Pencarian Kombo Maksimum Pada Karakter Botan di IdolShowdown Melalui Penerapan Jumlah Jalur Maksimum Dalam N-ary Tree

Muhammad Zahran Ramadhan Ardiana - 13523104[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13523104@std.stei.itb.ac.id

***Abstract*—**

***Idol Showdown* adalah gim pertarungan 2D berbasis karakter VTuber yang memadukan estetika anime dengan mekanisme permainan kompetitif. Salah satu karakter utamanya, Shishiro Botan, dikenal dengan kemampuan serangan jarak menengah dan opsi kombinasi serangan yang variatif. Penelitian ini memanfaatkan struktur pohon N-ary untuk merepresentasikan semua kemungkinan jalur serangan pada karakter Shishiro Botan. Setiap simpul pohon mewakili aksi serangan, sementara algoritma antrean (*queue*) sirkuler digunakan untuk traversal dan validasi jalur secara sistematis. Hasilnya menunjukkan bahwa struktur pohon N-ary memungkinkan identifikasi kombo maksimal sebanyak 71 hit dengan 26 gerakan dan penggunaan 3 bar *Star Meter*. Studi ini menunjukkan efektivitas pohon N-ary dalam mengoptimalkan strategi permainan melalui analisis kombo yang efisien tanpa harus dilakukan secara manual dan menyeluruh.**

***Keywords*— Idol Showdown, *N-ary Tree*, *Combo Optimization*, *Queue Traversal*.**

I. PENDAHULUAN

Gim pertarungan atau *fighting games* merupakan genre *video game* yang menyajikan pertempuran antara dua atau lebih karakter. Pemain mengendalikan suatu karakter yang ia pilih untuk bertarung dengan tujuan utama untuk mengalahkan lawan yang dikendalikan oleh pemain lain atau computer/AI. Setiap pertarungan memiliki jumlah babak dan batas waktu tertentu untuk memenentukan pemenangnya. Karakter pada *fighting games* umumnya menyerang lawan dengan menggunakan suatu seni bela diri atau senjata dan skill unik yang menggambarkan karakter tersebut. Selama awal hingga pertengahan 1990-an, gim pertarungan menjadi genre yang dominan dalam industri permainan video, terutama di arcade. Periode ini melahirkan banyak gim pertarungan populer, termasuk waralaba seperti "Street Fighter", "Mortal Kombat", "Super Smash Bros.", dan "Tekken".

Idol Showdown merupakan salah satu gim pertarungan 2D yang dikembangkan oleh Besto Games dan diluncurkan pada tanggal 5 May 2023. Idol Showdown sendiri merupakan sebuah proyek *fan made* yang menampilkan berbagai *VTuber* dari Hololive Production sebagai karakter yang dapat dimainkan. Salah satu karakter yang bisa dimainkan di gim ini yaitu Shishiro Botan yang memiliki gaya permainan dinamis dan beragam opsi serangan yang dapat dikombinasikan ditambah dengan *ability* Botan yang membuatnya dapat melakukan *zoning* dengan serangan jauh.

Dalam genre *fighting games* seperti idol showdown, kemampuan untuk mengoptimalkan kombo menjadi elemen penting yang dapat menentukan kemenangan. Kombo merupakan rangkaian aksi atau serangan yang dilakukan secara beruntun untuk memberikan kerusakan maksimal pada lawan. Proses eksplorasi dan penguasaan kombo sering kali menjadi daya tarik utama bagi pemain yang ingin meningkatkan performa mereka di tingkat kompetitif. Namun, pencarian kombinasi optimal dalam gim ini bukanlah tugas yang sederhana. Hal ini melibatkan analisis mendalam atas setiap jalur serangan yang memungkinkan, serta pemahaman akan sistem permainan secara keseluruhan.

Pendekatan tradisional untuk menentukan kombo terbaik sering kali dilakukan melalui eksplorasi manual, percobaan berulang, atau memanfaatkan panduan komunitas. Metode ini memakan waktu dan sering kali tidak menghasilkan solusi optimal. Untuk mengatasi keterbatasan ini, penerapan konsep dalam ilmu komputer seperti struktur pohon N-ary dan algoritma pencarian jalur dapat menjadi alat yang sangat berguna. Pohon N-ary memungkinkan representasi yang efisien dari setiap kemungkinan kombinasi gerakan, di mana setiap simpul mewakili aksi spesifik, dan cabang-cabangnya menunjukkan opsi lanjutan. Dengan menggunakan algoritma queue dalam pembuatan tree, kita dapat mengidentifikasi jalur pada tree yang menghasilkan kombo terbanyak secara sistematis.

II. KAJIAN TEORI

A. *Graph*

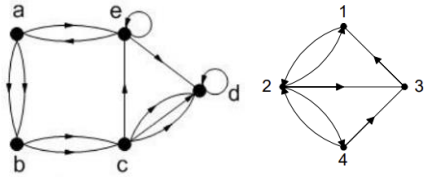
Graf adalah struktur data yang terdiri dari kumpulan simpul (*vertices*) dan sisi (*edges*) yang menghubungkan simpul-simpul tersebut. Graf dapat direpresentasikan dalam berbagai bentuk, seperti graf berarah (directed graph), graf tidak berarah (undirected graph), graf berbobot (weighted graph), dan graf tak berbobot (unweighted graph). Graf umumnya digunakan untuk merepresentasikan hubungan antara objek-objek yang bersifat diskrit [1]. Hubungan yang direpresentasikan dengan graph bermacam-macam seperti jaringan transportasi, relasi antar data, sirkuit elektronik, dan lain-lain.

