

Aplikasi Pohon Keputusan pada Tetris agar Menghasilkan *Perfect Clear* Berdasarkan *Piece* yang Tersedia pada Awal Game.

Muhammad Garebaldhie ER Rahman - 13520029¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13520029@std.stei.itb.ac.id

Abstract—Tetris merupakan permainan teka teki dimana pemain diminta untuk menyusun berbagai *piece* atau komponen dan ketika *piece* tersebut disusun dan dapat diselesaikan agar mendapatkan *score*. Sistem *scoring* pada tetris ini memiliki banyak jenisnya, berbagai macam *combo* dapat menghasilkan *score* yang berbeda beda. *Combo* yang memiliki *score* tertinggi ialah *Perfect Clear*. *Perfect Clear* merupakan kondisi dimana tumpukan tetris tersebut dapat terselesaikan tanpa menyisakan blok apapun. Namun kondisi ini sangatlah sulit untuk terjadi, pemain harus bisa menaruh blok pada posisi dan rotasi yang tepat. Permasalahan diatas ternyata dapat diselesaikan dengan menggunakan *Decision Tree* atau pohon keputusan yang akan dibahas pada makalah ini.

Keywords—Tetris, Perfect Clear, Combos, Pohon, Pohon Keputusan.

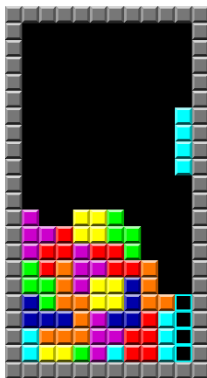
masing. Berbagai macam genre dapat dinikmati para pemain mulai dari *action*, *puzzle*, *mystery*, *FPS*, *Simulation* dan lainnya. Namun ada satu game yang sejak dahulu sampai sekarang masih tetap eksis dan dimainkan oleh banyak orang tanpa kehilangan popularitasnya sama sekali. *Game* tersebut merupakan *genre puzzle* yang sangat legendaris. Ya benar, game tersebut adalah Tetris, siapa yang tidak mengenal tetris? Tetris merupakan permainan yang mengasah logika, *problem solving*, serta *decision making* dalam memainkannya [2].

Pemain diminta untuk mendapatkan *score* setinggi mungkin dengan cara menyocokkan *piece* demi *piece* dari *piece* yang tersedia serta kesulitan akan meningkat seiring permainan dimainkan. Namun, Tetris sendiri sudah mengalami banyak perubahan. Pada awalnya game Tetris ini bisa dibilang sebagai tetris classic yang mana game ini hanya bergantung kepada *score* namun sekarang terdapat Tetris Battle yaitu pemain dapat bermain dengan pemain lainnya dan bertarung satu sama lain untuk saling memenangkan game nya [1].

Meskipun Tetris terlihat simple, namun nyatanya tetris memiliki *combo* serta mekanisme tertentu yang agak sulit untuk dipahami, banyak juga istilah istilah yang mungkin jarang didengar oleh orang yang jarang bermain tetris.

Opener merupakan salah satu istilah yang ada pada game Tetris. *Opener* berfungsi untuk membuka game tetris dengan setup atau setting tertentu sehingga akan memaksimalkan poin pemain. Ada banyak jenis *opener* yang tersedia pada game Tetris ini salah satunya ialah *Perfect Clear*. *Perfect Clear* ialah kondisi dimana *board* tetris benar-benar kosong tanpa menyisakan salah satu bagianpun. Ketika pemain berhasil membentuk *Perfect Clear* pemain akan mendapatkan *score* yang sangat besar dibandingkan *combo* ataupun mekanisme lainnya. Namun nyatanya *Perfect Clear* sangat sulit untuk dilakukan. Terdapat pola yang harus pemain pahami serta sistem rotasi pada tetris. Pemilihan keputusan itulah yang akan dibahas pada makalah ini sehingga pemain bisa melakukan *Perfect Clear* tanpa harus memikirkan segala macam kombinasi yang ada.

I. PENDAHULUAN



Gambar 1.1 Permainan Tetris (Sumber:

https://upload.wikimedia.org/wikipedia/commons/thumb/9/9c/Typical_Tetris_Game.svg/800px-Typical_Tetris_Game.svg.png)

Video game awalnya adalah sebuah media untuk hiburan semata, banyak orang bermain game entah karena mereka suka ataupun kompetisi.

Seiring berkembangnya zaman, banyak game yang bermunculan dan memiliki keunikan tersendiri untuk mengait hati para pemain seperti misalnya *Pac-man*, *Tamagochi*, *Sudoku*, *Snake*, serta masih banyak game yang lainnya [4]. Setiap game memiliki genre ataupun keunikannya masing

II. TEORI DASAR

A. Graf

Graf merupakan sebuah terminology atau istilah yang

memiliki *vertices and edge* [9].

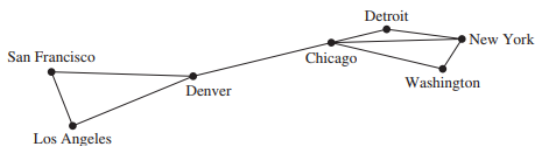
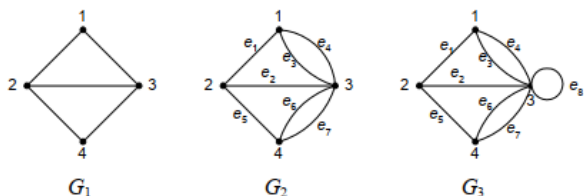


FIGURE 1 A computer network.

