan {C, B, A} dihitung sebagai 2 kemungkinan permutasi yang berbeda sehingga urutan berpengaruh.

Permutasi r dari n elemen yang tersedia ($r \leq n$) adalah jumlah susunan yang mungkin dari elemen-elemen r. Banyak permutasi dapat dihitung dengan rumus:

$$P(n, r) = n(n - 1)(n - 2)\ldots(n - (r - 1) = \frac{n!}{(n-r)!}$$

P(n, r) = Banyak permutasi yang dapat dibentuk

n = Total elemen yang ada

r = Jumlah elemen yang akan di pilih dan diatur posisinya

A.3. Kombinasi

Kombinasi merupakan bentuk khusus dari permutasi. Di permutasi, urutan kemunculan di perhatikan tetapi pada kombinasi, urutan kemunculan diabaikan. Sebagai contoh, {A, B, C} dianggap sama dengan {C, B, A} dihitung satu kombinasi.

Kombinasi r dari n elemen, C(n, r), adalah jumlah pemilihan yang tidak terurut r elemen yang diambil dari n buah elemen.

$$C(n, r) = \frac{n(n-1)(n-2)\ldots(n-(r-1))}{r!} = \frac{n!}{(n-r)!r!}$$

B. Teori Graf

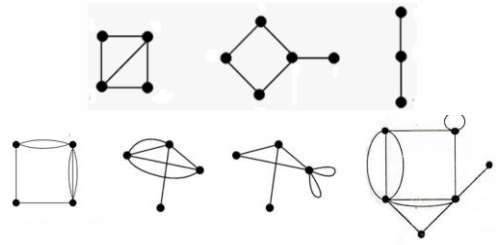
B.1. Apa itu Graf ?

Teori Graf adalah cabang ilmu matematika yang mempelajari struktur hubungan antara objek-objek yang disebut simpul dengan simpul (*vertices*) lainnya yang dihubungkan oleh sisi (*edges*). Graf G didefinisikan sebagai G = (V, E), yang dalam hal ini:

V = himpunan tidak kosong dari simpul $\{V_1, V_2, \ldots, V_n\}$

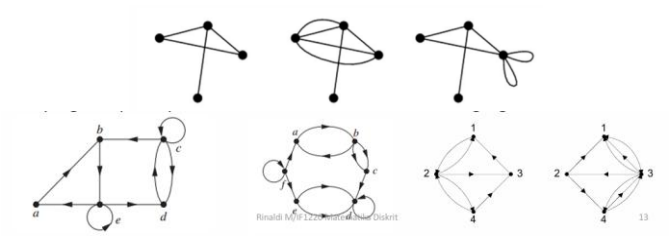
E = himpunan sisi yang menghubungkan satu pasang simpul $\{e_1, e_2, \ldots, e_n\}$

Graf dapat di bagi menjadi beberapa jenis. Yang pertama adalah berdasarkan ada atau tidaknya sisi ganda atau sisi gelang, graf dibagi menjadi 2 yaitu graf sederhana dan graf tak-sederhana. Graf sederhana merupakan graf yang tidak memiliki sisi ganda atau gelang, sementara itu graf tak-sederhana merupakan graf yang memiliki sisi ganda atau gelang.



Gambar 2. Graf Sederhana (atas) dan Graf Tak-Sederhana (bawah)
sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/20-Graf-Bagian1-2026.pdf, diakses pada 19/06/2026

Yang kedua, berdasarkan orientasi arah pada sisi graf dibagi menjadi 2, yaitu graf tak-berarah dan graf berarah. Graf tak-berarah adalah graf yang sisinya tidak mempunyai orientasi arah, sementara itu graf berarah adalah graf yang setiap sisinya diberikan orientasi arah.



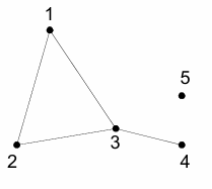
Gambar 3. Graf tak-berarah (atas) dan Graf berarah (bawah)
sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/20-Graf-Bagian1-2026.pdf, diakses pada 19/06/2026

B.2. Terminologi pada Graf

Ada berbagai macam terminologi pada Graf, sementara itu yang akan dipakai dan relevan dalam makalah ini ada 5.

