

Penerapan Pohon Keputusan dalam Permainan *Hidden Panel*

Muhammad Fikri. N and 13519069
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13519069@std.stei.itb.ac.id

Abstrak—Penerapan pohon dapat digunakan untuk memodelkan masalah dalam pengambilan keputusan. Pemodelan tersebut digunakan untuk mencari solusi yang tepat dari sebuah permasalahan. Pohon tersebut memiliki simpul yang berkorespondensi dengan sebuah keputusan. Simpul pada sub pohon merupakan kemungkinan dari hasil keputusan. Solusi yang mungkin dari sebuah masalah sesuai dengan jalur atau jalan menuju daun tersebut. Pada makalah ini akan dibahas pengaplikasian pohon keputusan untuk menyelesaikan salah satu permainan dalam game *Brain Training* yaitu *hidden panel* menggunakan analisis kasus dan parameter-parameter lainnya yang dapat membantu.

Kata kunci—*Brain, Training, Hidden, Pannel, pohon, keputusan.*

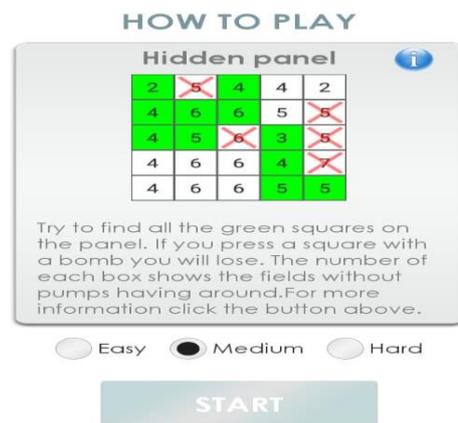
I. PENDAHULUAN

Brain Training adalah sebuah permainan yang dikembangkan oleh untuk membantu pemain mengevaluasi serta mengasah kemampuan dalam hal mengingat, menganalisis, ketepatan, ketajaman, dan persepsi. Dalam aplikasi ini, terdapat banyak kumpulan permainan yang menantang untuk dimainkan.



Gambar 1 : Tampilan game *brain training*.
Sumber : dokumen penulis

Salah satu permainan dalam aplikasi tersebut adalah *hidden panel*. Permainan ini tergolong dalam kelompok *analyze*. Untuk memainkan permainan ini, pemain harus dapat menemukan kotak berwarna hijau tanpa membuka kotak yang berisi silang berwarna merah dengan melakukan analisis pada keseluruhan kotak. Jika pemain membuka kotak yang berisi silang berwarna merah tersebut maka permainan akan berakhir dan pemain gagal memenangkan permainan tersebut. Untuk menyelesaikan permainan ini, pemain harus membuka kotak yang tepat sehingga tidak memicu kotak yang berisi silang merah. Dalam menyelesaikan permainan *hidden panel* ini, pemain akan diberikan petunjuk angka-angka pada setiap kotak yang merepresentasikan banyaknya kotak hijau di sekitar kotak tersebut.



Gambar 2 : Panduan bermain *hidden panel*
Sumber : dokumen penulis

Pada awal permainan, pemain diberikan sekumpulan kotak dengan ukuran yang berbeda-beda tergantung tingkat kesulitan yang dipilih oleh pemain. Pada tingkat kesulitan *easy*, pemain diberikan 64 kotak dengan ukuran 8 x 8. Pada tingkat kesulitan *medium*, pemain diberikan sejumlah 72 kotak dengan ukuran 8 x 9. Pada tingkat kesulitan *hard*, pemain diberikan sebanyak 90 kotak dengan ukuran 9 x 10. Kotak-kotak tersebut jika dibuka maka ada dua kemungkinan yaitu kotak berwarna atau kotak yang berisi silang berwarna merah. Untuk

selanjutnya, silang berwarna merah akan disebut sebagai *cross* dan kotak hijau akan disebut sebagai *green*.



1	2	3	1	2	2	2	1
3	4	4	4	5	4	6	4
2	3	5	5	5	5	5	3
3	5	4	5	6	7	6	4
3	5	5	7	6	5	6	3
2	5	3	5	5	5	5	2
3	5	4	6	5	5	4	3
3	4	4	3	5	5	6	3
3	3	3	2	3	3	4	2

Gambar 3 : Tampilan awal permainan *hidden panel*.
Sumber : dokumen penulis

hidden panel adalah permainan uang memanfaatkan analisis, logika, dan peluang (terutama ketika *current state* tidak memberikan keputusan yang pasti). Kita dapat menentukan pola-pola yang dapat membantu untuk memenangkan permainan ini serta mengambil keputusan yang tepat.