Gambar 2.1 Anologi Graf dengan *Computer Network* (Sumber : Rosen, Kenneth H. 2019. *Discrete Mathematics and Its Applications, Eighth Edition*)

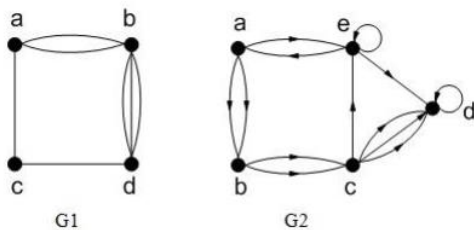
Secara formal, sebuah Graf G didefinisikan sebagai $G = (V, E)$ dimana V merupakan himpunan tidak kosong dari simpul-simpul dengan definisi $V = \{v_1, v_2, \dots\}$ serta E yang merupakan himpunan sisi (*Edges*) yang menghubungkan sepasan *vertices* (simpul) [6].

Graf memiliki banyak jenis dan bentuk. Graf dikatakan graf sederhana apabila setiap simpul terhubung tidak lebih dari satu sisi. Jika ada yang lebih dari satu sisi maka Graf tersebut dinamakan graf ganda. Jika suatu graf memiliki simpul yang Kembali ke simpul itu sendiri disebut dengan graf semu. Pada terminologi graf, istilah dari simpul Kembali ke simpul dinamakan dengan kalang atau loop [6].



Gambar 2.2 Ilustrasi graf sederhana, ganda serta semu secara berurutan (Sumber: Munir, Rinaldi. 2021. *Graf (Bag. 1): Bahan Kuliah IF2120 Matematika Diskrit*)

Graf juga dapat dibedakan berdasarkan orientasi arah yang dimiliki. Terdapat 2 jenis graf yang dibedakan dengan orientasi arah. Graf tak-berarah serta Graf berarah. Kedua graf tersebut memiliki perbedaan yaitu pada graf berarah *edges* yang terletak pada graf memiliki arah atau orientasi terhadap tujuannya.



G1 : graf tak-berarah; G2 : Graf berarah

Gambar 2.3 Ilustrasi graf tak-berarah dan graf berarah (Sumber: Munir, Rinaldi. 2021. *Graf (Bag. 1): Bahan Kuliah IF2120 Matematika Diskrit*)

B. Pohon

Pohon merupakan bagian dari graf yaitu graf tak berarah.

Pohon disebut pohon karena graf yang dibentuk terlihat seperti pohon. Misalkan pohon keluarga. Pohon keluarga merupakan bagian graf yang memiliki *vertices* berupa nama orang dan *edges* yang merupakan hubungan antara orang pada pohon keluarga tersebut [9].

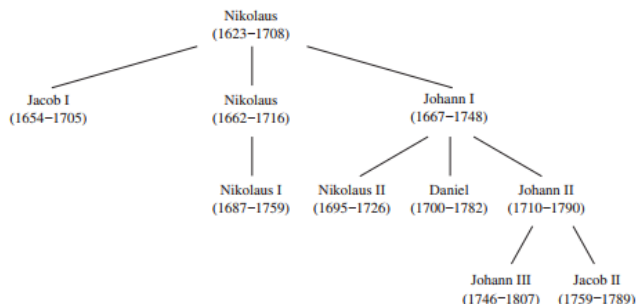


FIGURE 1 The Bernoulli family of mathematicians.

Gambar 2.4 Ilustrasi Pohon dengan silsilah keluarga (Sumber : Rosen, Kenneth H. 2019. *Discrete Mathematics and Its Applications, Eighth Edition*)

Meskipun pohon merupakan bagian dari graf, tetapi tidak semua graf adalah pohon. Agar sebuah graf G bisa disebut sebagai pohon, graf tersebut harus memenuhi sifat sifat pohon diantaranya [7]:

1. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
2. G terhubung dan memiliki $n - 1$ buah sisi
3. G tidak mengandung sirkuit
4. G tidak mengandung sirkuit dan penambahan satu sisi pada graf membuat hanya satu sirkuit
5. G terhubung dan semua sisinya adalah jembatan.

C. Jenis Pohon

Pohon dibagi menjadi beberapa jenis, pohon berakar, pohon terurut, pohon n -ary, serta pohon biner [8].

1. Pohon Berakar

Pohon Berakar (*Rooted Tree*) adalah jenis pohon yang salah satu simpulnya diperlakukan sebagai akar dan sisi lainnya diberi arah sehingga membentuk graf berarah. Agar dapat membentuk pohon berarah, ambil salah satu simpul lalu buat dia sebagai akarnya sehingga akan terbentuk sebuah pohon dengan akar baru yaitu simpul yang baru saja di pilih [9].

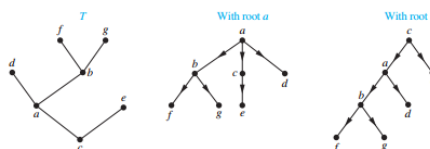


FIGURE 4 A tree and rooted trees formed by designating two different roots.

