# Analisis Penerapan Pohon Keputusan (*Decision Tree)* dalam Penyelesaian Permainan Minesweeper

Syarifa Dwi Purnamasari - 13521018
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*[1]13521018@std.stei.itb.ac.id*

***Abstrak*—Minesweeper merupakan salah satu permainan teka-teki logika bertipe *single-player* yang pertama kali diluncurkan pada tahun 1990 oleh Microsoft dan terinspirasi oleh *video game* Mined-Out dari Ian Andrew untuk ZX Spectrum. Permainan ini berbentuk area papan yang berisi sekumpulan petak dan dari beberapa petak tersebut terdapat petak yang berisikan ranjau didalamnya. Tujuan dari permainan ini adalah untuk membuka setiap petak yang ada kecuali petak yang berisikan ranjau dengan *clue* petak berisikan angka yang menyatakan jumlah ranjau yang ada di dekatnya. Terdapat beberapa algoritma yang dapat digunakan untuk menyelesaikan permainan tersebut, salah satunya dengan pohon keputusan (*decision tree)*. Pohon keputusan merupakan salah satu metode pengklasifikasian data dengan struktur seperti pohon, dimana setiap node menyatakan atribut, cabangnya merepresentasikan *outcome* dari atribut, dan tiap daun node nya digunakan untuk menunjukkan kelas.**

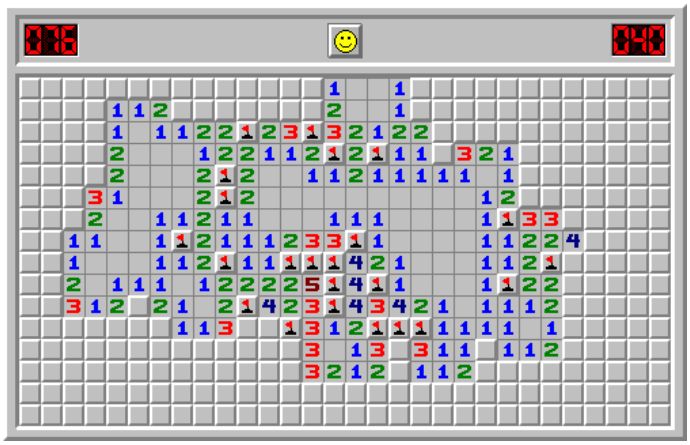
***Kata Kunci*—algoritma, *decision tree*, matematika diskrit, minesweeper.**

## I. PENDAHULUAN

Minesweeper merupakan permainan *video game* bergenre *logic puzzle* yang dimainkan secara tunggal dan menjadi salah satu permainan paling sukses yang pernah dibuat. Menurut *TechRadar*, minesweeper dirilis oleh Microsoft pada tahun 1990-an sebagai bagian dari Windows Entertainment Pack sebelum akhirnya dipromosikan sebagai fitur standar dari Windows 3.1, di samping permainan Solitaire. Akan tetapi, *Eurogamer* berpendapat bahwa minesweeper yang diluncurkan secara resmi oleh Microsoft banyak terinspirasi dari *video game* Mined-Out dari Ian Andrew untuk ZX Spectrum yang telah dibuat sejak tahun 1983.

Permainan minesweeper berbentuk area papan *grid* yang pada beberapa petaknya berisi ranjau dan tersebar di seluruh papan. Setiap petak dapat berisi angka, ranjau, ataupun kosong. Setiap petak juga memiliki tiga kondisi: belum dibuka, telah dibuka, atau ditandai. Petak yang belum dibuka merupakan petak yang harus ditebak isinya, petak yang sudah dibuka dan berisi angka merupakan petak yang dijadikan *clue* untuk membuka petak yang tertutup di sekitarnya, sedangkan petak yang ditandai merupakan petak tertutup yang ditandai oleh pemain sebagai petak yang berisikan ranjau. Objektif dari permainan ini adalah membuka semua petak selain petak yang berisikan ranjau. Permainan ini bukan merupakan jenis permainan yang berpacu dengan skor atau durasi waktu tertentu, tetapi ketelitian dan kecermatan dalam membaca informasi pada tiap petak. Tingkat kesulitan pada permainan ini dapat ditingkatkan dengan memilih petak dengan jumlah yang lebih besar atau menambahkan jumlah ranjau.