1. Ketetanggaan (*Adjacent*)
2. Bersisian (*Incidency*)
3. Lintasan (*Path*)
4. Upagraf (*Subgraph*)
5. Graf Berbobot (*Weighted Graph*)

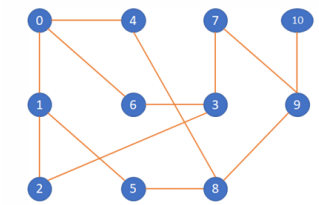
Ketetanggaan didefinisikan dengan dua buah simpul yang terhubung secara langsung oleh 1 sisi. Sementara itu, bersisian didefinisikan dengan sembarang sisi e = (V_j, V_k) diatakan e bersisian dengan simpul V_j atau e bersisian dengan simpul V_k.



Gambar 4. Contoh Graf Ketetanggan dan Bersisian
sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/20-Graf-Bagian1-2026.pdf, diakses pada 19/06/2026

Kita ambil contoh simpul 3, simpul 3 bertetangga dengan simpul 1, 2, dan 4 namun tidak bertetangga dengan simpul 5. Lalu, untuk sisi (2, 3) bersisian dengan simpul 2 dan simpul 3, namun tidak bersisian dengan simpul 1, 4, dan 5.

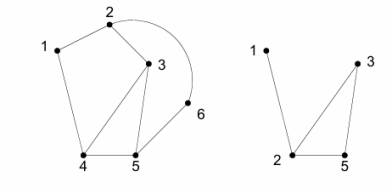
Lintasan didefinisikan dengan panjang n dari simpul awal V_0 ke simpul tujuan V_n dalam graf G sehingga $e_1 = (V_0, V_1)$, $e_2 = (V_1, V_2)$, …, $e_n = (V_{n-1}, V_n)$ adalah sisi dari graf G.



Gambar 5. Contoh lintasan pada Graf
sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/20-Graf-Bagian1-2026.pdf, diakses pada 19/06/2026

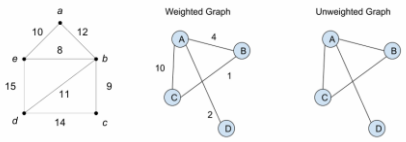
Sebagai contoh, lintasan dari 0 ke 9 adalah (0, 1), (1, 5), (5, 8), (8, 9) memiliki panjang 4. Lintasan di pakai karena pada makalah ini akan diberi satu tujuan dari titik mulai, sehingga akan dicari lintasan untuk mencapai titik tujuan tersebut, dalam konteks makalah ini titik tujuannya adalah site.

Lalu selanjutnya ada upagraf, upagraf merupakan subset atau bagian dari graf utama. Graf utama hanya diambil Sebagian simpul dan sisinya untuk membentuk graf baru, maka itulah upagraf. Upagraf dipakai karena dalam permainan Valorant, kita harus memilih salah satu *site* untuk menentukan rute *entry*, sehingga sisi Dimana *site* lain berada dapat kita buang dan membuat upagraf dari sisi *site* yang di pilih saja.



Gambar 5. Upagraf (kanan) merupakan bagian dari Graf kiri
sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/20-Graf-Bagian1-2026.pdf, diakses pada 19/06/2026

Terakhir, adalah graf berbobot. Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). Graf berbobot sangat cocok dalam menentukan masalah rute paling efektif seperti pada makalah ini.



Gambar 6. Contoh Graf berbobot
sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/20-Graf-Bagian1-2026.pdf, diakses pada 19/06/2026

C. Algoritma Djikstra

Merupakan algoritma yang berfungsi untuk memecahkan masalah lintasan terpendek pada sebuah graf berarah maupun tidak berarah yang memiliki bobot sisi non negatif. Algoritma ini dirancang oleh seorang ilmuwan komputer asal Belanda Bernama Edsger W. Djikstra pada tahun 1956.