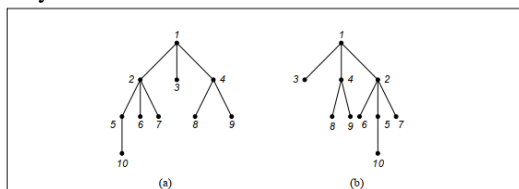
Gambar 2.5 Ilustrasi pohon berakar (Sumber : Rosen, Kenneth H. 2019. *Discrete Mathematics and Its Applications, Eighth Edition*)

Pohon berakar memiliki beberapa terminology yang harus dipahami. Diantaranya [8]:

- a) Anak (*Child* atau *children*) dan Orangtua (*parent*). Jika melihat gambar 2.5 bag 2, simpul a merupakan orang tua dan simpul b, c, d merupakan anak. Anak merupakan turunan dari simpul orang tua.
- b) Lintasan (*Path*) merupakan cara untuk mencapai suatu titik tertentu dari suatu titik tertentu. Misalkan lintasan dari a ke f ialah a, b, f.
- c) Saudara Kandung (*sibling*) merupakan simpul yang memiliki orang tua yang sama.
- d) Upapohon (*subtree*) merupakan pohon yang lebih kecil dari pohon awalnya, biasanya bisa diambil pada bagian kiri ataupun kanan dari pohon tersebut.
- e) Derajat (*degree*) sebuah simpul ialah jumlah upapohon (atau jumlah anak) pada simpul tersebut
- f) Daun (*leaf*) merupakan simpul berderajat nol, biasanya daun terletak pada bagian bawah pohon.
- g) Simpul Dalam (*internal nodes*) merupakan simpul yang memiliki anak.
- h) Aras (*level*) atau tingkat merupakan ketinggian dari sebuah pohon. Jika pohon tidak memiliki anak maka arasnya ialah nol
- i) Tinggi (*Height*) atau Kedalaman (*depth*) merupakan aras maksimum pada suatu pohon.

2. Pohon Terurut

Pohon terurut merupakan pohon yang urutan anak anaknya memiliki sebuah arti.

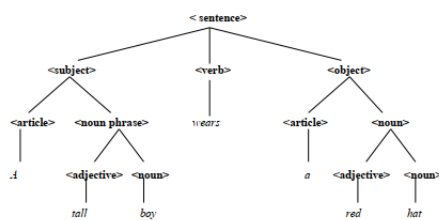


(a) dan (b) adalah dua pohon terurut yang berbeda

Gambar 2.6 *Ilustrasi pohon terurut* (Sumber: Munir, Rinaldi. 2021. *Pohon* (Bag. 2): *Bahan Kuliah IF2120 Matematika Diskrit*)

3. Pohon n-ary

Pohon n-ary merupakan pohon berakar yang setiap simpulnya memiliki n buah anak. Pohon n-ary dikatakan teratur atau penuh apabila setiap cabangnya memiliki tepat n anak.



Gambar Pohon parsing dari kalimat *A tall boy wears a red hat*

Gambar 2.7 *Ilustrasi pohon n-ary sebagai parsing kalimat* (Sumber: Munir, Rinaldi. 2021. *Pohon* (Bag.2): *Bahan Kuliah IF2120 Matematika Diskrit*)

4. Pohon biner

Pohon biner merupakan pohon n-ary dengan $n = 2$, pohon biner ini memiliki peranan yang sangat penting karena banyak aplikasinya. Biasanya pohon biner ini dibedakan antara *left child* dan *right child* nya karena anaknya hanya 2. Contoh aplikasi pohon biner ialah pohon awalan, pohon keputusan, kode awalan, serta kode Huffman.

5. Pohon Keputusan

Pohon keputusan merupakan turunan dari pohon biner karena pohon keputusan mengaplikasikan sifat dari pohon biner. Secara tidak sadar, pohon keputusan sering kali kita gunakan. Misalnya saja Ketika kita ingin menentukan makanan mana yang akan kita *order* secara tidak langsung kita menerapkan pohon keputusan di otak kita. Pohon keputusan juga biasanya sering digunakan untuk memodelkan sebuah masalah yang kemudian akan dicari solusinya menggunakan *machine learning*.

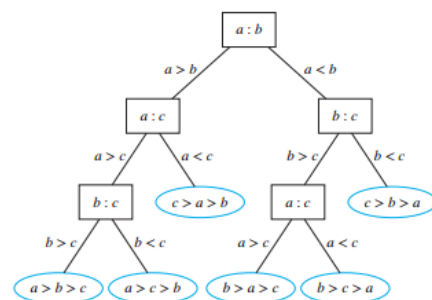


FIGURE 4 A decision tree for sorting three distinct elements.

Gambar 2.8 *Ilustrasi pohon keputusan dalam menentukan urutan bilangan* (Sumber: Munir, Rinaldi. 2021. *Pohon* (Bag.2): *Bahan Kuliah IF2120 Matematika Diskrit*)

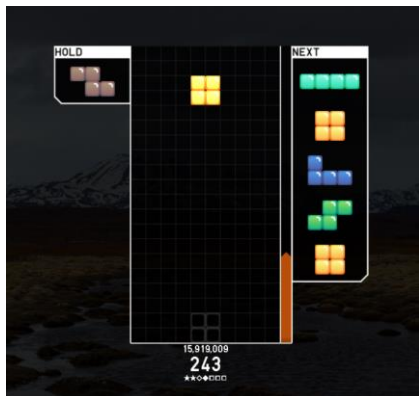
D. Tetris

Tetris merupakan *puzzle game* yang berkembang pada tahun 1984 yang diciptakan oleh Alexey Pajitnov. Pada awalnya Tetris digunakan untuk mengukur kemampuan komputer dalam menjalani sebuah tugas dan memberikan hiburan saat melakukan pengujian [1].