Berdasarkan orientasi arah pada sisi, graf dibedakan menjadi dua yaitu:

1. Graf tak-berarah (*undirected graph)*
   Graf yang pada tiap sisinya tidak memiliki arah.

Gambar 2.1 Graf tak-berarah
(Sumber: [1])

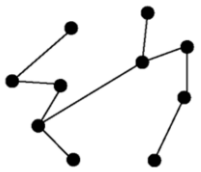
2. Graf berarah (*directed graph)*
   Graf yang memiliki orientasi arah pada tiap sisinya.



Gambar 2.2 Graf berarah
(Sumber: [1])

B. Pohon

Pohon (Tree) adalah struktur data hierarkis yang terdiri dari simpul (node) yang dihubungkan oleh sisi (edge). Pohon dapat didefinisikan sebagai graf tak berarah yang terhubung dan tidak mengandung siklus. Struktur tree sering digunakan untuk merepresentasikan hubungan hierarkis seperti silsilah keluarga, struktur organisasi, sistem file komputer, atau ekspresi aritmatika.



Gambar 2.3 Contoh Struktur Pohon
(Sumber: [1])

Suatu Graph G dengan jumlah simpul *n* dapat dikatogerikan atau ekivalen sebagai pohon dengan teorema jika memenuhi sifat-sifat (property) sebagai berikut:

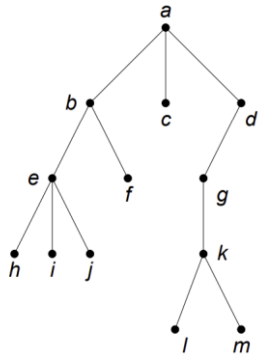
1. Setiap simpul di dalam G terhubung dengan lintasan Tunggal, tidak ada lintasan lain yang menghubungi kedua simpul tersebut (*multi-graph)*
2. Jumlah sisi pada G selalu berjumlah $m = n - 1$ buah sisi.
3. Graph G tidak memiliki sirkuit atau tidak ada lintasan yang melalui sisi-sisi G sehingga berawal dan berakhir pada simpul yang sama.

Berdasarkan karakteristik dan fungsinya, pohon dapat diklasifikasikan menjadi beberapa jenis:

1. Pohon Biner (*Binary Tree*): Pohon di mana setiap simpul memiliki paling banyak dua anak.
2. Pohon Biner Pencarian (*Binary Search Tree*): Pohon biner di mana simpul kiri memiliki nilai lebih kecil dan simpul kanan memiliki nilai lebih besar dari induknya.
3. *AVL Tree*: Pohon biner yang seimbang dengan perbedaan tinggi maksimum satu antara subpohon kiri dan kanan.
4. *Heap Tree*: Pohon biner yang memenuhi sifat *heap* (baik *max heap* atau *min heap*).
5. *N-ary Tree*: Pohon di mana setiap simpulnya memiliki paling banyak N anak.

C. Pohon Berakar

Pohon berakar (Rooted Tree) adalah pohon yang memiliki simpul khusus yang disebut sebagai akar (root). Dari akar, semua simpul lainnya dapat diakses melalui jalur yang unik. Dalam pohon berakar, hubungan hierarkis menjadi jelas antara induk dan anak. Struktur pohon ini memiliki properti rekursif, di mana setiap anak dari sebuah simpul juga merupakan pohon.



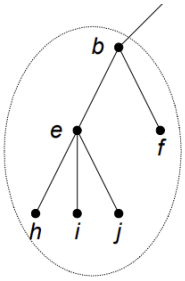
Gambar 2.4 Struktur Pohon Berakar
(Sumber: [1])

Terminologi dasar yang digunakan dalam struktur data pohon berakar adalah sebagai berikut:

1. Simpul Induk (*parent*)
   Simpul yang merupakan pendahulu langsung dari sebuah simpul lain disebut simpul induk dari simpul tersebut. Simpul *b* merupakan simpul induk dari *e* dan *f*.
2. Simpul Anak
   Simpul yang langsung berada di bawah simpul induk dan terhubung langsung dengannya. Sebuah simpul anak hanya memiliki satu induk tetapi bisa memiliki beberapa anak di bawahnya. Jika *b* adalah induk dari *e* dan *f*, maka *e* dan *f* adalah anak dari *b*.
3. Simpul Akar
   Simpul paling atas dari pohon atau simpul yang tidak memiliki induk disebut simpul akar. Simpul *a* pada gambar 2.4 adalah simpul akar. Pohon yang tidak kosong harus berisi tepat satu simpul akar dan tepat satu jalur dari akar ke semua simpul lain di pohon.
4. Derajat (*degree*)
   Derajat suatu simpul menyatakan jumlah suatu upapohon atau anak dari simpul tersebut. Contohnya simpul *a* pada gambar 2.4 berderajat tiga karena memiliki tiga simpul anak {*b, c, d*}.
5. Simpul Daun (*leaf*)
   Simpul yang berderajat 0 atau tidak memiliki anak. Simpul ini berada di posisi paling bawah dalam struktur pohon dan biasanya mewakili titik akhir dari suatu jalur di pohon.
6. Nenek moyang suatu simpul (*ascentor*)
   Nenek moyang (*ancestor*) suatu simpul adalah semua simpul yang berada di jalur dari akar menuju suatu simpul tersebut, termasuk simpul induk, kakek, buyut, dan seterusnya hingga ke akar. Sebagai contoh pada gambar 2.4, {*a,d,g,k*} merupakan simpul nenek moyang dari simpul *m* dan *l*.
7. Keturunan (*descendant*)
   Keturunan suatu simpul mencakup semua simpul yang berada di bawah simpul tersebut dalam struktur pohon, termasuk anak, cucu, dan seterusnya. Keturunan dapat

dianalogikan dengan suatu simpul *x* merupakan keturunan dari simpul *y* yang lain jika dan hanya jika *y* merupakan leluhur dari *x*

8. Saudara (*sibling*)
   Simpul-simpul yang memiliki induk yang sama disebut saudara kandung. Sebagai contoh {*h*,*i*,*j*}pada gambar 2.4 disebut saudara dari induk *e*.
9. Simpul Internal
   Simpul yang memiliki setidaknya satu anak. Berbeda dengan simpul daun, simpul internal bertindak sebagai penghubung dalam jalur pohon.
10. Upapohon (*Subtree*)
    Bagian dari pohon asal yang terdiri dari sebuah simpul dan seluruh keturunannya. Setiap simpul dalam pohon dapat dianggap sebagai akar dari sebuah *subtree*. Contoh upapohon dari gambar 2.4 dapat dilihat pada gambar di bawah ini

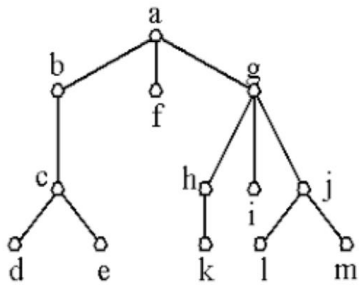


Gambar 2.5 Upapohon (*subtree*) dari gambar 2.4
(Sumber: [1])

11. Kedalaman (*Depth*)
    Kedalaman suatu simpul merujuk pada jarak dari simpul akar ke simpul tersebut, dihitung dengan jumlah sisi (edge) yang dilalui. Sebagai contoh simpul akar memiliki kedalaman 0, simpul anak langsung memiliki kedalaman 1, dan simpul *m* pada gambar 2.4 memiliki kedalaman 4.

D. *Pohon N-ary*

N-ary Tree adalah jenis pohon berakar di mana setiap simpul dapat memiliki paling banyak N anak. Pohon ini merupakan generalisasi dari pohon biner dan sering digunakan ketika jumlah anak yang mungkin untuk setiap simpul lebih dari dua. Struktur ini banyak digunakan dalam parsing data seperti XML atau HTML, serta pada permainan berbasis pohon seperti catur dan tic-tac-toe

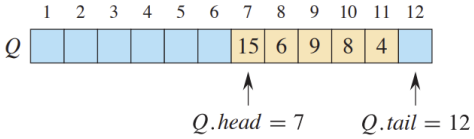


Gambar 2.6 Pohon 3-*ary*
(Sumber: [1])

E. *Queue*

Queue (antrean) adalah struktur data linear yang mengikuti prinsip FIFO (*First-In, First-Out*). Prinsip ini berarti elemen yang pertama kali dimasukkan ke dalam antrean akan menjadi elemen pertama yang dikeluarkan.

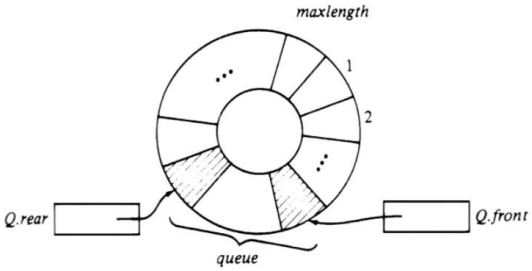
Queue memiliki kepala dan ekor. Ketika suatu elemen diantrekan, elemen tersebut mengambil tempatnya di ekor antrean, seperti halnya pelanggan yang baru datang mengambil tempat di akhir antrean. Elemen yang diantrekan selalu merupakan elemen yang berada di kepala antrean, seperti pelanggan di kepala antrean, yang telah menunggu paling lama [2]. *Queue* banyak digunakan dalam berbagai aplikasi seperti manajemen tugas dalam sistem operasi, traversal graf dan pohon, serta dalam algoritma jaringan.



