

SKRIPSI

**PENYELESAIAN PERMAINAN MASYU (PEARL PUZZLE)
DENGAN ALGORITMA KETETANGGAAN DAN
DEPTH-FIRST SEARCH (DFS)**



Joseph Febryano Nathan

NPM: 2016730007

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2022**

UNDERGRADUATE THESIS

**MASYU GAME SOLUTION (PEARL PUZZLE) WITH
ADJACENCY ALGORITHM AND DEPTH-FIRST SEARCH
(DFS)**



Joseph Febryano Nathan

NPM: 2016730007

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2022**

LEMBAR PENGESAHAN

**PENYELESAIAN PERMAINAN MASYU (PEARL PUZZLE)
DENGAN ALGORITMA KETETANGGAAN DAN
DEPTH-FIRST SEARCH (DFS)**

Joseph Febryano Nathan

NPM: 2016730007

Bandung, 14 Januari 2022

Menyetujui,

Pembimbing

**Digitally signed
by Cecilia Esti
Nugraheni**

Dr.rer.nat. Cecilia Esti Nugraheni

**Ketua Tim Penguji
Digitally signed
by Luciana
Abednego**

Luciana Abednego, M.T.

**Anggota Tim Penguji
Digitally signed
by Mariskha Tri
Adithia**

Mariskha Tri Adithia, P.D.Eng

Mengetahui,

**Ketua Program Studi
Digitally signed
by Mariskha Tri
Adithia**

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PENYELESAIAN PERMAINAN MASYU (PEARL PUZZLE) DENGAN ALGORITMA KETETANGGAAN DAN DEPTH-FIRST SEARCH (DFS)

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 14 Januari 2022



Joseph Febryano Nathan
NPM: 2016730007

ABSTRAK

Masyu merupakan salah satu puzzle yang diterbitkan oleh Nikoli Co.Ltd. Nikoli Co.Ltd. Masyu atau Pearl Puzzle merupakan sebuah permainan puzzle yang dimainkan dalam sebuah papan yang berbentuk persegi atau persegi panjang, beberapa persegi di dalam papan permainan akan diisi dengan lingkaran; setiap lingkaran yang ada memiliki warna antara “putih” (kosong) atau “hitam” (terisi). Tujuan dari permainan Masyu adalah untuk membuat satu garis tidak berpotongan dan melewati semua lingkaran.

Pada skripsi ini akan dibuat sebuah perangkat lunak untuk penyelesaian permainan Masyu yang berbasis web. Perangkat lunak yang dibuat merupakan pencari solusi atau *solver* permainan Masyu. Perangkat lunak akan dibuat dengan bantuan React.js. React (React.js atau ReactJS) adalah *open-source, front-end, JavaScript library* untuk membangun antarmuka pengguna atau komponen UI. Dari berbagai macam teknik yang dapat digunakan untuk menyelesaikan permainan Masyu, dipilih algoritma ketetangaan dan DFS (*Depth-first search*).

Algoritma ketetangaan titik adalah algoritma untuk mendapatkan nilai suatu titik yang melibatkan nilai titik-titik tetangganya dan DFS (*Depth-first search*) adalah metode pencarian yang dimulai dengan node awal grafik dan kemudian berjalan lebih dalam dan lebih dalam hingga menemukan node yang diperlukan atau node yang tidak memiliki anak. Algoritma ketetangaan dapat memberikan solusi untuk permainan Masyu tanpa bantuan DFS (*Depth-first search*) untuk soal soal tertentu, namun jika belum ditemukan solusi maka akan dilanjutkan oleh DFS. Algoritma ketetangaan akan berperan sebagai initial state dalam proses pencarian DFS (*Depth-first search*) yang akan menelusuri seluruh kemungkinan, sehingga solusi dari permainan Masyu akan lebih cepat ditemukan karena kondisi awal pencarian tidak kosong.

Hasil evaluasi dari perangkat lunak adalah permainan Masyu dengan algoritma ketetangaan dapat digunakan untuk menyelesaikan beberapa permainan Masyu tanpa bantuan DFS. Pemakaian algoritma ketetangaan dan DFS pada permainan Masyu untuk ukuran papan 6×6 , 8×8 , dan 10×10 memiliki persentase pemakaian yang hampir sama, dan jumlah titik pada permainan Masyu, jika semakin banyak belum tentu akan mempercepat waktu pencarian solusi pada permainan Masyu.