Tetris mengedepankan *problem solving* serta *decision making* dalam permainannya. Banyak hal dan mekanisme yang unik pada *game* ini.

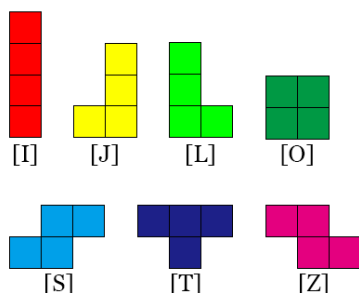
Awal mulanya Tetris hanya ada satu jenis yaitu *classic tetris*. *Classic Tetris* merupakan game yang memiliki tujuan untuk mendapatkan *score* setinggi mungkin dengan cara menyelesaikan puzzle berupa potongan potongan bentuk yang disimpan di dalam *board*. Seiring perkembangan jaman, *game* Tetris memiliki perkembangan yang begitu pesat. Banyak *tournament* di penjuru dunia yang diselenggarakan untuk para pemain tetris. Gameplay Tetris pun sudah mulai berkembang yang awalnya berfokus kepada mencari *score* atau poin tertinggi, terdapat *Battle Tetris* yang memiliki Gameplay unik yaitu pemain bisa melawan pemain lainnya atau masuk kedalam arena dimana arena tersebut berisi 99 pemain lainnya, mode permainan ini sering kali disebut sebagai *Tetris 99* [2].

a. *Gameplay*



Gambar 2.9 *Game Tetris* (Sumber: arsip penulis)

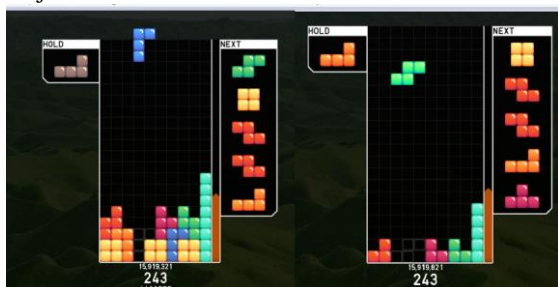
Ketika memulai permainan, layout dari game ini memiliki tiga komponen. *Next*, *Hold*, serta *Board*. *Board* merupakan papan atau alas yang dapat kita gunakan untuk menyimpan *piece*. *Hold* merupakan Mekanisme *game* sehingga pemain bisa melakukan pertukaran antara *piece* yang sedang dipegang dan *piece* yang ada pada board. *Next* merupakan bagian paling penting, pada Tetris ketika mengawali permainan, dapat dipastikan bahwa pemain mendapatkan masing masing *piece* yang unik, system ini seringkali disebut sebagai 7 Bag Randomizer. Namun, ketika mengawali game pemain sudah pasti mendapatkan masing masing jenis dari *piece* yang tersedia dan sisanya akan di acak Kembali menggunakan *system random* tersebut [3].



Gambar 2.10 *Jenis piece pada Tetris* (Sumber:

https://gigazine.net/gsc_news/en/20191116-tetris-algorithm/)

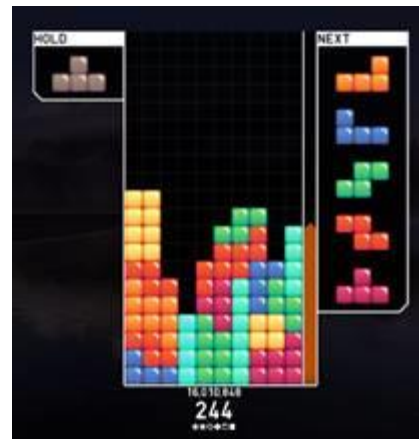
Ketika pemain menyelesaikan *puzzle*, maka board yang dipenuhi dengan *piece* akan hilang dan berubah menjadi score.



Gambar 2.11 *Kondisi board sebelum dan sesudah puzzle diselesaikan* (Sumber: arsip penulis)

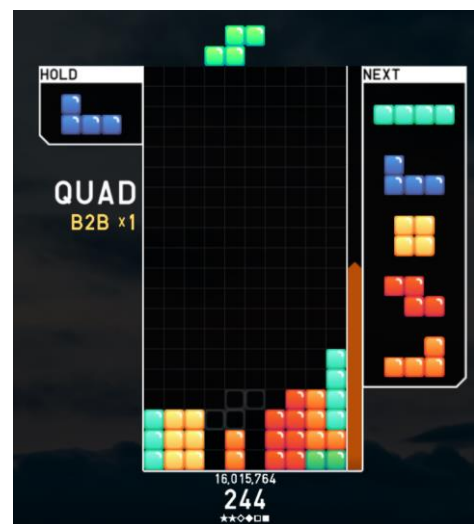
b. *Combo*

Combo merupakan mekanisme dimana player menyelesaikan *puzzle* secara berturut turut tanpa ada jeda sekalipun. *Combo* pada tetris ada dua jenis, *Combo* biasa dan *B2B*. *Combo* pada tetris digolongkan menjadi beberapa jenis. *Single*, *double*, *triple*, *quad/tetris* serta *T-Spin*.