II. TEORI DASAR

A. Graf

Graf adalah struktur diskrit yang terdiri dari *vertex* / simpul dan *edge* / sisi yang terhubung dengan simpul tersebut. Sebuah graf $G = (V, E)$ terdiri dari V yaitu himpunan tidak kosong dari simpul dan E adalah himpunan sisi. Setiap sisi memiliki satu atau dua simpul yang terkait dengannya, yang disebut sebagai endpoints atau titik ujung. Sebuah *edge* / sisi dikatakan menghubungkan titik ujungnya.

Jenis – jenis Graf :

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf dapat digolongkan menjadi dua jenis yaitu

1. Graf sederhana (*simple graph*)
Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi ganda.
2. Graf tak-sederhana (*unsimple-graph*)
Graf tak-sederhana adalah graf yang memiliki sisi ganda atau gelang.

Macam – macam graf tak-sederhana :

- a) Graf ganda (*multi graph*)
Graf ganda adalah graf yang mengandung sisi ganda.
- b) Graf semu (*pseudo-graph*)
Graf semu adalah graf yang mengandung sisi gelang.

Berdasarkan ada tidaknya arah pada suatu graf, maka graf dapat digolongkan menjadi dua jenis yaitu

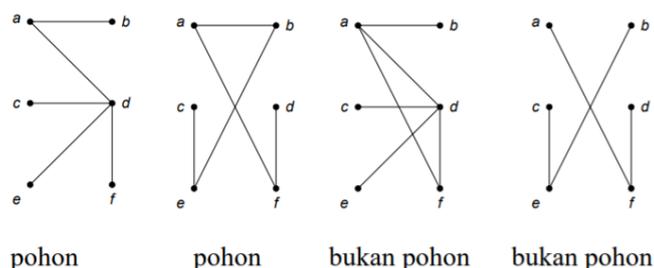
1. Graf tak-berarah (*undirected graph*)
Graf tak-berarah adalah graf yang sisinya tidak memiliki orientasi arah.
2. Graf berarah (*direct graph* atau *digraph*)
Graf berarah adalah graf yang setiap sisinya memiliki arah.

Dalam memodelkan masalah dalam pengambilan keputusan, graf yang digunakan adalah graf sederhana tak-berarah.

B. Pohon

Pohon adalah graf yang terhubung sedemikian sehingga tidak mengandung sirkuit serta tak memiliki arah / orientasi. Hutan adalah kumpulan pohon yang saling lepas atau graf tidak terhubung yang tidak mengandung sirkuit. Setiap komponen di dalam graf terhubung tersebut adalah pohon. Misalkan terdapat suatu graf tak-berarah sederhana $G = (V, E)$ dan jumlah simpulnya n . Maka semua pernyataan di bawah ini adalah ekuivalen :

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 mengandung 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 adalah jembatan

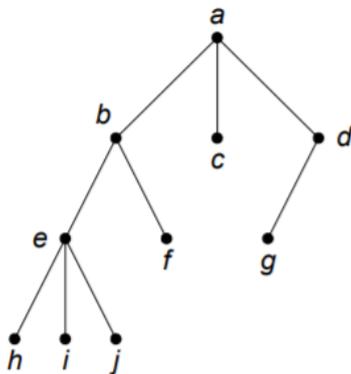


Gambar 4 : Contoh graf yang membentuk pohon
Sumber

:<http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf>

Pada gambar 4 di atas, gambar ke-3 dan gambar ke-4 bukan termasuk pohon. Pada Gambar ke-3 bukan termasuk pohon karena terdapat sirkuit dan gambar ke-4 bukan merupakan pohon karena tidak terhubung.

Jika salah satu simpul dari sebuah pohon diperlakukan sebagai akan dan sisi-sisinya diberi arah sehingga menjadi graf berarah dinamakan pohon berakar (*rooted tree*). Dalam pohon berakar, arah pada sisi-sisinya dapat dihilangkan sesuai dengan perjanjian akar tersebut.