Gambar 2.7 Illustrasi *queue* dalam bentuk *array*
(Sumber: [2])

Pada gambar 2.7, atribut *head*(kepala) dan *tail*(ekor) dari antrean *Q* merupakan indeks yang menunjuk ke elemen kepala dan ekor dari antrean tersebut. Atribut tersebut digunakan untuk memudahkan operasi utama dalam *queue* yaitu *enqueue* dan *dequeue*. Dimana *enqueue* menambahkan elemen ke antrean dan dequeue menghapus elemen terdepan dari antrean.

Dalam meningkatkan efisiensi *array* dalam *queue*, dapat digunakan *queue* yang bersifat sirkuler. Antrean melingkar (*circular queue*) adalah jenis antrean di mana elemen terakhir terhubung kembali ke elemen pertama, membentuk struktur seperti cincin. Ini mengatasi keterbatasan antrean linear, di mana ruang yang kosong di awal array tidak dapat digunakan kembali. Indeks pada *circular queue* dapat dihitung dengan menggunakan operasi modulus Rear = (Rear + 1) % Size. Atribut *rear* merupakan sebutan lain untuk *tail* pada *queue*.



Gambar 2.8 Illustrasi *circular queue*
(Sumber: [3])

F. *Idol Showdown*

Idol Showdown adalah gim pertarungan 2D berbasis karakter yang terinspirasi dari VTuber populer di bawah naungan agensi Hololive Production. Dikembangkan oleh komunitas penggemar (*fan-made game*) dengan perhatian khusus pada elemen serangan yang *flashy* dan representasi karakter yang sesuai dengan kepribadian virtual idol yang mereka perankan. Gim ini menonjol dengan estetika anime yang memiliki grafik dengan tema piksel, aksi, dan mekanisme yang mudah dipahami namun sulit dikuasai sepenuhnya.

Gambar 2.9 Gim Idol Showdown Botan VS Suisei
(Sumber: Arsip Penulis)

1. Mekanisme

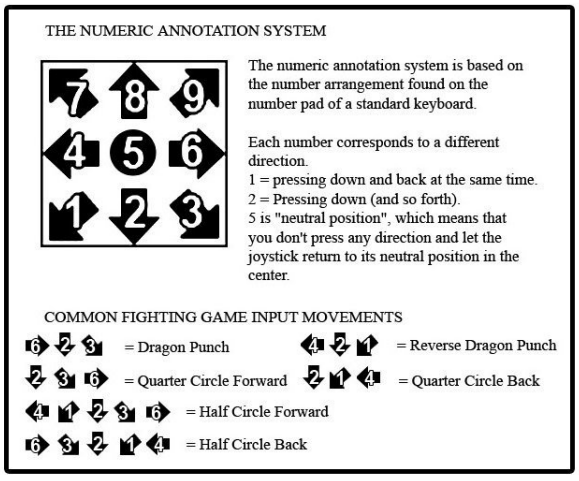
Pemain dapat memilih karakter VTuber favorit dan bertarung satu lawan satu dalam arena 2D. Setiap karakter memiliki serangan dasar, serangan spesial, dan movement yang mencerminkan kepribadian serta ciri khas VTuber tersebut. Dalam pertarungan, tiap karakter akan memiliki karakter pembantu (*collab*), *star meter*, dan burst mode. Selain itu karakter juga memiliki *attack cooldown* berupa attack *recovery/backswing* dan efek *hitstun* yang berbeda pada tiap karakter. Seitap pertarungan akan kondisi tertentu dan memiliki batas waktu tertentu dan ketika waktu habis akan ditentukan berdasarkan tujuan dari *match* tersebut (sisa darah terbanyak atau survive ketika melawan AI).

2. Shihsiro Botan

Shihsiro Botan merupakan salah satu karakter *fighting* utama dalam idol showdown dengan tingkat kesulitan 4-*star*. Botan adalah salah satu pertarung jarak menengah dengan kerusakan tinggi. Persenjataan balistik yang dimilikinya membuat area serangan Botan semakin luas. Namun, kecepatan serangan proyektilnya yang kurang cepat membuat *zoning* penuhnya kurang kuat, terutama melawan karakter yang memiliki alat serupa tetapi lebih cepat.

3. Movement

Movement (gerakan) menjadi hal yang wajib diketahui dalam setiap *video games*. Ketika baru pertama kali memainkan sebuah gim terutama ber-genre *fighting*, pemain harus dapat membaca gerakan yang biasa tertera pada *guide* yang umumnya dituliskan dalam bentuk *numerical directional input* atau *numeric annotation system*.

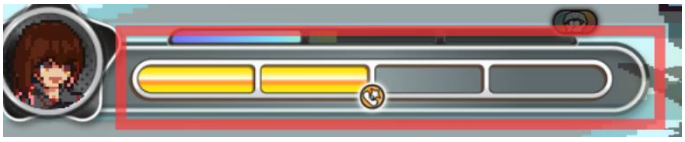


Gambar 2.10 Numeric Annitation System
(Sumber: https://critpoints.net/2018/03/04/how-to-perform-fighting-game-motions/)

Notasi tersebut membantu menghindari banyak masalah yang dialami pemula saat mempelajari cara melakukan gerakan karena nantinya gerakan akan mempengaruhi ke tipe serangan yang dilakukan dan lainnya seperti *special move* dan *guarding*.