Gambar 2.12 *Pemain berhasil mendapatkan tetris/quad* (Sumber: arsip penulis)

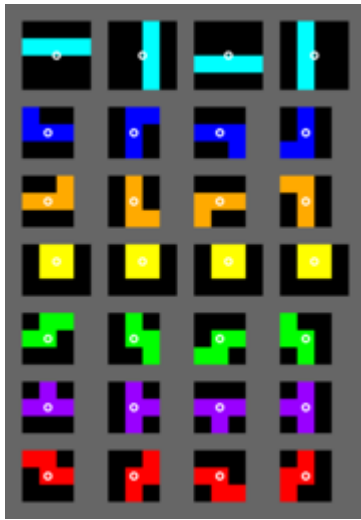
B2B merupakan singkatan dari *Back to Back*. *B2B* ialah combo Ketika pemain menyelesaikan tetris/quad atau *T-Spin* lalu dilanjutkan dengan tetris/quad atau *T-Spin* lainnya tanpa ada jeda.



Gambar 2.13 *Pemain berhasil combo B2B* (Sumber: arsip penulis)

c. *SRS (Super Rotation System)*

Pada tetris terdapat sistem rotasi *piece* yang memungkinkan pemain untuk merubah orientasi *piece* sesuai yang diinginkan. Rotasi hanya berlaku secara *CW (Clockwise)* dan *CCW (Counter Clockwise)*. Setiap *piece* memiliki *anchor* atau pusat massa yang menjadi pusat rotasi dari setiap *piece* [3].



Gambar 2.14 SRS (Sumber:

<https://static.wikia.nocookie.net/tetrisconcept/images/3/3d/SRS-pieces.png/revision/latest/scale-to-width-down/180?cb=20060626173148>)

d. *Pattern Matching*

Pattern matching bukanlah suatu mekanisme melainkan suatu *sense* dari pemain itu sendiri. *Pattern Matching* ini berguna untuk mempercepat proses berpikir. Ketika ingin menyimpan piece pada tempat tertentu. *Pattern Matching* tidak bisa didapat secara instan, diperlukan latihan khusus agar terbiasa melihat pola yang ada.



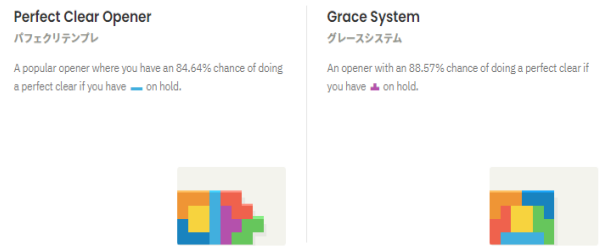
Gambar 2.15 PCO, Grace System, serta ILSZ merupakan *pattern* yang digunakan untuk melakukan *perfect clear* (Sumber: <https://four.lol/>)

e. *Opener*

Opener merupakan cara pemain memulai tetris dengan menggunakan *pattern* yang ada. Dengan menggunakan *opener*, kemungkinan pemain mendapatkan score ataupun kemenangan menjadi jauh lebih tinggi serta kemungkinan pemain mendapatkan *perfect clear* meningkat menjadi sebesar 85%.

III. APLIKASI POHON KEPUTUSAN

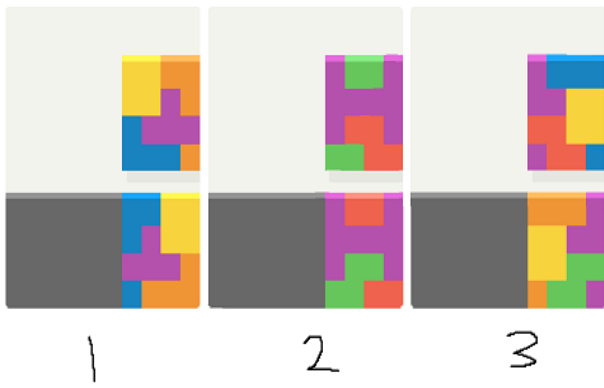
Agar pemain dapat menyelesaikan *perfect clear* maka Langkah pertama yang harus dilakukan adalah menentukan *pattern* mana yang ingin dipakai oleh pemain agar dapat membuat *perfect clear*. Penulis menyarankan menggunakan *pattern* PCO dan *Grace System* karena memiliki *rate* yang lebih tinggi untuk terjadi *perfect clear* dibandingkan dengan *pattern* yang lain.



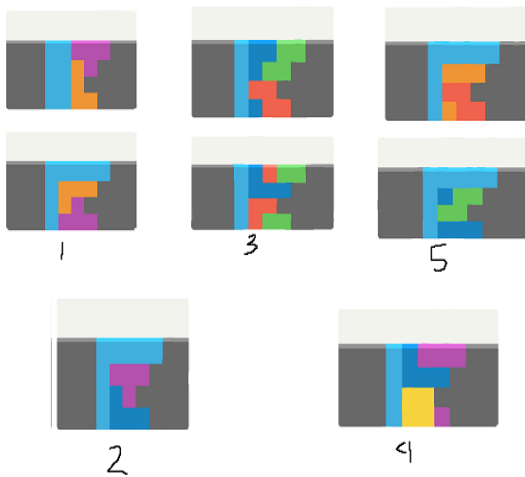
Gambar 3.1 PCO dan Grace System (Sumber: <https://four.lol/>)

Jika pemain ingin membuat pola *perfect clear* maka player harus memperhatikan isi dari *bag* atau *next* nya karena hal itu akan sangat memengaruhi proses pencarian solusi pada pola *perfect clear*. Jika seandainya pemain memiliki *T-Piece On Hold* maka pemain bisa menggunakan *Grace System* sebagai pola dasar dari solusi *Perfect Clear*. Namun, jika pemain memiliki *I-Piece On Hold* maka pemain bisa menggunakan *PCO* sebagai dasar dari solusi nya.