Gambar 5 : Gambar Pohon berakar
Sumber :

<http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>

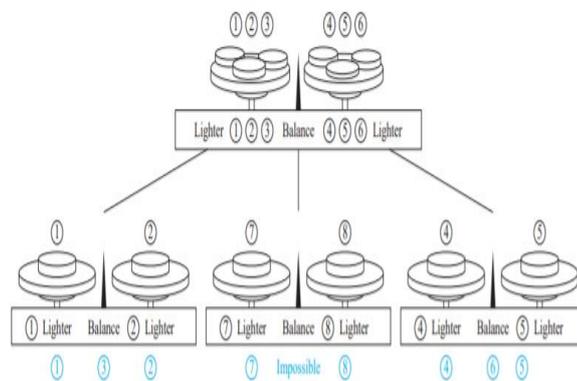
Terdapat beberapa terminologi pada pohon berakar yaitu :

1. Anak (*child* atau *children*) dan orangtua (*parent*)
Orangtua merupakan sumber dari anak tersebut dengan kata lain orangtua dihubungkan dengan sebuah sisi dari orangtua ke anak. Contoh : e dan f adalah anak dari b, e adalah orangtua dari h, i, dan j.
2. Lintasan (*path*)
Lintasan adalah jalur yang dilewati antara dua simpul pada pohon berakar. Contoh : Lintasan dari a ke j adalah a, b, e, j. Dengan panjang lintasan dari a ke j adalah 3.
3. Saudara kandung (*sibling*)
Saudara kandung adalah simpul yang memiliki orangtua yang sama. Contoh : f adalah saudara kandung e, tetapi g bukan saudara kandung e, karena orangtua mereka berbeda.
4. Upapohon (*subtree*)
Upapohon adalah pohon yang merupakan bagian dari sebuah pohon yang lebih besar. Contoh : f-b-e-h-i-j adalah *subtree* dari pohon tersebut.
5. Derajat (*degree*)
Derajat sebuah simpul adalah jumlah upapohon atau jumlah anak pada simpul tersebut. Contoh : derajat a adalah tiga, derajat b adalah dua, derajat d adalah satu, dan derajat c adalah nol. Dengan kata lain, derajat pada pohon berakar ini adalah derajat keluar. Derajat maksimum dari semua simpul merupakan derajat pohon itu sendiri. Pohon pada gambar di atas berderajat tiga

6. Daun (*leaf*)
Daun adalah simpul yang berderajat nol atau tidak mempunyai anak. Contoh : simpul h, i, j, f, c, l, dan m adalah daun.
7. Simpul dalam (*internal nodes*)
Simpul dalam adalah simpul yang mempunyai anak. Contoh : simpul b, d, e, dan k adalah simpul dalam karena memiliki anak namun f dan g tidak termasuk simpul dalam.
8. Aras (*level*) atau tingkat
Aras/tingkat adalah jarak dari akar utama menuju suatu simpul pada pohon berakar tersebut. Jarak tersebut dihitung dari banyaknya sisi yang dilewati dari akar utama ke simpul tersebut. Contoh : tingkat dari g adalah dua karena melewati dua buah sisi yaitu a-d dan d-g.
9. Tinggi (*height*) atau kedalaman (*depth*)
Tinggi atau kedalaman merupakan Aras/level terjauh sebuah pohon. Contoh : tinggi dari pohon tersebut adalah 3 karena jarak dari akar utama ke daun paling jauh adalah 3 buah sisi, salah satunya adalah a- h melewati sisi a-b, b-e, dan e-h.

C. Pohon Keputusan

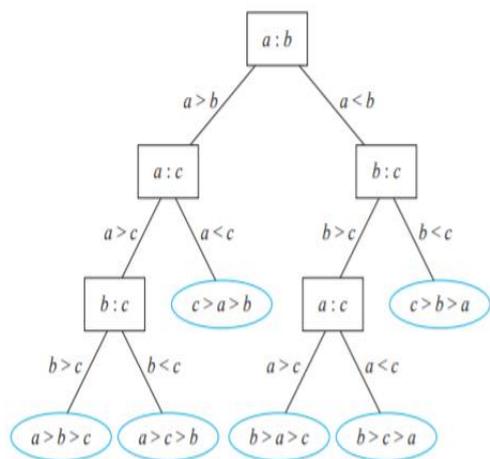
Pohon berakar dapat digunakan untuk memodelkan sebuah masalah yang akan membimbing menuju sebuah keputusan hingga mendapatkan solusi dari permasalahan tersebut. Pohon berakar untuk setiap simpulnya berkorespondensi dengan keputusan, dengan upapohon / *subtree* dari simpul ini merupakan setiap kemungkinan yang muncul dalam pengambilan keputusan. Solusi yang mungkin dari sebuah permasalahan berkorenspondensi dengan lintasan menuju sebuah daun.