4. *Star Meter* dan *Superchat Meter*

Star Meter dalam idol showdown sama seperti meteran dalam gim pertarungan lainnya yang berfungsi untuk melakukan suatu gerakan tertentu. *Star Meter* dapat diisi dengan melakukan berbagai serangan walau serangan terkena *block* oleh lawan. Ketika menerima damage dari lawan, *Star Meter* juga akan meningkat walau lebih sedikit dibanding saat kita yang menyerang.*Star Meter* yang diperolah dapat ditampung hingga empat bar.



Gambar 2.11 *Star Meter System*
(Sumber: Arsip Penulis)

Superchat Meter merupakan meter yang terpisah dan digunakan untuk mengganti gerakan secara instan (*Superchat Cancel*) dan *Off Collab*. *Superchat Meter* akan terisi seiring pertempuran di mulai dan dapat dipercepat dengan melakukan gerakan-gerakan tertentu seperti *idol move*. Tidak seperti *Star Meter* yang akan tetap bertahan di ronde selanjutnya, *Superchat Meter* akan reset ketika satu ronde selesai.

Gambar 2.12 *Superchat Meter System*
(Sumber: Arsip Penulis)

5. *Offense*

Offense atau serangan dalam Idol Showdown terdiri dari berbagai tipe sebagai berikut:

a) *Normal Attacks*
Terdiri dari serangan berjenis ringan (*L*), medium (*M*), dan berat (*H*). *Normal Attacks* dapat dihubungkan ke serangan yang lebih kuat atau serangan yang berbeda dengan kekuatan yang sama untuk membentuk kombo.

b) *Throws* dan *Grab*
Throws adalah serangan jarak pendek dan cepat yang tidak dapat diblokir. *Throws* tidak dapat digunakan di tengah kombo. *Grab* merupakan hal yang sama dengan *Throws*, tetapi dalam karakter tertentu ia memiliki tipe *air attack* yang dapat digunakan dalam kombo.

c) *Universal Overhead*
Gerakan serangan dari atas yang lambat dan anti *Throws* dan *Grab* yang berguna untuk menghancurkan block lawan yang rendah.

d) *Special Moves*
Gerkan spesial/khusus (*S*) dapat dilakukan dengan cara menekan serangan spesial itu sendiri atau dengan suatu *motion input* pada serangan normal *L/M/H*.

e) *Star Special*
Versi serangan yang lebih kuat dibanding *Special Moves* dan membutuhkan satu bar *Star Meter* untuk menggunakannya.

f) *Idol Moves*
Gerakan spesial yang memberikan *Superchat Meter* yang

tinggi dengan input 22S. Dalam beberapa karakter, *Idol Moves* membutuhkan satu bar *Star Meter*.

g) *Super Star Attacks*
Serangan semacam *ultimate moves* dalam gim pertarungan. *Super Star Attacks* merupakan serangan terkuat dan membutuhkan dua bar *Star Meter*. Meskipun begitu, serangan tersebut masih memungkikan untuk diblok lawan.

h) *Superchat Cancels*
Serangan spesial yang dilakukan ketika dalam *state* serangan spesial lain. *Superchat Cancels* membutuhkan satu bar *Superchat Meter* untuk beralih ke serangan spesial tersebut tanpa harus menunggu serangan spesial sebelumnya selesai.

6. *Collab*

Pemain tidak hanya memilih karakter utama saja untuk bertempur, tetapi juga dapat memilih karakter pembantu yang disebut *Collab* untuk melancarkan serangan dan mengganggu lawan atau memberikan *buff* ke karakter utama pemain. *Collab* dapat dipanggil dengan dua cara, yaitu *off Collab* yang membutuhkan dua sampai tiga *Superchat Meter* dan *Call-In* yang membutuhkan dua *Star Meter*.

Gambar 2.13 *Call-In Collab* Ayunda Risu
(Sumber: Arsip Penulis)

7. *Hitbox*

Hitbox merupakan area yang umumnya tak terlihat dalam gim yang digunakan untuk mendeteksi *collision* atau interaksi antara objek. Dalam gim pertarungan seperti Idol Showdown, *hitbox* memainkan peran yang sangat penting dalam memastikan serangan dan pertahanan berjalan secara adil dan konsisten. *Hitbox* dapat dibagi menjadi beberapa jenis, contohnya *hurtbox* di mana suatu objek menerima suatu serangan dan *attackbox* yang menentukan area serangan yang dapat melakukan kerusakan [5].

Gambar 2.14 *hurtbox player* (biru) dan *attackbox* (merah)
(Sumber: Arsip Penulis)

8. *Frame data*

Frame data adalah informasi yang menggambarkan durasi waktu setiap aksi karakter dalam satuan frame—satuan waktu terkecil dalam animasi gim yang biasanya berjalan pada kecepatan 60 *frame* per detik (FPS). Setiap gerakan dalam gim pertarungan memiliki tiga fase penting dalam frame data: *startup frames*, *active frames*, dan *recovery frames* [6].

Frame data sangat penting untuk memahami kapan serangan aman untuk digunakan (*safe on block*), kapan pemain rentan terhadap serangan balik (*punishable*), dan bagaimana kombo dapat dihubungkan dengan presisi.