Grace System merupakan sebuah dasar solusi dari *perfect clear*. Pemain diwajibkan untuk membuat pola dasarnya terlebih dahulu agar bisa menemukan solusinya. Ketika sudah membuat pola dasarnya, pemain harus melihat isi *bag* ataupun *next piece* yang akan muncul karena akan memengaruhi proses pencarian solusi *Perfect Clear*. Pemain dapat menentukan apakah akan menaruh *T-Piece* secara vertical ataupun tidak, jika pemain menaruh *T-Piece* secara vertical maka terdapat lima kemungkinan solusi yang bisa terjadi. Setiap solusi memiliki pola dan rotasi yang unik sehingga dibutuhkan *piece* yang berbeda-beda setiap solusinya. Pada solusi pertama dibutuhkan dua *I-Piece* serta 1 *L-Piece* agar memenuhi pola pertama. Jika ternyata pemain bukan memiliki *L-Piece* melainkan *J-Piece* maka pemain bisa memakai solusi ke dua agar bisa memenuhi *perfect clear*. Pola ke 3 merupakan pola yang agak unik karena urutan penempatan serta sistem rotasi akan memengaruhi penyelesaian *perfect clear*. Pola ke 4 membutuhkan *L, O, serta I-piece* agar terselesaikan. Jika pemain memilih *grace system* sebagai dasarnya maka hanya pola 1, 2, dan 4 yang bisa dilakukan. Jika pemain tidak memilih untuk menaruh *T-Piece* secara vertical maka akan masuk ke solusi *grace system* itu sendiri. Pada solusi ini terdapat 3 kemungkinan solusi. Solusi pertama dibutuhkan *L, O, serta J-Piece* agar dapat diselesaikan. Pola kedua memerlukan *T, S, serta Z-Piece* agar dapat terselesaikan. Perhatikan bahwa pola ke dua ini memiliki dua urutan tergantung penempatan pertama nya. Pola ke 3 pada *Grace System* merupakan mirror satu sama lain. *S, O, L Piece* serta *J, O, Z Piece* merupakan solusi dari pola ini. Perhatikan bahwa *S dan Z* serta *J dan L* adalah *mirror* satu sama lain.



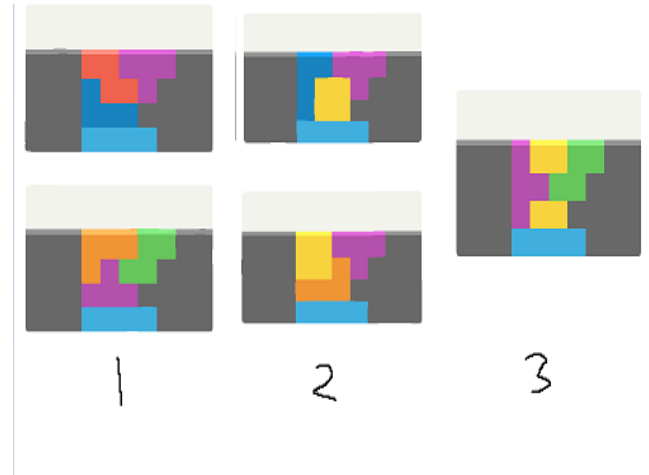
Gambar 3.2 Pola Grace System (Sumber: arsip penulis)



Gambar 3.3 Pola Vertical (Sumber: arsip penulis)

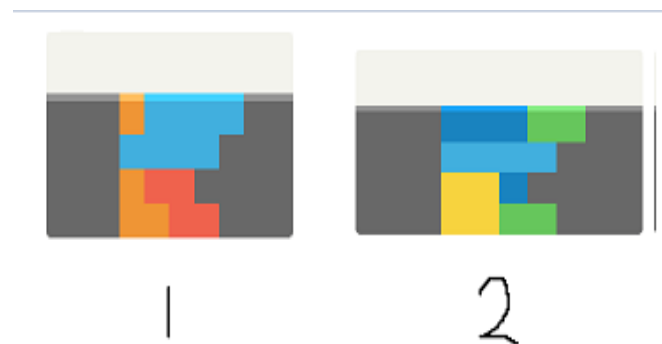
PCO atau *Perfect Clear Opener* merupakan sebuah dasar solusi dari *perfect clear*. Pemain diwajibkan untuk membuat pola dasarnya terlebih dahulu agar bisa menemukan solusinya. Ketika sudah membuat pola dasarnya, pemain harus melihat isi *bag* ataupun *next* piece yang akan muncul karena akan memengaruhi proses pencarian solusi *Perfect Clear*. Jika pada *Grace system* penentuan solusi ditentukan dari cara menaruh *T-Piece*, pada PCO penentuan solusi diambil berdasarkan keputusan pemain dalam menaruh *I-piece*. Agar dapat menyelesaikan pola PCO, pemain bisa menaruh *I-Piece* secara vertical ataupun secara horizontal. Jika pemain memilih untuk menaruh *I-Piece* secara horizontal maka solusinya akan sama seperti solusi pola vertical pada *Grace System* ditambah 2 solusi lainnya. Solusi lainnya yaitu Pola ke-3 dan ke-5. Pola ke tiga membutuhkan *J, S, dan Z Piece*. Solusi ini agak unik karena pemain bisa mencari solusi lainnya dengan menukar penempatan serta orientasi dari *S* dan *Z Piece*. Pola ke 5 merupakan *mirror* satu sama lain. Jika bagian pertama membutuhkan *I, L, dan Z Piece* maka solusi lainnya membutuhkan *I, J, dan S Piece* agar bisa menyelesaikan *perfect clear*nya. Namun, jika pemain memilih untuk menaruh *I-Piece* secara horizontal terdapat kasus lainnya yaitu jika pemain menaruh pada baris 1 ataupun baris 3 (dihitung dari bawah). Jika pemain menaruh *I-Piece* pada baris pertama maka akan ada 3