Permainan ini dimulai ketika pemain pertama kali memilih petak secara bebas yang terdapat di papan dan petak yang dibuka tersebut akan membuka beberapa petak lain sebagai informasi untuk langkah selanjutnya. Angka diantara 1-8 pada petak yang berisikan angka merupakan petunjuk jumlah ranjau yang terdapat di sekitar petak angka tersebut. Pemain dapat menandai petak yang dirasa berisi ranjau di dalamnya dan membuka petak yang dirasa berisi angka. Apabila pemain salah dalam memilih dan membuka petak berisi ranjau, permainan berakhir.



**Gambar 1.1.** Permainan Minesweeper
Sumber https://minesweeper.online/

Permainan minesweeper sendiri pada dasarnya memiliki kompleksitas algoritma yang setara dengan persoalan NP (*Nondeterministic Polinomial*). *Nondeterministic* sendiri mengartikan bahwa tidak ada aturan tertentu yang dapat diikuti untuk menyelesaikan permasalahan yang juga secara tidak langsung menandakan bahwa permainan ini menjadi salah satu persoalan matematika yang cukup sulit untuk dipecahkan.

Pada makalah ini, penulis akan membahas lebih lanjut mengenai analisis aplikasi pohon keputusan (*decision tree*) dalam penyelesaian permainan minesweeper. Pohon keputusan (*decision tree*) sendiri merupakan salah satu algoritma dengan

struktur pohon yang cukup sering digunakan dalam memodelkan data dalam menyelesaikan suatu persoalan.

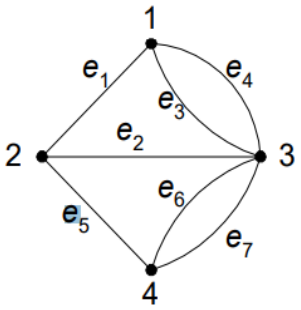
## II. Landasan Teori

### A. *Graf*

Graf merupakan pasangan tak-terurut dari *vertex* ($V$) dan *edge* ($E$) dan dapat dituliskan sebagai $G = (V, E)$ yang dalam hal ini:

$V$ = himpunan tidak-kosong dari simpul-simpul (*vertices*)
= $\{v_1, v_2, \ldots, v_n\}$

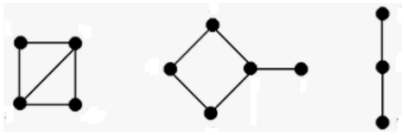
$E$ = himpunan sisi (*edges*) yang menghubungkan sepasang simpul
= $\{e_1, e_2, \ldots, e_n\}$




**Gambar 2.1.** Graf
Sumber
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

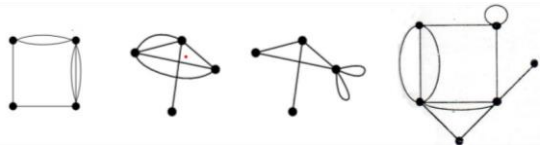
Berdasarkan ada tidaknya gelang (*loop*) atau sisi-ganda (*multiple edges*) pada suatu graf, graf dapat digolongkan menjadi dua jenis:

1. Graf sederhana *(simple graph)*
   Graf yang tidak memiliki gelang maupun sisi-ganda.



**Gambar 2.2.** Graf Sederhana
Sumber
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

2. Graf tak-sederhana (*unsimple graph*)
   Graf yang memiliki gelang, sisi-ganda, ataupun keduanya. Graf tak-sederhana dapat dibedakan menjadi menjadi 2 jenis yaitu:
   a. Graf Ganda (*multi-graph*)
      Graf yang memiliki sisi ganda (*multiple edges*) dan tidak memilik sisi gelang
   b. Graf Semu (*pseudo-graph*)
      Graf yang memiliki sisi kalang atau gelang (*loop*) dan dapat memiliki sisi ganda juga.



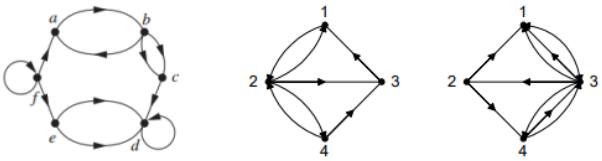
**Gambar 2.3.** Graf Tak-Sederhana
Sumber
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

Berdasarkan orientasi arah pada sisi, graf dapat dibedakan menjadi dua jenis:

1. Graf tak-berarah *(undirected graph)*
   Graf yang setiap sisinya tidak mempunyai orientasi arah.

**Gambar 2.4.** Graf Tak-Berarah
Sumber
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

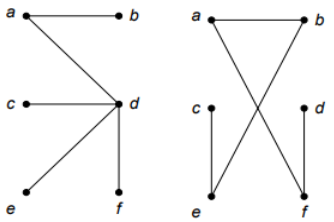
2. Graf berarah (*directed graph* atau *digraph*)
   Graf yang setiap sisinya mempunyai orientasi arah.