Gambar 2.15 *Frame data* pada serangan karakter Botan
(Sumber: Arsip Penulis)

III. METODOLOGI

A. Batasan

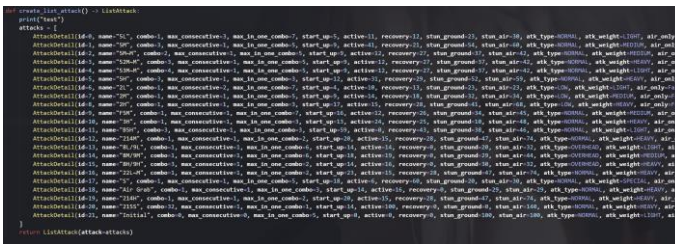
Dalam mengimplementasikan program untuk melakukan uji coba, terdapat beberapa batasan yang digunakan untuk menyederhanakan masalah pencarian kombo maksimum. Batasan tersebut diantaranya adalah:

1. Hitbox dalam pencarian disederhanakan menjadi satu bagian area saja tidak dipecah pecah.
2. Serangan yang di data adalah serangan yang memungkinkan untuk melakukan kombo. Gerakan-gerakan seperti menghancurkan defense atau gerakan retreat tidak di masukkan.
3. Musuh tidak melakukan blocking di awal, sehingga serangan awal diasumsikan selalu mengenai lawan.
4. Perhitungan kecepatan jatuh atau gravitasi dilakukan secara regresi karena tidak ada sumber yang memberikan rumus secara jelas mengenai system gamenya.
5. *Collab* tidak disertakan setelah beberapa kali percobaan karena kurang efektif dalam menghasilkan kombo yang optimal untuk karakter Botan. Selain itu, banyaknya jenis collab yang tersedia menyebabkan operasi program menjadi terlalu berat dalam menghasilkan kombinasi kombo.

B. Pengumpulan Data

Pengumpulan data dilakukan dengan mengambil referensi dari komunitas. Namun, data tersebut perlu melalui proses validasi untuk memastikan keakuratannya. Validasi diperlukan karena serangan (attack) dalam data tersebut tidak selalu memiliki status aktif secara penuh, sehingga hasil perhitungan yang dihasilkan mungkin tidak sepenuhnya akurat. Proses validasi ini melibatkan modifikasi terhadap serangan, yang dilakukan untuk mengatasi penyederhanaan pada *hitbox* serta efek kekuatan serangan (*attack force/effect*). Penyederhanaan tersebut dapat sangat memengaruhi posisi musuh (*enemy*) di langkah-langkah selanjutnya dalam simulasi, implementasi atau analisis.

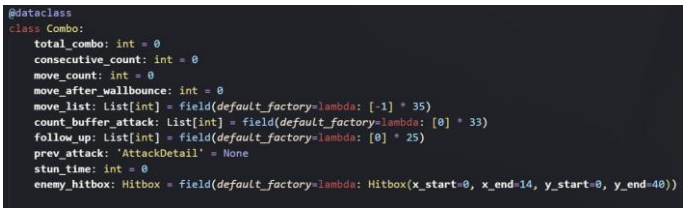
Berikut Pengumpulan data berupa skill set yang telah dilakukan:



Gambar 3.1 Data *skillset* dari karakter Botan
(Sumber: Arsip Penulis)

C. *Implementasi*

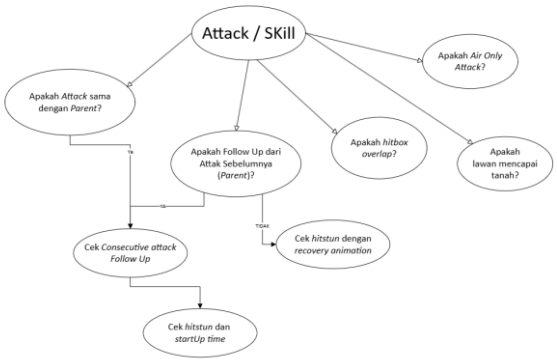
Program pencarian kombo maksimum dibuat menggunakan Python. Pemilihan Python didasarkan pada keterbatasan memori yang ditemukan saat mencoba implementasi serupa menggunakan C. Dalam program ini, pembuatan struktur pohon (*tree*) untuk pencarian jalur maksimum dilakukan dengan menggunakan algoritma antrean (*queue*) sirkuler. Elemen yang dimasukkan ke dalam antrean merupakan objek dari kelas Combo, yang memiliki struktur seperti yang ditunjukkan pada gambar di bawah ini.



```
@dataclass
class Combo:
    total_combo: int = 0
    consecutive_count: int = 0
    move_count: int = 0
    move_after_wallbounce: int = 0
    move_list: List[int] = field(default_factory=lambda: [-1] * 35)
    count_buffer_attack: List[int] = field(default_factory=lambda: [0] * 33)
    follow_up: List[int] = field(default_factory=lambda: [0] * 25)
    prev_attack: 'AttackDetail' = None
    stun_time: int = 0
    enemy_hitbox: Hitbox = field(default_factory=lambda: Hitbox(x_start=0, x_end=14, y_start=0, y_end=40))
```

Gambar 3.2 *Class* untuk simpul pembuatan tree
(Sumber: Arsip Penulis)

Proses pembuatan pohon kombo dimulai dengan melakukan traversal terhadap seluruh data skillset. Setiap skillset dimasukkan ke dalam antrean untuk pengecekan validitasnya. Jika serangan atau skill tersebut dianggap valid dalam kondisi tertentu, serangan tersebut akan ditambahkan ke dalam kombo dan dimasukkan kembali ke dalam antrean. Selanjutnya, serangan tersebut akan diolah untuk memeriksa kemungkinan serangan lanjutan.