Kata-kata kunci: Masyu, *puzzle*, ketetangaan, DFS, React.js, solusi, permainan

ABSTRACT

Masyu is one of the puzzles published by Nikoli Co.Ltd. Nikoli Co.Ltd. Masyu or Pearl Puzzle is a puzzle game that is played on a square or rectangular board, some squares on the game board will be filled with circles; each existing circle has a color between “white” (empty) or “black” (filled). The goal of Masyu game is to make one non-intersecting line and pass through all the circles.

In this thesis, a web-based software for the completion of the Masyu game will be created. The software created is a solution finder or solver Masyu game. The software will be built with the help of React.js. React (React.js or ReactJS) is a open-source, front-end, JavaScript library for building user interfaces or UI components. From the various techniques that can be used to solve Masyu game, the neighboring algorithm and DFS(Depth-first search) are chosen.

The point-neighborhood algorithm is an algorithm to get the value of a point involving the values of its neighboring points and DFS(Depth-first search) is a search method that starts with the initial node of the graph and then goes deeper and deeper until it finds the node that required or nodes that have no children. The neighboring algorithm can provide solutions for Masyu game without the help of DFS(Depth-first search) for certain questions, but if a solution has not been found, it will be continued by DFS. The neighboring algorithm will act as the initial state in the DFS search process (Depth-first search) which will explore all possibilities, so that the solution from Masyu game will be found faster because the initial search conditions are not empty.

The evaluation result of the software is that the Masyu game with the neighboring algorithm can be used to solve several Masyu games without the help of DFS. The use of the neighboring algorithm and DFS in the Masyu game for board sizes 6×6 , 8×8 , and 10×10 have almost the same percentage of usage, and the number of points in Masyu game, if there are more it will not necessarily speed up the time to find solutions in Masyu game.

Keywords: Masyu, puzzle, adversity, DFS, React.js, solution, game

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul “Penyelesaian Permainan Masyu (Pearl Puzzle) dengan Algoritma Ketetangaan dan Depth-first search (DFS)”.

Skripsi ini dibuat dan diajukan untuk memenuhi syarat guna memperoleh gelar Sarjana Komputer pada Fakultas Teknologi Informasai dan Sains di Universitas Katolik Parahyangan. Selain itu, tujuan dari penulisan skripsi ini adalah untuk memberikan pengetahuan kepada pembaca mengenai Penyelesaian Permainan Masyu (Pearl Puzzle) dengan Algoritma Ketetangaan dan Depth-first search (DFS).

Selama penulisan skripsi ini, penulis banyak menerima bantuan dan dukungan sehingga dapat menyelesaikan skripsi ini. Oleh karena itu, penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Keluarga penulis, yang senantiasa mendampingi serta memotivasi penulis untuk menyelesaikan skripsi ini.
2. Dr.rer.nat.Cecilia Esti Nugraheni yang selalu membimbing penulis dalam menyusun skripsi ini.
3. Teman-teman khususnya Kevin Renaldy Halim, Febrian Nathan Mintareja, Gerry Lesmana, Cahyadi Hartanto, Gunawan Christanto, Cindia Winarta yang pernah bekerja dengan penulis serta membantu penulis.
4. Teman-teman dekat penulis yang selalu memberikan dukungan moral.
5. Rekan-rekan informatika angkatan 2016 yang tidak dapat disebutkan satu-persatu.
6. Pihak-pihak lain yang senantiasa mendoakan penulis.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna karena adanya keterbatasan ilmu dan pengalaman yang dimiliki. Oleh karena itu, semua kritik dan saran yang bersifat membangun akan penulis terima dengan senang hati. Penulis berharap, semoga skripsi ini dapat bermanfaat bagi semua pihak yang memerlukan.