Prinsip kerja algoritma ini berdasarkan strategi *greedy*, di mana pada setiap Langkah akan mengambil bobot simpul terkecil yang belum di lewati. Rumus utama yang digunakan oleh algoritma ini adalah rumus relaksasi Djikstra yaitu:

$$d(v) = \min(d(v), d(u) + w(u,v))$$

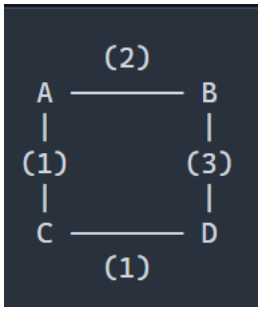
keterangan:

- $d(v)$ = jarak sementara menuju simpul v
- $d(u)$ = jarak terpendek yang telah diketahui menuju simpul u
- $w(u, v)$ = bobot sisi dari u ke v

Sebelum proses pencarian dimulai, dilakukan inisialisasi Djikstra sebagai berikut:

- d(titik awal) = 0
- d(v) = ∞

Proses pencarian jalur dengan bobot paling sedikit dapat dilihat sebagai berikut:



Gambar 7. Contoh pencarian dengan algoritma Djikstra
sumber: https://networkx.org/documentation/stable/reference/algorithms/shortest_paths/dijkstra.html, diakses pada 19/06/2026

Bobot sisi dari A → B = 2, bobot sisi A → C = 1, bobot sisi B → D = 3, bobot sisi C → D = 1. Kita lakukan inisialisasi simpul terhadap jarak dari A, sehingga simpul A = 0, B = ∞, C = ∞, dan D = ∞. Simpul terkecil yang belum dikunjungi adalah A yaitu 0, kemudia lakukan relaksasi ke simpul yang bertetangga dengan simpul A yaitu B dan C. Di dapatkan d(B) = min(∞, 0 + 2) dan d(C) = (∞, 0 + 1). Karena C lebih kecil maka algoritma akan menuju C terlebih dahulu dan mengetes simpul yang bertetangga dengan C yaitu D. Ditemukan di hasil akhir bahwa jalur terpendek untuk mencapai D dari A adalah A → C → D dengan bobot 2.

III. METODE PERHITUNGAN

Pertama, perhitungan akan dimulai dari menghitung kombinasi *agent* untuk membentuk tim yang ideal. Lalu, setelah itu baru di lanjut penghitungan rute *entry* paling efektif.

A. Mengumpulkan Data Agent yang Ada di Valorant

Pertama-tama, untuk menghitung kombinasi tim yang ideal untuk mencapai kemenangan, kita harus mengetahui berbagai macam *agent* yang ada dan dapat di mainkan di Valorant. *Agent* di Valorant di bagi menjadi 4 jenis berdasarkan kemampuan (*role*) yang ada. *Role* yang ada antara lain berupa *duelists*, *controller*, *sentinel*, dan *initiator*. Setiap *role* memiliki fungsi dan tujuan yang berbeda. Pemain yang menggunakan *duelist* akan menjadi orang yang pertama kali melakukan *entry* ke dalam suatu *site* dan biasanya mendapatkan poin *kill* pertama. *Agent* dengan *role* ini biasanya memiliki *abilities* yang sangat berpusat pada dirinya sendiri agar mudah untuk masuk ke dalam *site*, pemain yang memainkan *role* ini juga harus bermain dengan agresif. *Controller*, *role* ini pasti memiliki satu *abilities* untuk mengeluarkan asap (*smoke*) yang berfungsi untuk menghalangi pandangan lawan serta membatasi gerakan lawan agar berpikir dua kali sebelum berjalan melewati *smoke*. *Agent* dengan *role* ini dapat dengan signifikan mengurangi resiko di *kill* oleh pihak lawan. *Sentinel*, merupakan *role* yang pasif saat menyerang namun sangat berguna untuk mempertahankan suatu *site* saat bertahan. *Initiator*, merupakan *role* yang memiliki *abilities* yang berfungsi untuk mengetahui posisi lawan dan membuat kekacauan di *site* untuk membantu *duelist* melakukan *entry site*.

Sebelum melakukan penghitungan, kita harus mengetahui ke 29 *agent* yang saat ini terdapat di Valorant dan *role* nya.