Gambar 3.3 Illustrasi sebagian contoh dalam validasi serangan
(Sumber: Arsip Penulis)

Validasi dilakukan secara mendetail untuk memastikan efisiensi dan optimalitas. Sebagai contoh:

- Suatu gerakan seperti 9L hanya dianggap valid pada kondisi musuh berada di udara. Namun, pada penggunaan kedua, gerakan tersebut tidak dianggap valid jika tidak memberikan keuntungan strategis untuk mempertahankan kombo yang lebih panjang.
- Serangan bertipe *wallbounce* seperti 214M, 22L~M, dan 214H juga dibatasi. Serangan tersebut sangat berpengaruh untuk mempertahankan kombo yang lama sehingga validasi gerakan ini mengharuskan setidaknya enam serangan lain dilakukan di antara ketiga serangan tersebut, walaupun dalam gim-nya tidak seperti itu. Hal ini bertujuan untuk menjaga penggunaan memori dan operasi tetap efisien.

Selain itu, pencegahan kombo tak terbatas sudah diantisipasi oleh pengembang game dengan menetapkan batasan penggunaan maksimum (max use) untuk setiap serangan, seperti yang terlihat pada gambar berikut.

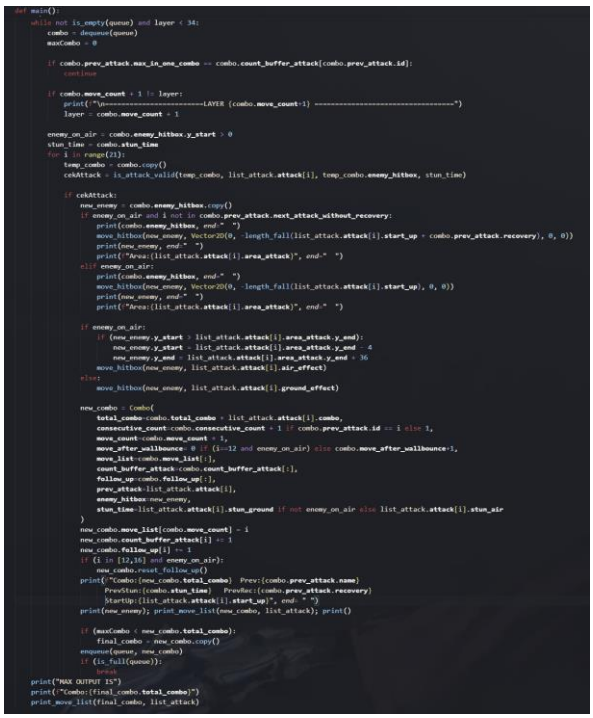
Move Type	Max Uses
L Normal	7
M Normal	5
H Normal	3
Air Normal	6
Special	3
Air Special	4
Star Special	3

Gambar 3.4 Infinite Prevention System
(Sumber: https://wiki.gbl.gg/w/Idol_Showdown/System)

Queue diinisialisasi dengan simpul awal berupa kombo bertipe "initial". Kapasitas elemen antrean yang digunakan adalah 30.000 elemen. Hal ini karena banyaknya kemungkinan serangan lanjutan yang harus dimasukkan ke dalam antrean. Sebelumnya, kapasitas 10.000 elemen telah dicoba, tetapi menyebabkan "*queue is full*".

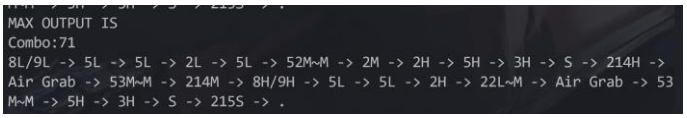
Simpul awal (*initial combo*) yang berfungsi sebagai akar pohon akan di-dequeue untuk diolah. Pohon yang dibentuk dalam program ini menggunakan array sebagai penyimpanan serangan-serangan sebelumnya (parent) pada field move_list. Setiap parent dapat memiliki maksimal 21 child node, yang merepresentasikan semua kemungkinan serangan valid berdasarkan kondisi parent.

Kondisi pada setiap child node akan diperbarui dengan mengubah *field* seperti *enemy_hitbox* dan *stun_time*. *Field* tersebut digunakan untuk menentukan *child node* berikutnya yang memenuhi kondisi. Struktur utama program untuk pembuatan pohon kombo dapat dilihat pada cuplikan kode berikut.