kemungkinan solusi. Sama seperti sebelumnya solusi pertama ini adalah pencerminan satu sama lain. Dibutuhkan *T, S, L* ataupun *T, J, Z piece* agar dapat terselesaikan. Pada solusi dua ajika pemain tidak memiliki *S* ataupun *Z piece* tetapi memiliki *O* piece maka solusi ini dapat dicapai. Solusi ketiga dari *1st Row* merupakan kombinasi dari solusi pertama dan kedua. Dibutuhkan *S, O, serta T-Piece* agar dapat menyelesaikannya.



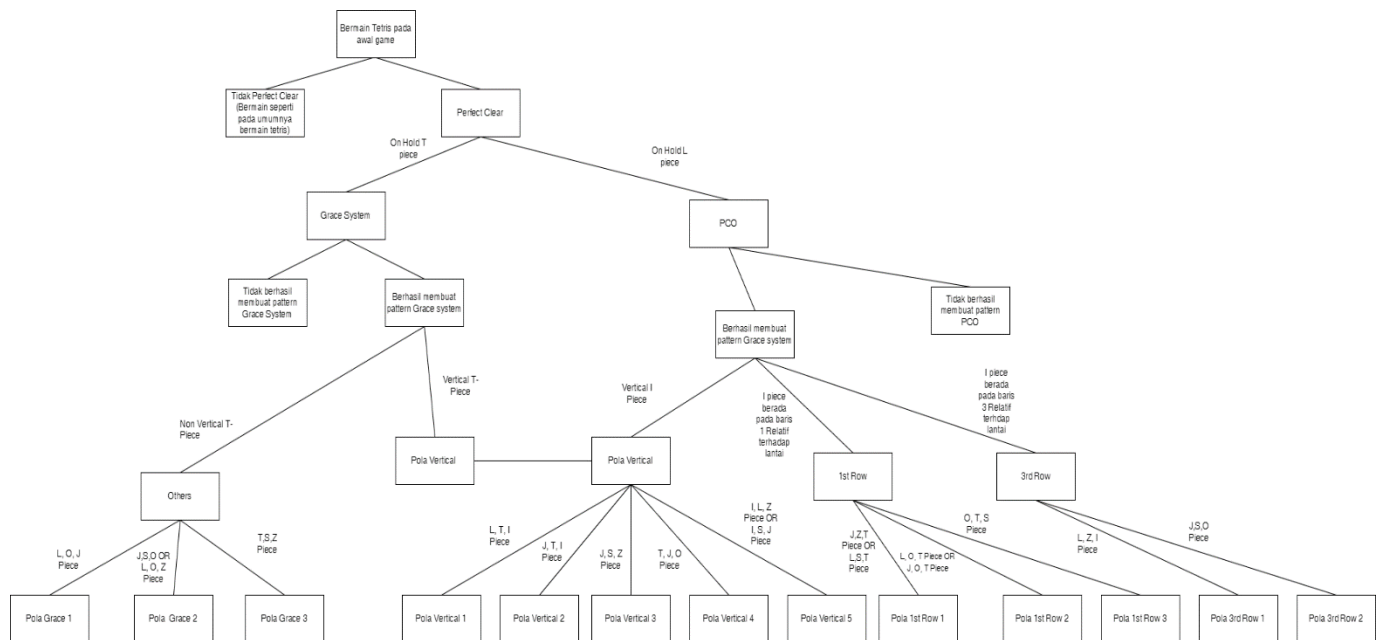
Gambar 3.4 Pola 1st Row (Sumber: arsip penulis)

Lain halnya jika pemain memilih untuk menaruh *I-Piece* pada baris ke 3. Pada kasus ini pemain hanya memiliki dua solusi yaitu *L, Z, I* serta *O, J, S*. Kedua solusi ini membutuhkan penempatan urutan yang tepat karena jika salah menempatkan maka *Perfect Clear* tidak akan terselesaikan dan akan meninggalkan *block* sisa. Pada setiap kemungkinan yang ada, pemain diwajibkan untuk menaruh *I-Piece* terlebih dahulu. Pada pola pertama pemain bisa menaruh *Z, L, serta I-Piece* secara berurutan. Pada pola kedua pemain harus menaruh *S atau O* lalu *J* secara berutan agar dapat terselesaikan.