Agent	Role	Origin	Release Patch
Brimstone	Controller	United States	
Viper	Controller	United States	Beta
Omen	Controller	*Unknown*	
Killjoy	Sentinel	Germany	1.05
Cypher	Sentinel	Morocco	
Sova	Initiator	Russia	
Sage	Sentinel	China	Beta
Phoenix	Duelist	England	
Jett	Duelist	South Korea	
Reyna	Duelist	Mexico	1.0
Raze	Duelist	Brazil	Beta
Breach	Initiator	Sweden	
Skye	Initiator	Australia	1.11
Yoru	Duelist	Japan	2.0
Astra	Controller	Ghana	2.04
KAY/O	Initiator	*Alternate Timeline Earth*	3.0
Chamber	Sentinel	France	3.10
Neon	Duelist	Philippines	4.0
Fade	Initiator	Türkiye	4.08
Harbor	Controller	India	5.08
Gekko	Initiator	United States	6.04
Deadlock	Sentinel	Norway	7.0
Iso	Duelist	China	7.09
Clove	Controller	Scotland	8.05
Vyse	Sentinel	*Unknown*	9.04
Tejo	Initiator	Colombia	10.00
Waylay	Duelist	Thailand	10.04
Veto	Sentinel	Senegal	11.07b
Miks	Controller	Croatia	12.05

Gambar 8. Tabel *agent* dan *role* nya di Valorant
sumber:https://wiki.playvalorant.com/en-us/Agents, diakses pada 18/06/2026

Dari tabel diatas diperoleh data jumlah *agent* untuk setiap role sebagai berikut:

1. Controller = 7 (Brimstone, Viper, Omen, Astra, Harbor, Clove, dan Miks)

2. Sentinel = 7 (Killjoy, Cypher, Sage, Chamber, Deadlock, Vyse, dan Veto)
3. Initiator = 7 (Sova, Breach, Skye, KAY/O, Fade, Gekko, dan Tejo)
4. Duelist = 8 (Phoenix, Jett, Reyna, Raze, Yoru, Neon, Iso, dan Waylay)

Kita mengetahui bahwa komposisi tim yang ideal memiliki minimal 1 *agent* dari setiap *role*, karena Valorant adalah permainan 5 melawan 5, maka akan ada 2 *agent* yang memiliki *role* yang sama karena hanya ada total 4 *role* di Valorant.

B. Strategi Perhitungan Kombinasi Tim yang Ideal

Untuk menghitung Kombinasi Tim yang Ideal akan di bagi menjadi beberapa kasus berdasarkan *role* apa yang akan di isi oleh 2 *agent*.

Pertama, jika ingin bermain secara lebih agresif dan memasuki site dengan mudah, maka kita dapat memilih 2 *agent* dengan *role duelist*. Perhitungan untuk jumlah kemungkinannya adalah seperti ini.

1. Hitung Kombinasi 2 *agent* dari 8 *agent duelist* yang tersedia

   $C(8, 2) = \frac{8!}{(8-2)!2!} = \frac{8!}{6!2!} = \frac{8.7.6!}{6!2!} = \frac{56}{2} = 28$ Kombinasi
2. Hitung Kombinasi 1 *agent* dari 7 *agent controller* yang tersedia

   $C(7, 1) = \frac{7!}{(7-1)!1!} = \frac{7!}{6!1!} = \frac{7.6!}{6!1!} = \frac{7}{1} = 7$ Kombinasi
3. Hitung Kombinasi 1 *agent* dari 7 *agent sentinel* yang tersedia

   $C(7, 1) = \frac{7!}{(7-1)!1!} = \frac{7!}{6!1!} = \frac{7.6!}{6!1!} = \frac{7}{1} = 7$ Kombinasi
4. Hitung Kombinasi 1 *agent* dari 7 *agent initiator* yang tersedia

   $C(7, 1) = \frac{7!}{(7-1)!1!} = \frac{7!}{6!1!} = \frac{7.6!}{6!1!} = \frac{7}{1} = 7$ Kombinasi

Untuk total seluruh kemungkinan cara memilih Komposisi tim ideal dengan 2 *duelist* akan digunakan kaidah *rule of product*.