Gambar 6 : Pohon keputusan untuk menemukan koin palsu.

Sumber : Rosen, Kenneth H.. 2012. Discrete Mathematics and Its Application. 7th Ed. New York: McGraw-Hill.

Pada gambar 6 di atas merupakan pohon keputusan untuk menentukan dari delapan buah koin yang di antaranya ada satu koin palsu. Koin palsu tersebut memiliki berat yang lebih ringan dibandingkan yang lain.



Gambar 7 : Pohon keputusan untuk menentukan urutan bilangan dari yang terbesar ke terkecil
 Sumber : Rosen, Kenneth H.. 2012. Discrete Mathematics and Its Application. 7th Ed. New York: McGraw-Hill.

Pada gambar 7, pohon keputusan untuk menentukan urutan bilangan dari yang terbesar ke terkecil dari tiga buah bilangan yang berbeda. Pada pohon keputusan ini, kemungkinan paling cepat menemukan solusinya adalah dengan dua langkah dan yang paling lama adalah dengan menggunakan empat langkah.

Pohon keputusan merupakan salah satu cara untuk menggambarkan algoritma yang mempunyai *conditional*. Penggunaan pohon keputusan memiliki kelebihan yaitu sederhana dan mudah dipahami namun memiliki kekurangan yaitu tidak terlalu akurat dalam kondisi-kondisi tertentu.

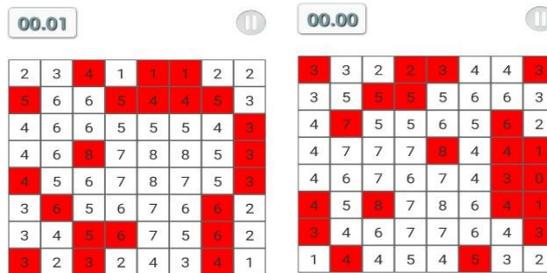
III. STRATEGI DAN ANALISIS DALAM PERMAINAN HIDDEN PANEL

A. Kemunculan Cross

Pada permainan *hidden panel* terdapat tiga tingkat kesulitan yaitu *easy*, *medium*, dan *hard*. Tiap tingkat kesulitan memiliki perbedaan jumlah kotak, banyaknya *cross*, dan banyaknya *green*. Penulis melakukan pengujian apakah banyaknya *cross* dalam tingkat kesulitan yang sama akan berbeda atau tidak. Data pengujian dapat dilihat pada gambar di bawah ini namun untuk mengemat pemakaian ruang, hanya ditampilkan dua buah pengujian untuk tiap tingkat kesulitan.

Tingkat kesulitan :

a) Easy



Gambar 8 : Data uji coba banyaknya *cross* pada tingkat *easy*
 Sumber : dokumen penulis
 Diperoleh bahwa pada tingkat kesulitan *easy* terdapat 21 *cross* dari total kotak sebanyak 64 kotak.

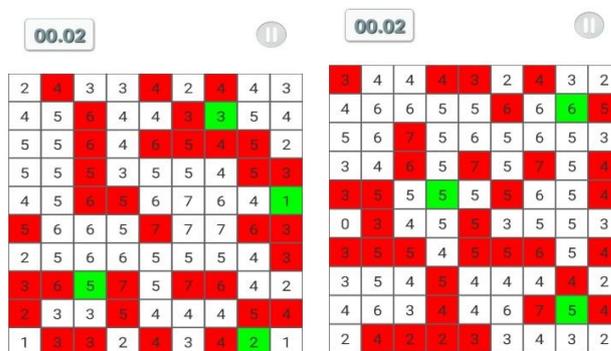
b) Medium



Gambar 9 : Data uji coba banyaknya *cross* pada tingkat *medium*

Sumber : dokumen penulis
 Diperoleh bahwa pada tingkat kesulitan *easy* terdapat 28 *cross* dari total kotak sebanyak 72 kotak.

c) Hard



Gambar 10 : Data uji coba banyaknya *cross* pada tingkat *hard*
 Sumber : dokumen penulis

Diperoleh bahwa pada tingkat kesulitan *easy* terdapat 33 *cross* dari total kotak sebanyak 90 kotak.