Gambar 3.5 Pola 3rd Row (Sumber: arsip penulis)

Setelah penjelasan analisis terhadap setiap kasus, maka dibuatlah pohon keputusan berdasarkan analisis kasus tersebut.



IV. KESIMPULAN

Tetris adalah sebuah permainan *puzzle* yang mengedepankan *problem solving* serta *decision making* dalam memainkannya. Pemilihan pattern yang tepat dari pemain dapat mengarahkan pemain ke *perfect clear* pada awal permainan tetris ini. Dengan mengaplikasikan *decision tree* yang sudah dipelajari pada mata kuliah Matematika Diskrit maka pemain dapat menyelesaikan *game* tetris ini sehingga mendapatkan *perfect clear* nantinya. *Decision tree* yang diperoleh adalah *tree n-ary* yang awalnya akan mewakili prekondisi-prekondisi pattern yang sesuai pada game ini. Tiap cabang pada *decision tree* ini akan mengarahkan pemain untuk memilih pattern terbaik berdasarkan *piece* yang tersedia agar mendapatkan *perfect clear* akhirnya. Dengan memahami aspek-aspek dalam permainan dan konsep dari *decision tree* dalam matematika diskrit, pembaca dapat menerapkan aplikasi *decision tree* dalam pengambilan keputusan dalam basis permainan ini. Penulis juga mengharapkan bahwa aplikasi *decision tree* ini tidak hanya terbatas di kegiatan-kegiatan ringan seperti permainan, namun juga dapat digunakan di kehidupan sehari-hari dalam pengambilan keputusan ataupun dimanfaatkan dalam ranah akademisi ke depannya.

V. UCAPAN TERIMA KASIH

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa. Atas berkat dan rahmat-Nya, maka makalah “Aplikasi Pohon Keputusan pada Tetris agar Menghasilkan *Perfect Clear* Berdasarkan *Piece* yang Tersedia pada Awal Game” ini dapat diselesaikan dengan tepat waktu. Penulis juga mengucapkan terima kasih kepada Bapak Dr. Ir.Rinaldi Munir, M.T. sebagai dosen pengajar Mata Kuliah IF2120 Matematika Diskrit Kelas 01 atas bimbingan dan pengajaran yang tak henti-hentinya dilakukan di kelas terkait teori-teori dasar yang ada di makalah ini. Penulis juga berterimakasih atas pengingat dari beliau untuk senantiasa menuliskan dan menyelesaikan makalah ini dengan tepat waktu.

Penulis juga mengucapkan terima kasih kepada beliau karena telah mengelola situs berisi referensi dan sumber pembelajaran mata kuliah Matematika Diskrit bagi penulis dan sesama mahasiswa Teknik Informatika ITB lainnya. Konten yang ada di situs tersebut dimanfaatkan sebagai sumber informasi dan gambar pada makalah ini. Terakhir, penulis ingin mengucapkan terima kasih kepada orang tua, keluarga, dan rekan-rekan sesama mahasiswa Teknik Informatika yang telah mendukung dan menyemangati penulis dalam pengerjaan makalah ini.

REFERENCES

- [1] Pajitnov, Alexey. 1985. *Tetris*. <https://tetris.com/play-tetris> diakses pada tanggal 12 Desember 2021
- [2] Setiawan, Renalto. 2020. *Sejarah Tetris, Gim Paling Populer Buatan Negara Komunis* <https://tirto.id/sejarah-tetris-gim-paling-populer-buatan-negara-komunis-fE4x> diakses pada tanggal 12 Desember 2021
- [3] Ramos, Jeff. 2019. *Tetris advanced guide: -spins, perfect clears, and combos*. <https://www.polygon.com/guides/2019/3/6/18244590/tetris-t-spin-perfect-clear-combo-guide/> diakses pada tanggal 12 Desember 2021
- [4] Carisinyal. 2019. *Yuk Kenali Berbagai Jenis Game Online Terbaru* <https://carisinyal.com/jenis-game-online-terbaru/> diakses pada tanggal 13 Desember 2021
- [5] Davdav1233. 2018. *Perfect Clear Opener*. <https://four.lol/perfect-clears/opener> diakses pada tanggal 12 Desember 2021
- [6] Munir, Rinaldi. 2021. *Graf (Bag. 1): Bahan Kuliah IF2120 Matematika Diskrit*. Diakses pada tanggal 14 Desember 2021
- [7] Munir, Rinaldi. 2021. *Pohon (Bag. 1): Bahan Kuliah IF2120 Matematika Diskrit*. Diakses pada tanggal 14 Desember 2021
- [8] Munir, Rinaldi. 2021. *Pohon (Bag. 2): Bahan Kuliah IF2120 Matematika Diskrit*. Diakses pada tanggal 14 Desember 2021
- [9] Rosen, Kenneth H. 2019. *Discrete Mathematics and Its Applications, Eighth Edition*. New York: Mc Graw Hill Education.

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, 13 Desember 2020

A handwritten signature in black ink, consisting of the letters 'G', 'm', and 'e' in a cursive style. The 'G' is large and loops back, the 'm' is smaller and connected to the 'G', and the 'e' is a simple cursive flourish.

Muhammad Garebaldhie ER Rahman 13520029