Bandung, Januari 2022

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	3
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Permainan Masyu	5
2.1.1 Aturan Permainan Masyu	5
2.1.2 Heuristik Permainan Masyu	6
2.1.3 Situasi Khusus Pada Permainan Masyu	8
2.2 State space	10
2.3 Algoritma	10
2.4 Teknik Pencarian	12
2.5 Himpunan	13
2.6 JavaScript	13
2.7 npm	14
2.8 Node.js	14
2.9 ReactJS	14
3 ANALISIS PERANGKAT LUNAK	17
3.1 Pemodelan Masalah	17
3.1.1 Pemodelan Papan dan Petak	18
3.1.2 Pemodelan Algoritma Ketetangaan	19
3.1.3 Pemodelan DFS	23
3.2 Analisis Masukan	25
3.3 Analisis Keluaran	26
3.4 Analisis Penyelesaian Masalah dengan algoritma ketetangaan dan DFS	27
3.4.1 <i>Filtering</i> Bentuk Tabel Solusi	27
3.4.2 Flow Diagram Penyelesaian Permainan Masyu	29
3.4.3 Use Case Diagram Penyelesaian Permainan Masyu	31
3.4.4 Kelas Diagram Permainan Masyu	32
3.4.5 Penyelesaian Masyu dengan Algoritma	34
4 PERANCANGAN	53

4.1	Perancangan Kelas - Kelas Pada Perangkat Lunak	53
4.1.1	Kelas Game	54
4.1.2	Kelas Board	55
4.1.3	Kelas Square	56
4.1.4	Kelas Solver	57
4.2	Perancangan algoritma ketetangaan untuk permainan Masyu	62
4.3	Perancangan DFS untuk permainan Masyu	68
4.4	Perancangan Tampilan Perangkat Lunak	71
4.4.1	Flow Penggunaan Perangkat Lunak	71
4.4.2	Perancangan Halaman Perangkat Lunak	72
5	IMPLEMENTASI DAN PENGUJIAN	75
5.1	Implementasi	75
5.1.1	Implementasi Framework React.js	75
5.1.2	Implementasi Kelas - Kelas Pada React.js	77
5.1.3	Implementasi Algoritma Ketetangaan	80
5.1.4	Implementasi DFS	84
5.1.5	Implementasi Tampilan Perangkat Lunak	87
5.2	Pengujian	88
5.2.1	Pengujian Fungsionalitas	88
5.2.2	Pengujian Performa Algoritma Ketetangaan dan DFS	100
5.2.3	Kesimpulan Pengujian Fungsional dan Pengujian Peforma	107
6	KESIMPULAN DAN SARAN	109
6.1	Kesimpulan	109
6.2	Saran	109
	DAFTAR REFERENSI	111
	A KODE PROGRAM	113
	B HASIL EKSPERIMEN	139

DAFTAR GAMBAR

1.1	Permainan Masyu ketika awal permainan dan penyelesaiannya	1
2.1	Papan permainan Masyu ketika awal permainan dan penyelesaiannya	5
2.2	Kemungkinan gerakan garis hitam	6
2.3	Kemungkinan gerakan garis putih	6
2.4	Titik hitam dekat tepi papan	6
2.5	Titik putih ditepi papan	7
2.6	Titik hitam yang bersebelahan	7
2.7	Titik putih yang bersebelahan	8
2.8	Titik hitam diantara titik putih	8
2.9	Garis yang bercabang	8
2.10	<i>Line loops</i>	9
2.11	<i>Dead line</i>	9
2.12	Garis yang membuat <i>error</i> titik lain	10
2.13	Representasi Bentuk	10
2.14	Ilustrasi bentuk 5 yang mempengaruhi nilai disekitarnya	11
2.15	Ilustrasi nilai yang dihasilkan dari ketetanggaan	11
2.16	Ilustrasi pemetaan posisi 4-ketetanggaan dan 8-ketetanggaan	11
2.17	Teknik pencarian Depth-first search	12
3.1	Pemodelan Tabel Pearl	18
3.2	Pemodelan Bentuk	19
3.3	Pemodelan Tabel Solusi	19
3.4	4-Ketetanggaan	20
3.5	Algoritma ketetanggaan untuk Bentuk 0 (kosong)	20
3.6	Algoritma ketetanggaan untuk Bentuk 1 (atas kanan)	21
3.7	Algoritma ketetanggaan untuk Bentuk 2 (bawah kanan)	21
3.8	Algoritma ketetanggaan untuk Bentuk 3 (bawah kiri)	22
3.9	Algoritma ketetanggaan untuk Bentuk 4 (atas kiri)	22
3.10	Algoritma ketetanggaan untuk Bentuk 5 (garis horizontal)	23
3.11	Algoritma ketetanggaan untuk Bentuk 6 (garis vertikal)	23
3.12	Contoh Hasil Algoritma Ketetanggaan	24
3.13	DFS pada Masyu	25
3.14	Contoh masukan perangkat lunak	26
3.15	Hasil tabel solusi	26
3.16	Hasil tabel solusi yang digambar	27
3.17	Proses filtering tepi papan	28
3.18	Hasil filtering tepi papan	28
3.19	Proses filtering berdasarkan tipe lingkaran	29
3.20	Hasil filtering berdasarkan tipe lingkaran	29
3.21	Flowchart permainan Masyu	31
3.22	Use Case permainan Masyu	32
3.23	Kelas Diagram permainan Masyu	33