**Gambar 2.5.** Graf Berarah
Sumber
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

### B. *Pohon*

Pohon merupakan graf terhubung tak-berarah (*undirected graph*) dan tidak memiliki sirkuit maupun sisi-ganda.




**Gambar 2.6.** Pohon
Sumber
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf

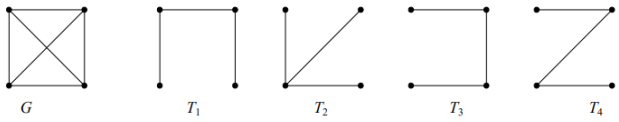
Misalkan $G = (V, E)$ merupakan graf tak-berarah sederhana dan jumlah simpulnya n, pernyataan di bawah ini adalah *equivalent*.

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

## C. *Pohon Merentang*

Pohon Merentang (*Spanning Tree*) dari graf terhubung adalah upagraf merentang yang berupa pohon dan didapatkan dengan memutus sirkuit di dalam graf. Setiap graf terhubung setidaknya memiliki satu buah pohon merentang.

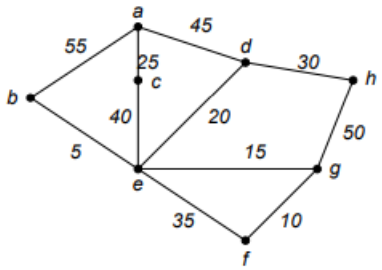
Aplikasi *spanning tree* sering diterapkan pada perutean (routing) pesan pada jaringan komputer.




**Gambar 2.7.** Graf (*G*) dan Pohon Merentangnya (*T*)
Sumber
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf

## D. *Pohon Merentang Minimum*

Pohon Merentang Minimum (*Minimum Spanning Tree)* merupakan pohon merentang berbobot minimum dari graf terhubung-berbobot. Dalam menentukan *minimum spanning tree*, terdapat dua algoritma yang paling sering digunakan yairu Algoritma Prim dan Algoritma Kruskal.

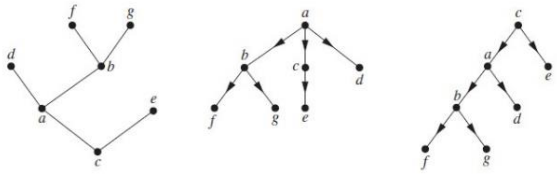



**Gambar 2.8.** Graf Terhubung-Berbobot
Sumber
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf

## E. *Pohon Berakar*

Pohon Berakar (*Rooted Tree*) merupakan pohon yang sebuah simpulnya dijadikan sebagai simpul utama (akar) dan orientasi arah semua sisi serta simpul lainnya menjauhi simpul utama tersebut.

Pada dasarnya, pohon berakar termasuk graf berarah. Akan tetapi, sudah ditetapkan bahwa pada penggambarannya untuk tidak ditunjukkan orientasi arah karena sifatnya.



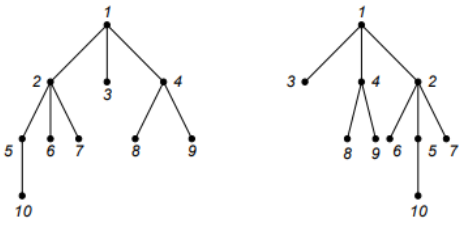
**Gambar 2.9.** Pohon Berakar
Sumber
Discrete Mathematics & Mathematical Reasoning, Chapter 11: Trees, U. of Edinburgh, UK

Beberapa terminologi pada pohon berakar dengan rujukan graf ketiga pada Gambar 2.9, yaitu:

1. Anak (*child/children*) dan Orang Tua (*parent*)
   Simpul a dan e merupakan anak dari simpul c. Simpul c merupakan orang tua dari simpul a dan e.
2. Lintasan (*path*)
   Lintasan dari c ke d adalah c, a, d. Panjang lintasan dari a ke f adalah dua.
3. Saudara Kandung (sibling)
   Simpul f dan g merupakan saudara kandung karena memiliki orang tua yang sama yaitu b.
4. Upapohon (*subtree*)
   Graf yang dibentuk oleh simpul b, f, g merupakan salah satu upapohon.
5. Derajat (*degree*)
   Derajat sebuah simpul merupakan jumlah anak atau upapohon simpul tersebut. Derajat c, a, dan b adalah 2, sedangkan derajat e, d, f, dan g adalah 0.
6. Daun (*leaf*)
   Daun merupakan simpul yang memiliki derajat 0 yaitu e, d, f, dan g.
7. Simpul Dalam (*internal nodes*)
   Simpul dalam merupakan simpul yang memiliki anak yaitu simpul c, a, dan b.
8. Aras (*level*) atau Tingkat
   Aras atau tingkat merupakan angka yang dihitung mulai dari akar dan bertambah pada tiap anaknya. Aras akar bernilai 0.
9. Tinggi (*height*) atau Kedalaman (*depth*)
   Tinggi atau kedalaman merupakan aras maksimum yang terdapat pada sebuah pohon berakar. Tinggi pohon tersebut adalah 3.

## F. *Pohon Terurut*

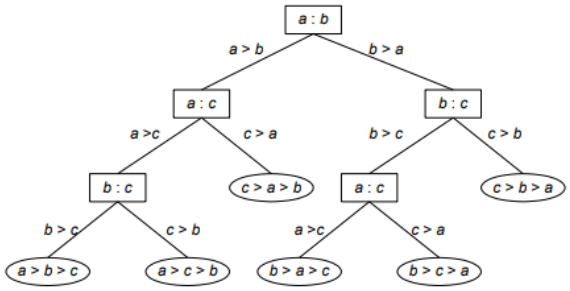
Pohon Terurut (*Ordered Tree*) merupakan pohon berakar yang mempertimbangkan urutan anaknya sehingga urutan anak-anaknya menjadi penting.



**Gambar 2.10.** Pohon Terurut
Sumber
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Pohon-2021-Bag2.pdf

## G. *Pohon Biner*

Pohon Biner (*Binary* Tree) merupakan pohon yang jumlah anak maksimalnya pada setiap simpul adalah 2. Pada *binary tree*, anak kiri biasa disebut dengan *left child* dan anak kanan disebut dengan *right child*. Pohon biner merupakan pohon terurut karena perbedaan urutan anak.



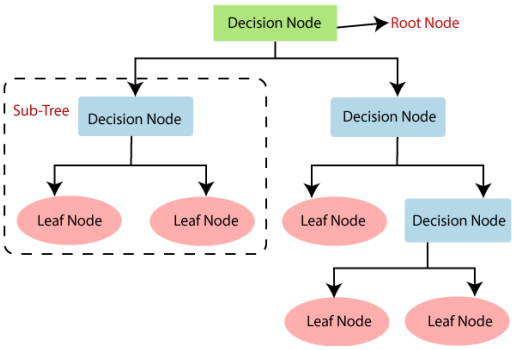

**Gambar 2.11.** Pohon Biner
Sumber
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Pohon-2021-Bag2.pdf

*Binary tree* merupakan salah satu algoritma yang terkenal dan dapat diaplikasikan pada beberapa algoritma sebagai berikut.

1. Pohon Ekspresi
   Pohon yang terdiri atas operan pada bagian daun dan operator pada bagian simpul dalam dan akar. Pohon ini biasa digunakan untuk menyatakan ekspresi aritmatika.
2. Pohon Keputusan
   Pohon yang digunakan untuk mengeksplorasi data dengan metode perbandingan hasil pada sisi dan bagian daun sebagai representasi hasil akhir dari keputusan.
3. Kode Awalan
   Pohon yang sisinya teridiri atas bobot 0 dan 1 dan daunnya menunjukkan himpunan bobot sesuai lintasan yang dilalui.
4. Kode Huffman
   Pohon yang digunakan untuk men-*compress* rangkaian teks melalui pengkodean dalam bentuk bit untuk mewakili data karakter. Cara kerja dari algoritma ini ialah dengan mengkombinasikan dua buah frekuensi/peluang karakter yang muncul paling kecil dalam sebuah rangkaian teks dan hal ini dilakukan secara iteratif sampai seluruh karakter habis. Lintasan dari akar ke daun yang tiap sisinya menunjukkan bobot menjadi kode Huffman pada daun tersebut.
5. Pohon Pencarian Biner (*Binary Search Tree*)
   Pohon yang digunakan sebagai metode pencarian data dengan eliminasi biner secara iteratif. Kunci (*key*) akan selalu dibandingkan dengan data yang berada di tengah, apabila belum ditemukan data yang sama dan data kunci lebih kecil, telusuri anak kiri, sedangkan apabila data kunci lebih besar, telusuri anak kanan.