Total Kombinasi tim seluruhnya dengan 2 *duelist* = $C(8, 2) \times C(7, 1) \times C(7, 1) \times C(7, 1) = 28 \times 7 \times 7 \times 7 = 9.604$ Cara

Selanjutnya, jika ingin bermain secara lebih pasif saat menyerang kita dapat menggunakan 2 *sentinel,* jika ingin permainan lebih kacau dan susah untuk diprediksi kita dapat memakai 2 controller atau 2 initiator. Perhitungan ini di dasarkan oleh gaya bermain yang diinginkan. Perhitungan untuk jumlah kemungkinan dari 2 *sentinel*, *initiator*, dan *controller* akan berjumlah sama karena mereka semua sama-sama memiliki 7 *agent* yang dapat dimainkan. Perhitungannya adalah seperti ini.

1. Hitung Kombinasi 2 *agent* dari 7 *agent controller*, *sentinel*, atau *initiator* yang tersedia

   $C(7, 2) = \frac{7!}{(7-2)!2!} = \frac{7!}{5!2!} = \frac{7.6.5!}{5!2!} = \frac{42}{2} = 21$ Kombinasi
2. Hitung Kombinasi 1 *agent* dari 8 *agent duelist* yang tersedia

   $C(8, 1) = \frac{8!}{(8-1)!1!} = \frac{8!}{7!1!} = \frac{8.7!}{7!1!} = \frac{8}{1} = 8$ Kombinasi
3. Hitung Kombinasi 1 *agent* dari 7 *agent agent controller*, *sentinel*, atau *initiator* yang tersedia

   $C(7, 1) = \frac{7!}{(7-1)!1!} = \frac{7!}{6!1!} = \frac{7.6!}{6!1!} = \frac{7}{1} = 7$ Kombinasi
4. Hitung Kombinasi 1 *agent* dari 7 *agent agent controller*, *sentinel*, atau *initiator* yang tersedia

   $C(7, 1) = \frac{7!}{(7-1)!1!} = \frac{7!}{6!1!} = \frac{7.6!}{6!1!} = \frac{7}{1} = 7$ Kombinasi

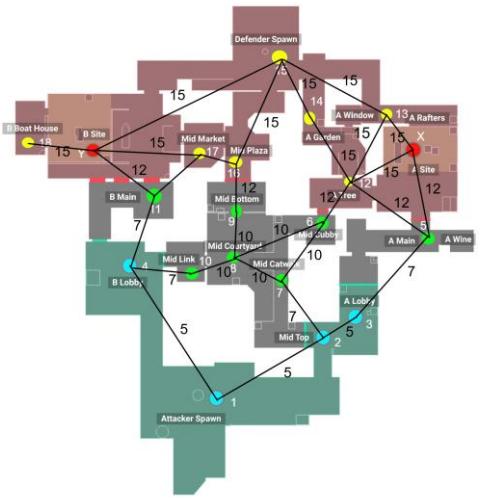
Untuk total seluruh kemungkinan cara memilih Komposisi tim ideal dengan 2 *controller*, *sentinel*, atau *initiator* akan digunakan kaidah *rule of product*.

Total Kombinasi tim seluruhnya dengan 2 *controller*, *sentinel*, atau *initiator* = $3 \times (C(7, 2) \times C(8, 1) \times C(7, 1) \times C(7, 1)) = 3 \times (21 \times 8 \times 7 \times 7) = 3 \times (8.232) = 24.696$ Cara

Sehingga total kombinasi tim ideal yang mungkin di Valorant adalah 9.604 + 24.696 = 34.300 Cara

Perhitungan ini mencakup seluruh kemungkinan dari *agent* yang ada, namun tidak mencakup *agent* yang paling efektif di *map* tertentu.