3.24	Langkah permulaan	34
3.25	Langkah <i>filtering</i> tepi	35
3.26	Langkah <i>filtering</i> lingkaran	35
3.27	Langkah ketetangaan 1	36
3.28	Langkah ketetangaan 2	37
3.29	Langkah ketetangaan 3	37
3.30	Langkah ketetangaan 4	38
3.31	Langkah ketetangaan 5	39
3.32	Langkah ketetangaan 6	39
3.33	Langkah ketetangaan 7	40
3.34	Langkah ketetangaan 8	41
3.35	Langkah ketetangaan 9	41
3.36	Langkah ketetangaan 10	42
3.37	Langkah ketetangaan 11	43
3.38	Langkah ketetangaan 12	43
3.39	Langkah ketetangaan 13	44
3.40	Langkah ketetangaan 14	45
3.41	Langkah ketetangaan 15	45
3.42	Langkah ketetangaan 16	46
3.43	Langkah ketetangaan 17	47
3.44	Langkah ketetangaan 18	47
3.45	Langkah ketetangaan 19	48
3.46	Langkah ketetangaan 20	49
3.47	Langkah DFS 1	49
3.48	Langkah DFS 1A ketetangaan	50
3.49	Langkah DFS 2	51
3.50	Langkah DFS 2A	51
3.51	Langkah DFS 2B	52
3.52	Hasil permainan Masyu	52
4.1	Diagram kelas permainan Masyu	54
4.2	Diagram kelas Game	54
4.3	Diagram kelas Board	55
4.4	Diagram kelas Square	56
4.5	Diagram kelas Solver	57
4.6	Flowchart tampilan permainan Masyu	72
4.7	Halaman utama permainan Masyu	73
5.1	Verifikasi npm dan node.js	75
5.2	Struktur folder React.js	76
5.3	Import kelas pada React.js	76
5.4	Constructor pada React.js	77
5.5	Implementasi Tampilan Permainan Masyu	88
5.6	Pengguna memilih tombol titik hitam	89
5.7	Hasil pengisian papan dengan titik hitam	89
5.8	Pengguna memilih tombol titik putih	90
5.9	Hasil pengisian papan dengan titik putih	90
5.10	Pengguna memilih tombol titik putih	91
5.11	Hasil penghapusan titik putih	91
5.12	Pengguna menghapus isi papan dengan tombol clear all	92
5.13	Pengguna memilih ukuran papan 7×7	92
5.14	Hasil dari pemilihan ukuran papan 7×7	93

5.15	Pengguna memilih ukuran papan 8×8	93
5.16	Hasil dari pemilihan ukuran papan 8×8	94
5.17	Pengguna memilih tombol solve untuk menyelesaikan kasus	94
5.18	Hasil pemilihan tombol solve	95
5.19	Contoh hasil kasus dengan 1 titik hitam	96
5.20	Contoh hasil kasus dengan 1 titik putih	96
5.21	Contoh hasil kasus dengan 2 titik hitam	97
5.22	Contoh hasil kasus dengan 2 titik putih	97
5.23	Contoh hasil kasus dengan 1 titik hitam dan 1 titik putih	98
5.24	Contoh hasil kasus dengan 3 titik hitam	98
5.25	Contoh hasil kasus dengan 3 titik putih	99
5.26	Contoh hasil kasus dengan 2 titik hitam dan 1 titik putih	99
5.27	Contoh hasil kasus dengan 1 titik hitam dan 2 titik putih	100
5.28	Contoh pemasukan kasus 6×6	102
5.29	Perbandingan hasilnya dengan kunci jawaban untuk papan 6×6	102
5.30	Contoh pemasukan kasus 8×8	103
5.31	Perbandingan hasilnya dengan kunci jawaban untuk papan 8×8	103
5.32	Contoh pemasukan kasus 10×10	104
5.33	Perbandingan hasilnya dengan kunci jawaban untuk papan 10×10	105