## H. *Pohon Keputusan*

Pohon keputusan (*Decision Tree*) merupakan salah satu metode pengklasifikasian data dengan struktur seperti pohon, dimana setiap node menyatakan atribut, cabangnya merepresentasikan *outcome* dari atribut, dan tiap daun node nya digunakan untuk menunjukkan kelas.




**Gambar 2.12.** Pohon Keputusan
Sumber
https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

Mengutip dari *Venngage*, *Decision Tree* memiliki tiga elemen, yaitu:

1. Akar (*Root Node*)
   Keputusan besar yang akan diambil dan merupakan simpul utama.
2. Cabang/Ranting (*Branches*)
   Berbagai opsi tindakan yang akan diambil.
3. Daun (*Leaf Node*)
   Probabilitas hasil atas setiap tindakan dan digolongkan menjadi 2 jenis yaitu:
   a. *Leaf Node* berbentuk persegi merepresentasikan keputusan yang diambil (*Decision Node*).
   *b.* *Leaf Node* berbentuk lingkaran merepresentasikan hasil yang tidak pasti

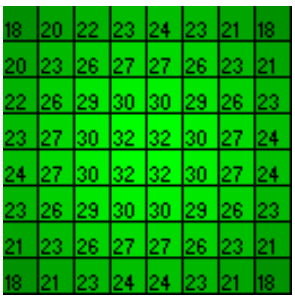
# III. STRATEGI DALAM PERMAINAN MINESWEEPER

## A. *First Click*

Pada permainan *Windows Minesweeper*, petak pertama yang dipilih dalam papan (*first click*) selalu aman dalam artian dapat dipastikan petak yang dipilih tidak berisikan ranjau. Akan tetapi, langkah pertama begitu penting untuk menentukan langkah selanjutnya, sehingga *first click* menentukan kualitas dan kuantitas informasi papan selanjutnya yang akan dibuka.

Pada 2002, Tim Kostka menganalisis dan mengkalkulasikan *probability* dan *average size* dari setiap petak dalam papan minesweeper dijadikan sebagai *first click* dalam permainan. Dari hasil analisis, probabilitas bukaan dari setiap petak pertama yang dibuka meningkat dari *corner* menuju daerah sisi, tetapi

ukuran bukaan meningkat jauh lebih besar lagi menuju daerah tengah papan. Hal ini dapat terjadi karena petak bagian tengah papan memiliki jumlah petak tetangga yang lebih banyak dibandingkan petak di bagian sisi maupun sudut, sehingga membuat ukuran bukaan pertama menjadi lebih besar.




**Gambar 3.1.** Rata-rata ukuran bukaan setiap petak pada level *beginner*
Sumber https://minesweepergame.com/strategy/first-click.php

Jika pemain lebih mementingkan kuantitas ukuran bukaan pada *first click* permainan, mungkin memilih petak pada bagian tengah papan menjadi pilihan yang terbaik, tetapi jika pemain lebih mementingkan kualitas dari informasi bukaan, petak pada bagian sudut mungkin dapat menjadi opsi terbaik lain yang bisa dipilih. Hal ini dikarenakan petak bagian sudut sering kali menjadi petak yang secara logika lebih sulit ditebak dibandingkan petak pada posisi lain karena minimnya informasi dari petak disekitarnya. Apabila kita bandingkan, petak bagian tengah memiliki maksimum 8 petak lain yang dapat menjadi informasi lanjutan bagi petak tengah yang belum dibuka tersebut, sedangkan petak bagian sudut hanya memiliki maksimum 3 petak lain yang dapat menjadi informasi lanjutan, sehingga dapat disimpulkan, semakin sedikit petak sudut yang masih tertutup, semakin besar kesempatan menyelesaikan persoalan petak yang sulit.