C. Penggambaran dan Implementasi Graf Berbobot di Map Ascent Valorant



Gambar 9. Map Ascent Valorant yang sudah diimplementasikan graf berbobot
sumber:https://tracker.gg/valorant/db/maps/ascent, diakses pada 19/06/2026

Gambar diatas merupakan salah satu *map* yang ada di Valorant Bernama *ascent*. Gambar dari map tersebut telah digambar sebuah graf berbobot dengan setiap lokasi yang bernama di *map* tersebut dijadikan simpul, kemudian di hubungkan dengan satu sisi ke simpul lain jika lokasi tersebut

bertetangga dan dapat di capai secara langsung. Angka berwarna hitam merupakan bobot sebuah sisi, sementara angka berwarna putih merupakan penamaan simpul. Simpul-simpul yang ada di bedakan menjadi 4 warna, warna biru merupakan simpul yang berada di tempat mulai tim penyerang yaitu area yang berwarna hijau yang meliputi *attacker spawn, mid top, A lobby,* dan *B lobby*. Simpul berwarna biru ini merupakan area paling aman karena memang merupakan area awal yang bisa di capai oleh tim penyerang sebelum setiap babak di mulai. Sisi yang menghubungkan sesama simpul biru diberi bobot 5 karena merupakan jalur aman.

Selanjutnya, ada simpul berwarna hijau, simpul ini merupakan simpul yang terletak di area berwarna abu-abu yang meliputi *B main, mid link, mid courtyard, mid catwalk, mid bottom, mid cubby, dan A main*. Area berwarna abu-abu ini merupakan area netral yang sama-sama tidak dapat dicapai oleh tim penyerang maupun bertahan sebelum babak dimulai. Area ini baru dapat diakses setelah babak permainan dimulai. Area ini merupakan area dengan tingkat bahaya sedang karena tidak dapat diprediksi, sisi yang menghubungkan antara simpul biru dan hijau diberi bobot 7 sementara yang menghubungkan simpul hijau dan hijau diberi bobot 10.

Terakhir, ada simpul berwarna kuning, simpul ini merupakan simpul yang terletak di area berwarna merah yang meliputi *A site, A rafters, A tree, A garden, A window, defender spawn, mid plaza, mid market, B site, dan B boat house.* Area merah ini merupakan area bagi pemain bertahan, *site* merupakan tujuan dari tim penyerang sehingga area berwarna merah ini merupakan area yang sangat berbahaya bagi tim penyerang yang akan melakukan *entry* ke dalam *A site* maupun *B site*. Maka dari itu, sisi yang menghubungkan simpul hijau dengan kuning diberi bobot 12, sedangkan yang menghubungkan simpul kuning dengan kuning diberi bobot 15. Untuk simpul berwarna merah, kurang lebih sama dengan simpul kuning, warna merah hanya untuk menandakan tujuan *entry* dari tim penyerang yaitu *site*.

D. Penghitungan Rute Entry Paling Efektif Untuk Site A

Untuk mengetahui rute *entry* paling efektif, akan dicari rute entry menuju *A site* dengan bobot diurutkan dari paling kecil menurut Algoritma Djikstra. Hasil perhitungannya dapat dilihat di tabel di bawah ini:

No	Rute dari *attacker spawn* → *A site* (Menggunakan nama simpul)	Hasil Perhitungan
1.	1 → 2 → 3 → 5 → X	5 + 5 + 7 + 12 = 29
2.	1 → 2 → 7 → 6 → 12 → X	5 + 7 + 10 + 12 + 15 = 49
3.	1 → 2 → 7 → 6 → 12 → 13 → X	5 + 7 + 10 + 12 + 15 + 15 = 64
4.	1 → 2 → 7 → 8 → 9 → 16 → 15 → 13 → X	5 + 7 + 10 + 10 + 12 + 15 + 15 + 15 = 89
5.	1 → 2 → 7 → 6 → 12 → 14 → 15 → 13 → X	5 + 7 + 10 + 12 + 15 + 15 + 15 + 15 = 94

Tabel 1. Hasil Perhitungan rute ke *A site*

Jika dilihat dari tabel, tampak seperti sudah jelas bahwa rute *entry* paling efektif adalah melalui *A main* karena jaraknya paling pendek dan juga hanya sedikit melewati simpul yang berbahaya. Namun, pada faktanya, perjalanan melalui *A main* termasuk rute yang paling berbahaya karena rute tersebut merupakan rute yang paling umum, sehingga di keluaran *A main,* akan banyak pemain yang bertahan sudah bersiap dengan menaruh *abilities* jebakan dan *smoke*. Pemain *sentinel* yang bertahan biasanya akan menaruh 2 jebakan pada keluaran rute yang paling umum, serta pemain *controller* juga dapat dengan mudah memberikan *smoke* pada keluaran *A main*. Total ada 3 *abilities* yang menghalangi proses *entry* tim penyerang. 1 *abilities* akan memberikan tambahan +15 pada hasil akhir sehingga rute melalui *A main* yaitu 1 → 2 → 3 → 5 → X akan menjadi = 29 + 15 × 3 = 74.