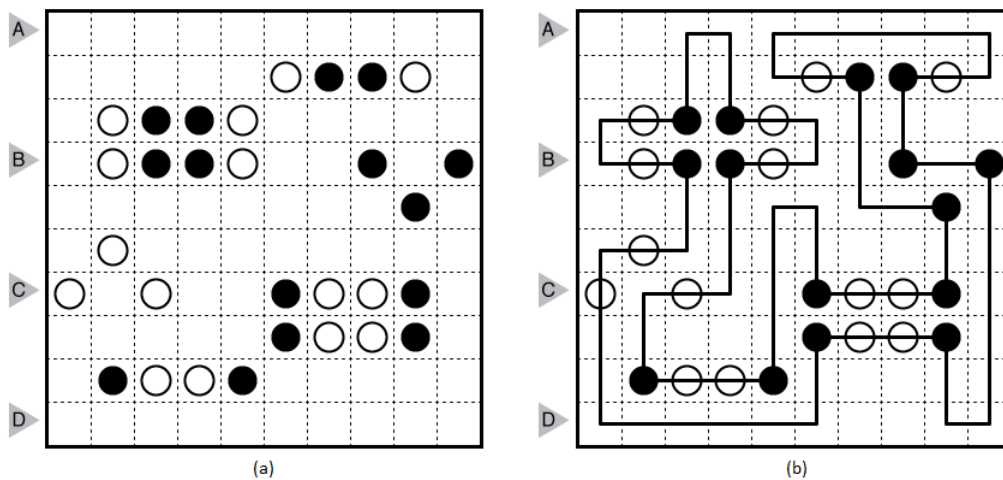
BAB 1

PENDAHULUAN

1.1 Latar Belakang

Masyu (“evil influence”) merupakan salah satu puzzle yang diterbitkan oleh Nikoli Co.Ltd. Nikoli Co.Ltd merupakan perusahaan yang menerbitkan puzzle sejak 1980. Versi awal Masyu pertama kali muncul di Puzzle Communication Nikoli 84 dengan judul Shinju no Kubikazari ("kalung mutiara"). Puzzle itu hanya berisi lingkaran putih. Lingkaran hitam diperkenalkan di Puzzle Communication Nikoli 90, dan puzzle tersebut diubah namanya menjadi Shiroshinju Kuroshinju ("mutiara putih dan mutiara hitam"), kemudian pada akhirnya berubah nama menjadi Pearl Puzzle.

Masyu atau Pearl Puzzle merupakan sebuah permainan puzzle yang dimainkan dalam sebuah papan yang berbentuk persegi atau persegi panjang seperti yang tampak pada Gambar 1.1(a), beberapa persegi di dalam papan permainan akan diisi dengan lingkaran; setiap lingkaran yang ada memiliki warna antara “putih” (kosong) atau “hitam” (terisi). Memahami posisi lingkaran dan bagaimana mereka berinteraksi satu sama lain adalah kunci untuk memecahkan permainan Masyu. Tujuan dari permainan Masyu adalah untuk membuat satu garis tidak berpotongan dan melewati semua lingkaran tampak pada Gambar 1.1(b).



Gambar 1.1: Permainan Masyu ketika awal permainan dan penyelesaiannya

Aturan permainan Masyu adalah:

- Semua titik hitam dan titik putih hanya dilewati satu garis.
- Garis yang melewati titik hitam harus langsung berbelok, tetapi tidak dapat langsung berbelok lagi; semua belokan 90 derajat.
- Garis harus melewati titik putih, dan setelah melewati titik putih harus berbelok baik ke kiri maupun kekanan titik putih yang dilewatinya.
- Garis tidak boleh bercabang.

Pada skripsi ini dibuat sebuah perangkat lunak untuk permainan Masyu yang berbasis web. Perangkat lunak akan dibuat dengan bantuan React.js. React (React.js atau ReactJS) adalah *open-source, front-end*, JavaScript library untuk membangun antarmuka pengguna atau komponen UI. React dikelola oleh Facebook, komunitas, dan perusahaan. React digunakan karena React memungkinkan pengembang membuat aplikasi web besar yang dapat mengubah data, tanpa memuat ulang halaman. Tujuan utama React adalah menjadi cepat, terukur, dan sederhana. Ini hanya berfungsi pada antarmuka pengguna dalam aplikasi. Ini dapat digunakan dengan kombinasi pustaka atau kerangka kerja JavaScript lainnya, seperti Angular JS.