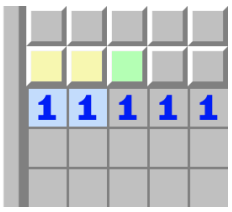
B. *Minesweeper Patterns*

Seiring dengan bertambahnya frekuensi dalam bermain minesweeper, pemain juga akan makin mengenali medan area papan serta terbiasa dengan pola petak angka dan berbagai kombinasi tertentu. Hal ini sangat membantu dalam mempelajari *patterns* yang ada dalam permainan sehingga mampu mengurangi waktu dalam berpikir dan menganalisis petak yang masih tertutup.

Pada permainan minesweeper terdapat beberapa *patterns* yang dapat digunakan antara lain *basic patterns, reduction, holes, triangle, high complexity patterns,* dan *last turns.* Pada tulisan ini, penulis hanya akan menjelaskan tentang beberapa pola yang sering ditemui dan menjadi dasar dalam permainan minesweeper.

1. *Basic Patterns*
   a. *Pattern* 1-1

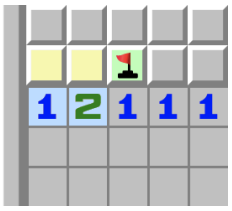
      Jika terdapat dua buah angka 1 pada tepi papan, maka dapat kita simpulkan salah satu petaknya (area kuning) merupakan petak berisi ranjau (*mine*) dan petak hijau merupakan petak yang aman.




**Gambar 3.2.** Gambar *Pattern* 1-1
Sumber https://minesweeper.online/help/patterns

b. *Pattern* 1-2

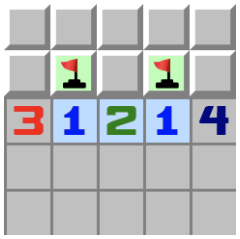
Pola mirip dengan 1-1 akan tetapi petak hijau dapat dipastikan sebagai *mine*.




**Gambar 3.3.** Gambar *Pattern* 1-2
Sumber https://minesweeper.online/help/patterns

c. *Pattern* 1-2-1

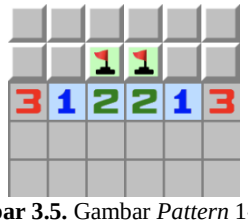
Jika petak di belakang 3 petak angka berisi 1-2-1 merupakan petak aman, maka petak di depan angka 1 dapat dipastikan sebagai *mine*.




**Gambar 3.4.** Gambar *Pattern* 1-2-1
Sumber https://minesweeper.online/help/patterns

d. *Pattern* 1-2-2-1

Jika petak di belakang 4 petak angka berisi 1-2-2-1 merupakan petak aman, maka petak di depan angka 2 dapat dipastikan sebagai *mine*.



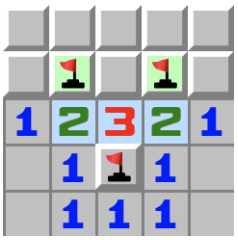

**Gambar 3.5.** Gambar *Pattern* 1-2-2-1
Sumber https://minesweeper.online/help/patterns

2. *Reduction*

*Reduction* merupakan pola dalam permainan minesweeper dengan mengurangkan angka pada petak dengan jumlah *mine* disekitar petak angka tersebut dan mengembalikannya sebagai *basic patterns*.

Gambar 3.6. merupakan contoh pola *reduction* yang menunjukkan petak angka berpola 2-3-2 yang berubah menjadi 1-2-1 ketika dikurangkan

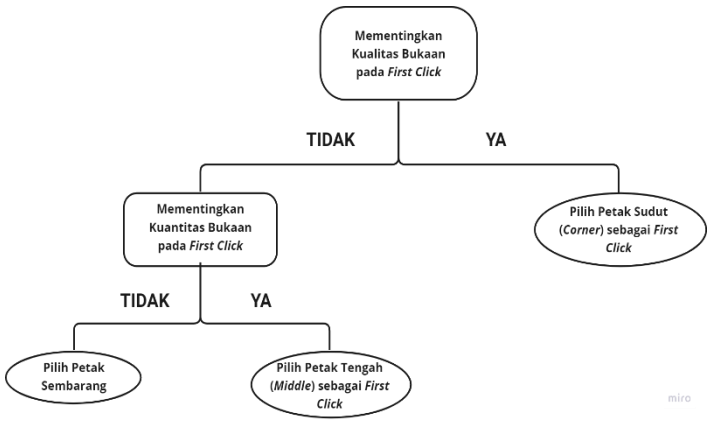
dengan jumlah *mine* disekitarnya yang telah diketahui.