Hal ini membuat rute yang melewati *A tree* 1 → 2 → 7 → 6 → 12 → X dan *A window* 1 → 2 → 7 → 6 → 12 → 13 → X menjadi jauh lebih efektif karena rute tersebut tidak umum dan tidak masuk ke dalam ekspektasi pemain yang sedang bertahan. Hal ini membuktikan bahwa rute uji ke-2 dan ke-3 merupakan rute paling efektif untuk menuju *A site*.

E. Penghitungan Rute Entry Paling Efektif Untuk Site B

Untuk *B site*, langkah yang digunakan kurang lebih sama seperti dengan mencari rute untuk *A site*. Dilakukan pencarian kemungkinan rute *entry* menuju *B site* yang mungkin dan diurutkan dari yang paling kecil menurut Algoritma Djikstra. Hasil perhitungannya dapat dilihat di tabel di bawah ini:

No	Rute dari *attacker spawn* → *B site* (Menggunakan nama simpul)	Hasil Perhitungan
1.	1 → 4 → 11 → Y	5 + 7 + 12 = 24
2.	1 → 4 → 10 → 8 → 9 → 16 → 17 → Y	5 + 7 + 10 + 10 + 12 + 15 + 15 = 74
3.	1 → 4 → 10 → 8 → 9 → 16 → 15 → Y	5 + 7 + 10 + 10 + 12 + 15 + 15 = 74

Tabel 2. Hasil Perhitungan rute ke *B site*

Jika dilihat dari gambar *map Ascent*, terdapat satu simpul kuning di dekat simpul *B site*. Hal ini menandakan bahwa masih ada tempat yang harus di cek oleh tim penyerang yang datang ke *B site* agar benar-benar aman, maka dari itu semua hasil perhitungan harus ditambah 15. Untuk rute nya sendiri, sekilah rute melewati *B main* yang paling efektif saat ingin menuju *B site*, tetap kita harus mengeceknya terlebih dahulu. Kondisi yang di pakai sama seperti di *A main* pada saat membahas *entry A site*, tim yang bertahan akan mencoba menghambat tim penyerang dengan 2 *abilities* jebakan dan 1 *ability smoke*. Setiap *abilities* akan menambahkan 15 ke hasil perhitungan. Jika kita mempertimbangkan semua kondisi tadi, maka hasil perhitungan rute melewati *B main* yaitu 1 → 4 → 11 → Y adalah 24 + 15 × 4 = 84. Sementara itu, untuk hasil perhitungan

ke-2 melewati *mid market* yaitu 1 → 4 → 10 → 8 → 9 → 16 → 17 → Y = 74 + 15 = 89 dan hasil perhitungan ke-3 melewati *defender spawn* yaitu 1 → 4 → 10 → 8 → 9 → 16 → 15 → Y = 74 + 15 = 89. Bobot resiko yang paling kecil masih di pegang oleh rute umum melalui *B main* sehingga dapat di buktikan bahwa rute uji ke-1 melalui *B main* merupakan rute *entry* paling efektif dalam menuju *B site*.

IV. Hasil

Setelah dilakukan perhitungan secara rinci dan sistematis sesuai dengan kombinatorika dan teori Graf, telah di peroleh hasil yang cukup detail. Untuk kombinasi tim yang ideal dengan mendefinisikannya dengan memiliki minimal 1 *agent* dari setiap role dan membaginya menjadi beberapa kasus, diperoleh sebanyak 34.300 kombinasi tim yang bisa dianggap ideal dalam Valorant. 34.300 kombinasi tim ini memperhitungkan seluruh jumlah *agent* yang dapat dimainkan secara umum, belum mempertimbangkan mengenai klasifikasi tingkat kelayakan *agent* dan faktor lain.