	Easy	Medium	Hard
Total Kotak	64	72	90
Frekuensi <i>cross</i>	21	28	33
Persentase banyaknya <i>cross</i>	32,8125 %	38,8888 %	36,6667 %

B. Pola – pola dalam *hidden panel*

Pada awal permainan diperlukan langkah awal untuk mulai membukak kotak – kotak yang belum diketahui. Dengan awalan yang baik akan membuat kotak – kotak semakin banyak yang terbuka dan kita dapat mengetahui letak *cross*.

A. Pola 8

Ketika kotak memiliki angka 8 maka dapat dipastikan semua kotak di sekitar kotak tersebut berisi *green*.

4	5	3
7	8	5
6	7	4

Gambar 11 : Contoh pola 8
Sumber : dokumen penulis

B. Pola 5

Ketika kotak yang memiliki angka 5 pada tepi kotak maka dapat dipastikan semua kotak di sekitar kotak tersebut berisi *green*.

2	5	4	4
3	6	6	6
4	7	5	4

Gambar 12 : Contoh pola 5
Sumber : dokumen penulis

C. Pola 3

Ketika kotak yang memiliki angka 5 pada pojok *area* permainan maka dapat dipastikan semua kotak di sekitar kotak tersebut berisi *green*.

3	5	4
4	5	4
3	4	5

Gambar 13 : Contoh pola 3
Sumber : dokumen penulis

D. Pola 0

Ketika kotak memiliki angka 0 maka dapat dipastikan semua kotak di sekitar kotak tersebut adalah *cross*.