**Gambar 3.6.** Gambar *Pattern* 1-2-1R
Sumber https://minesweeper.online/help/patterns

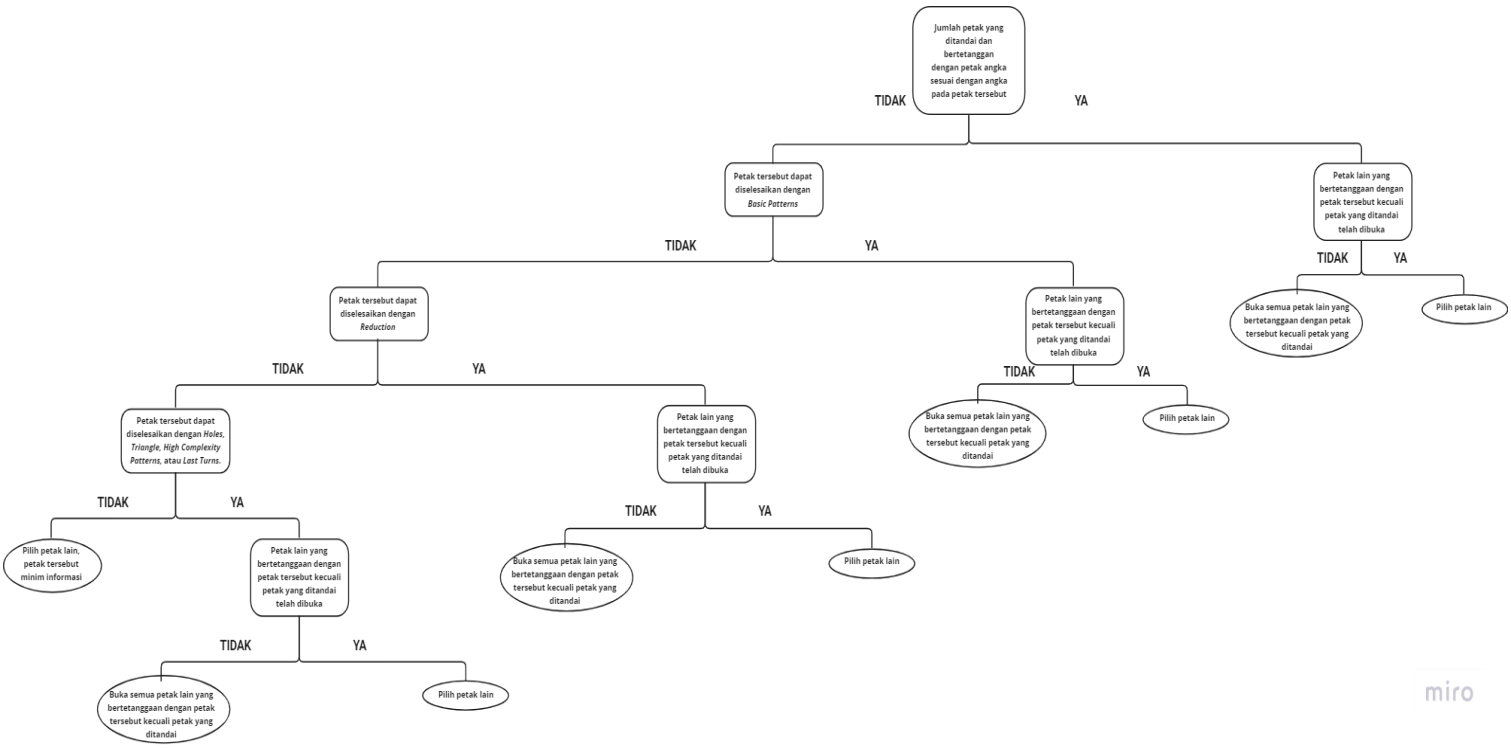
## IV. Aplikasi Pohon Keputusan dalam Permainan Minesweeper

Setelah mempelajari strategi yang dapat diterapkan dalam menyelesaikan permainan minesweeper, pohon keputusan dapat dibuat dengan memanfaatkan teori serta pola tersebut. Pada strategi *first click*, pohon keputusan dapat dibuat dengan dua pilihan. Apabila pemain lebih mementingkan kuantitas dari bukaan petak pertama, pemain akan sangat disarankan untuk memilih petak bagian tengah karena menurut analisis yang telah dijelaskan sebelumnya, petak bagian tengah memiliki jumlah bukaan *first click* yang paling besar. Akan tetapi, jika pemain lebih mementingkan kualitas informasi yang ada di dalam petak, pemain hendaknya memilih petak bagian sudut karena petak bagian sudut sering kali menjadi *unsolvable positions*.