Sementara itu, untuk rute *entry* yang efektif di *map Ascent* Valorant ini di peroleh bahwa rute melalui daerah Tengan dan melalu *A tree* adalah rute yang paling efektif jika menuju *A site*. Sementara itu, rute yang paling efektif menuju *B site* adalah melalu *B main* secara langsung karena rute lain untuk menuju *B site* lebih berisiko.

V. Kesimpulan

Dari hasil perhitungan yang di peroleh dapat di simpulkan terdapat banyak sekali kombinasi tim yang ideal di Valorant tergantung dengan gaya bermain setiap tim. Hal ini membuat Valorant menjadi salah satu permainan yang paling susah untuk diprediksi. Selain itu, rute *entry* yang paling dekat dan umum pun tidak selalu menjadi rute yang paling efektif, melainkan harus mempertimbangkan berbagai macam factor dalam permainan itu sendiri. Dalam memainkan Valorant, di perlukan pertimbangan dan kemampuan pengaturan strategi yang sangat baik agar dapat memenangkan permainan dengan konsisten.

Link Video dan Source Code

1. Link Video Presentasi: https://youtu.be/VwxubLIhEKQ
2. Link Source Code: https://drive.google.com/drive/folders/1mMYJzFcVxnzkX71Yb4LXk-uT57lZyf-v?usp=sharing

VI. Ucapan Terima Kasih

Puji dan Syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas izin dan rahmatnya makalah dengan judul "Penerapan Kombinatorika dan Teori Graf Berbobot dalam Analisis Komposisi Tim Ideal serta Rute Entry di Map Ascent Valorant" ini dapat terselesaikan dengan baik. Penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada Bapak Dr. Rinaldi Munir selaku dosen pengampu mata kuliah Matematika Diskrit yang telah memberikan ilmu dan bimbingan selama satu semester ini. Penulis juga menyampaikan terima kasih kepada pihak keluarga dan teman-teman yang selalu memberikan do'a serta masukan dan saran selama penulis mengerjakan makalah ini.

Referensi

[1] R. Munir, "Kombinatorika (Bagian 1)," Bahan Kuliah IF1220 Matematika Diskrit, Program Studi Teknik Informatika STEI-ITB, 2026. Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/18-Kombinatorika-Bagian1-2026.pdf, diakses 19 Juni 2026.

[2] R. Munir, "Kombinatorika (Bagian 2)," Bahan Kuliah IF1220 Matematika Diskrit, Program Studi Teknik Informatika STEI-ITB, 2026. Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/19-Kombinatorika-Bagian2-2026.pdf, diakses 19 Juni 2026.

[3] R. Munir, "Graf (Bagian 1)," Bahan Kuliah IF1220 Matematika Diskrit, Program Studi Teknik Informatika STEI ITB, 2026. Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/20-Graf-Bagian1-2026.pdf, diakses 19 Juni 2026.

[4] R. Munir, "Kombinatorika (Bagian 2)," Bahan Kuliah IF1220 Matematika Diskrit, Program Studi Teknik Informatika STEI ITB, 2026. Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/19-Kombinatorika-Bagian2-2026.pdf, diakses 19 Juni 2026.

[5] Direktorat PuTI, "Teori Graf: Sejarah, Manfaat, dan Aplikasinya," Universitas Telkom Surabaya, 30 Oktober 2023. Sumber: https://surabaya.telkomuniversity.ac.id/teori-graf-sejarah-manfaat-dan-aplikasinya, diakses 19 Juni 2026.

[6] Master of Computer Science BINUS University, "Algoritma Dijkstra," 28 November 2017. Sumber: https://mti.binus.ac.id/2017/11/28/algoritma-dijkstra, diakses 19 Juni 2026.

[7] Project Lighthouse, "Dijkstra's Algorithm," DSA Fundamentals. Sumber: https://projectlighthouse.io/en/books/dsa-fundamentals/pages/dijkstras-algorithm, diakses 19 Juni 2026.

[8] NetworkX Developers, "Dijkstra's Algorithm," NetworkX Documentation. Sumber: https://networkx.org/documentation/stable/reference/algorithms/shortest_paths/dijkstra.html, diakses 19 Juni 2026.

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.

Bekasi, 19 Juni 2026

Athallah Nanda Andita - 13525148