4	4	4	2
5	4	2	2
3	3	0	1

Gambar 14 : Contoh pola 0
Sumber : dokumen penulis

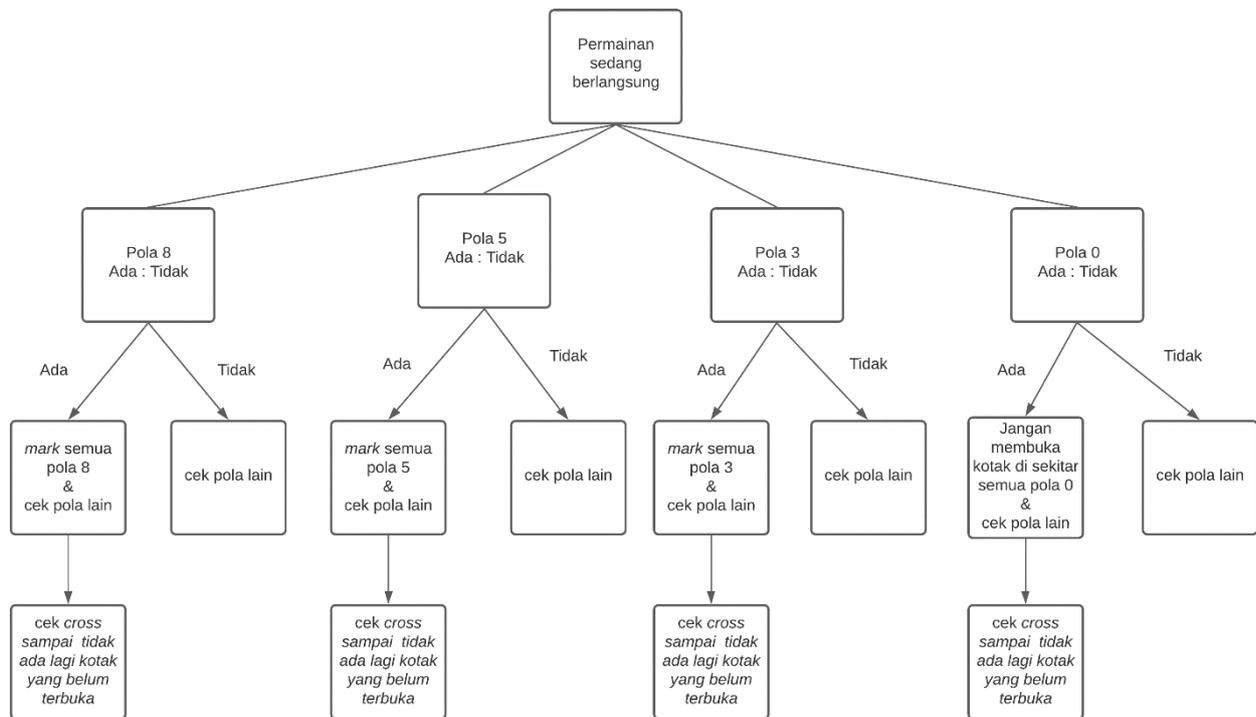
IV. PEMBAHASAN

A. Pohon Keputusan dalam Permainan *Hidden Panel*

Dengan mengetahui persentase kemunculan *cross* dan mengetahui pola-pola yang menjadi panduan terutama pada awal permainan. Setelah membuka kotak-kotak dengan pola yang sudah dijelaskan pada BAB 3. Dengan demikian, kotak yang berisi *cross* sudah mulai bermunculan sehingga dengan mengetahui *cross* yang ada pada *area* permainan maka kita dapat mulai menganalisis kotak-kotak yang berada dekat *cross*. Dapat diambil contoh pada gambar 14,

4	4	4	2
5	4	2	2
3	3	0	1

fokus pada kotak yang dilingkari maka dapat diperhatikan bahwa kotak 3 dan kotak 5 adalah *green* karena di sekeliling kotak tersebut ada 6 kotak yang sudah diketahui 2 *cross* dan 1 *green*. Sehingga dapat dipastikan bahwa kotak 3 dan 5 adalah *green*. Teknik ini penulis beri nama “cek *cross*”.



Gambar : Pohon Keputusan permainan *hidden panel*
 Sumber : dokumen penulis

Pada pohon keputusan di atas, hal pertama yang dilakukan adalah cek pola-pola seperti yang sudah dijelaskan pada BAB 3. Setelah membuka kotak-kotak dengan pola tersebut maka dilakukan teknik cek *cross* hingga tidak ada lagi kotak yang belum terbuka. Ketika pada suatu posisi, pemain mengalami *stuck* maka dapat mencoba membuka kotak yang belum pasti. Akan tetapi, pemilihan kotak yang akan dibuka hendaklah diperhitungkan dengan baik agar kemungkinan mendapat *green* lebih besar. Strategi yang dapat diterapkan adalah membuka kotak yang memiliki pusat dengan angka yang sebesar mungkin karena dengan angka pada kota pusat semakin besar maka kemungkinan *green* muncul lebih besar. Setelah itu, dapat dilakukan cek lagi dengan menggunakan teknik cek *cross*.

V. KONKLUSI

Permainan *hidden panel* dapat dimodelkan dengan pohon keputusan untuk memenangkan permainan. Dengan membuat pohon keputusan tersebut, kita dapat menentukan tindakan selanjutnya yang akan kita lakukan. Selain itu, perlu dilakukan cek ulang / *looping* dalam menemukan pola serta melanjutkan kotak-kotak yang sudah mulai terbuka dengan teknik cek *cross*. Sehingga, kita dapat menyelesaikan permainan *hidden panel* semakin cepat dengan mengetahui pola – pola kunci pada permainan tersebut.

VI. UCAPAN TERIMA KASIH

Penulis bersyukur kepada Allah SWT, yang telah melimpahkan rahmaat-Nya sehingga penulis dapat menyelesaikan makalah ini. Penulis mengucapkan terima kasih kepada seluruh Bapak dan Ibu dosen pengampu mata kuliah IF2120 Matematika Diskrit atas segala ilmu dan bimbingannya. Penulis juga ingin menyampaikan terima kasih kepada kedua orang tua yang selalu memberikan dukungan kepada penulis serta kepada teman – teman yang telah memberikan bantuan dan dukungan dalam proses penyusunan makalah ini.

REFERENCES

- [1] Munir, Rinaldi, 2020. Pohon Bagian I dan II. Bandung. <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf>, <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>.
- [2] Rosen, Kenneth H. 2012. Discrete Mathematics and Its Application. 7th Ed. New York: McGraw-Hill.
- [3] Ekananda, Faris Rizki. 2019. Penerapan Pohon Keputusan (Decision Tree) dalam Permainan Minesweeper. Paper

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.

Payakumbuh, 11 Desember 2020

Muhammad Fikri. N 13519069