**Gambar 4.1.** Pohon Keputusan pada Penentuan *First Click*
Sumber Dokumen Pribadi

Setelah memilih petak pertama yang akan dipilih, minesweeper akan membuka beberapa petak lain yang akan menjadi petunjuk sekaligus informasi untuk langkah selanjutnya. Dari beberapa petak lain yang sudah terbuka pada bukaan pertama tersebut, pemain dapat membaca informasi pada petak angka untuk menebak *mine* yang ada pada sekitar petak angka tersebut. Dengan mengamati *patterns* yang telah dijelaskan sebelumnya, pemain dapat lebih cepat dalam menentukan petak mana yang aman untuk dibuka dan petak mana yang harus ditandai. Namun, terkadang butuh pemahaman lebih dari sekedar memanfaatkan pola yang ada untuk memecahkan solusi dari minesweeper. Ada beberapa petak yang sulit untuk ditebak karena minimnya informasi.




**Gambar 4.2.** Pohon Keputusan pada Tiap Petak yang Telah Terbuka
Sumber Dokumen Pribadi

## V. KESIMPULAN

Permainan minesweeper merupakan salah satu permainan teka-teki logika yang penyelesaiannya dapat dipecahkan dengan menggunakan pohon keputusan (*decision tree*). Selain ketelitian dan kecermatan, permainan minesweeper juga memerlukan strategi yang mampu mempermudah pemain dalam memecahkan teka-teki petak serta mengurangi waktu berpikir pemain sehingga permainan mampu lebih cepat diselesaikan. Pohon keputusan dapat digunakan dalam menentukan strategi yang tepat dengan memanfaatkan teori *first click* serta *patterns* yang ada. Dengan pohon keputusan, pemain dapat meninjau ulang langkah yang ia lakukan dan menggunakannya secara iteratif sehingga tujuan permainan dapat diselesaikan hingga akhir.

## VI. UCAPAN TERIMA KASIH

Pertama dan yang paling utama, rasa syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas rahmat-Nya, penulis dapat menyelesaikan makalah ini dengan tepat waktu. Penulis juga ingin mengucapkan terima kasih kepada seluruh tim pengajar IF2120 Matematika Diskrit Institut Teknologi Bandung, terutamanya Bapak Dr. Ir. Rinaldi Munir, M.T., selaku Dosen Pembimbing K03 Matematika Diskrit karena tanpa bimbingan beliau, makalah ini tidak dapat terselesaikan dengan baik. Di samping itu, penulis juga ingin mengucapkan rasa terima kasih kepada orang tua yang selalu mendukung penulis baik secara morel maupun materiel, serta teman-teman yang selalu memberi dukungan kepada penulis sehingga penulis mampu menyelesaikan makalah ini.

## DAFTAR PUSTAKA

[1] Etessami, Kousha. *Discrete Mathematics & Mathematical Reasoning Chapter 11: Trees U. of Edinburgh, UK.* https://www.inf.ed.ac.uk/teaching/courses/dmmr/slides/18-19/Ch11.pdf. Diakses pada 10 Desember 2022.

[2] Gupta, Saloni. *Decision Tree.* https://www.geeksforgeeks.org/decision-tree. Diakses pada 10 Desember 2022.

[3] Javatpoint. *Decision Tree Classification Algorithm.* https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm. Diakses pada 10 Desember 2022.

[4] Minesweeper Online. *Patterns.* https://minesweeper.online/help/patterns. Diakses pada 11 Desember 2022.

[5] Munir, Rinaldi. 2020. *Bahan Kuliah IF2120 Matematika Diskrit: Graf (Bagian: 1).* https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf. Diakses pada 11 Desember 2022.

[6] Munir, Rinaldi. 2020. *Bahan Kuliah IF2120 Matematika Diskrit: Pohon (Bagian: 1).* https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf. Diakses pada 11 Desember 2022.

[7] Munir, Rinaldi. 2020. *Bahan Kuliah IF2120 Matematika Diskrit: Pohon (Bagian: 2).* https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Pohon-2021-Bag2.pdf. Diakses pada 11 Desember 2022.

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Desember 2022

Syarifa Dwi Purnamasari 13521018