Gambar 3.5 *Main code* pembuatan pohon kombo
(Sumber: Arsip Penulis)

Hasil yang di dapat setelah program berjalan kurang lebih lima menit menunjukkan bahwa maksimum kombo yang dapat dilakukan oleh Botan adalah sebanyak 71 kombo dengan total gerakan sebanyak 26 dan penggunaan *Star Meter* sebanyak 3 *Bar*.



Gambar 3.6 Hasil program pencarian kombo maksimum
(Sumber: Arsip Penulis)

Gerakan yang dihasilkan oleh program tersebut sebagian besar benar. Namun, perhitungan yang dilakukan oleh program hingga ke tingkat frame atau sekitar 1/60 detik menyebabkan gerakan tersebut sangatlah sulit dilakukan sehingga ketika input yang dilakukan terlambat tiga sampai lima frame saja dapat menyebabkan kombo hancur dan tidak dapat dilanjutkan. Untuk menangani itu, *player* dapat memodikasi beberapa gerakan agar sedikit mempermudah untuk melakukan kombo tersebut.

Penulis mencoba melakukan untuk mengganti gerakan ke-18 (5L) pada program dengan gerakan 5M agar musuh tidak mudah jatoh dan mencoba menambahkan serangan 5L setelah gerakan ke-22 agar *hurtbox* lawan lebih mudah di-*hit* akibat sedikit efek *stun* yang diberikan. Video pembuktian gerakan dapat dilihat pada video berikut: https://drive.google.com/file/d/1WnCQVleQzYtybKYKp2F8Kbkj2ZrB3AHu/view?usp=sharing.

Gambar 3.7 Contoh gerakan *miss* akibat terlambat 3 – 5 *frame*
(Sumber: Arsip Penulis)

IV. Kesimpulan

Kombo didapat paling banyak sebesar 71 kombo di mana kombo ini melebihi pada guide yang tersedia baik di media manapun. Meskipun gerakan dalam kombo ini masih dapat dimodifikasi dan menghasilkan kombo yang lebih seperti dalam video, penggunaan pohon *N-ary* sangat membantu dalam mencari berbagai kemungkinan jalur serangan. Tanpa struktur ini, pemain atau analis harus mencari secara manual seluruh kombinasi yang mungkin, sedangkan dengan program berbasis *N-ary tree*, hanya perlu melakukan modifikasi kecil pada data atau aturan untuk menguji kombinasi baru secara efisien.

Namun demikian, optimalisasi program masih perlu ditingkatkan mengingat adanya batasan-batasan dalam implementasi saat ini, seperti penyederhanaan *hitbox* dan penghilangan elemen tertentu yang dapat memengaruhi hasil analisis.

V. Ucapan Terima Kasih

Penulis mengucapkan syukur kepada Tuhan Yang Maha Esa atas rahmat dan kelancaran yang diberikan sehingga penulis dapat menyelesaikan makalah ini dengan baik. Penulis juga menyampaikan terima kasih kepada Pak Rila Mandala selaku dosen pengampu mata kuliah Matematika Diskrit 2024 Kelas 02, atas ilmu dan bimbingan yang telah diberikan selama perkuliahan berlangsung. Ilmu yang beliau sampaikan sangat membantu penulis dalam memahami materi kuliah serta menyelesaikan tugas makalah ini.

Tak lupa, penulis juga mengucapkan terima kasih kepada kedua orang tua yang telah memberikan dukungan selama proses perkuliahan. Penulis berharap makalah ini dapat memberikan kontribusi positif dalam memahami lebih dalam materi yang telah dipelajari.

References

[1] R. Munir, "Aljabar Geometri – Tahun Ajaran 2024/2025," Institut Teknologi Bandung (ITB). [Online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2024-2025/algeo24-25.htm. [Diakses: Jan. 6, 2025].

[2] s\T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th ed. Cambridge, MA: MIT Press, 2022. [Online]. Available: https://dl.ebooksworld.ir/books/Introduction.to.Algorithms.4th.Leiserson.Stein.Rivest.Cormen.MIT.Press.9780262046305.EBooksWorld.ir.pdf. [Accessed: Jan. 9, 2025].

[3] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*. [Online]. Available:

https://users.dcc.uchile.cl/~voyanede/cc4102/dS&A%20Book%20By%20Alfred%20-Aho.pdf. [Accessed: Jan. 9, 2025].
[4] C. Wagar, "How to perform fighting game motions," *Critical Points*, Mar. 4, 2018. [Online]. Available: https://critpoints.net/2018/03/04/how-to-perform-fighting-game-motions/. [Accessed: Jan. 9, 2025].
[5] G2A.COM Editorial Team, "Hitboxes in games: A simple guide," *G2A News*, Feb. 28, 2024. [Online]. Available: https://www.g2a.com/news/features/hitboxes-in-games-a-simple-guide/. [Accessed: Jan. 9, 2025].
[6] E. P. Wibowo, "Mengenal frame data: Game fighting itu penuh perhitungan," *Metaco.gg*, Nov. 24, 2020. [Online]. Available: https://metaco.gg/fighting-games/mengenal-frame-data-game-fighting-itu-penuh-perhitungan. [Accessed: Jan. 9, 2025].

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, 10 Januari 2024

Muhammad Zahran Ramadhan Ardiana
13523104