Pada skripsi ini, algoritma yang akan digunakan untuk membantu menyelesaikan permainan Masyu adalah algoritma ketetangaan dan *Depth-first search*(DFS). Algoritma ketetangaan dapat memberikan solusi untuk permainan Masyu tanpa bantuan DFS untuk soal soal tertentu, namun jika belum ditemukan solusi maka akan dilanjutkan oleh DFS. Algoritma ketetangaan akan berperan sebagai initial state dalam proses pencarian DFS yang akan menelusuri seluruh kemungkinan. Sehingga solusi dari permainan Masyu akan lebih cepat ditemukan karena kondisi awal pencarian tidak kosong.

Pada skripsi ini, dibuat sebuah perangkat lunak permainan Masyu berbasis web yang dapat digunakan untuk mencari jawaban dari soal yang dimasukkan dan juga dibuat solver permainan Masyu yang menerapkan algoritma ketetangaan dan *Depth-first search*(DFS). Soal permainan Masyu akan diambil dari sebuah website yang bernama krazydad [1]. Skripsi ini akan diberi judul "Penyelesaian Permainan Masyu (Pearl Puzzle) dengan Algoritma Ketetangaan dan *Depth-first search*(DFS)".

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana cara kerja algoritma ketetangaan dan *Depth-first search*(DFS)?
2. Bagaimana cara mengaplikasikan algoritma ketetangaan dan *Depth-first search*(DFS) pada penyelesaian permainan Masyu?
3. Bagaimana mengembangkan perangkat lunak penyelesaian permainan Masyu dengan menggunakan algoritma ketetangaan *Depth-first search*(DFS)?
4. Bagaimana kinerja algoritma ketetangaan dan *Depth-first search*(DFS) pada penyelesaian permainan Masyu?

1.3 Tujuan

Berdasarkan Rumusan masalah yang telah dipaparkan, tujuan penelitian ini adalah sebagai berikut:

1. Mempelajari cara kerja algoritma ketetangaan dan *Depth-first search*(DFS).
2. Mempelajari cara mengaplikasikan algoritma ketetangaan dan *Depth-first search*(DFS) pada penyelesaian permainan Masyu.
3. Mengembangkan perangkat lunak penyelesaian permainan Masyu dengan menggunakan algoritma ketetangaan dan *Depth-first search*(DFS).
4. Menganalisis kinerja algoritma ketetangaan dan *Depth-first search*(DFS) pada penyelesaian permainan Masyu.

1.4 Batasan Masalah

Untuk mempermudah pembuatan skripsi ini, ada hal - hal yang harus dibatasi, diantaranya:

1. Permainan Masyu minimal adalah 5×5 dan maksimal adalah 10×10 .
2. Jumlah lingkaran yang ada pada permainan Masyu minimal 1 dan maksimal 50% dari ukuran soal, Masyu 5×5 memiliki minimal satu lingkaran dan maksimal 25 lingkaran.

3. Perangkat lunak dibangun pada web dan bahasa pemrograman yang digunakan adalah javascript, html, dan framework React.js.
4. Pengujian perangkat lunak akan menggunakan soal - soal yang dapat diselesaikan atau memiliki solusi yang benar.

1.5 Metodologi

Metodologi yang diterapkan pada penulisan skripsi ini adalah :

1. Tahap Pengumpulan data. Tahap ini adalah tahap untuk mengumpulkan data dari buku-buku, karya ilmiah, dan sumber-sumber dari internet yang dapat berguna dan berhubungan dengan penulisan skripsi ini.
2. Tahap perancangan. Tahap ini adalah tahap untuk merancang kelas, algoritma dan tampilan.
3. Tahap pembuatan perangkat lunak. Tahap ini adalah tahap untuk membuat perangkat lunak menggunakan framework react.js .
4. Tahap pengujian perangkat lunak. Tahap ini adalah tahap untuk menguji algoritma ketetangaan dan *Depth-first search*(DFS) pada perangkat lunak.
5. Tahap penulisan dokumen.

1.6 Sistematika Pembahasan

Sistematika pembahasan pada penelitian ini adalah sebagai berikut :

1. Bab 1 Pendahuluan
Bab 1 berisi tentang latar belakang masalah, rumusan masalah, tujuan penelitian, batasan masalah, dan metodologi yang digunakan pada penelitian ini.
2. Bab 2 Landasan Teori
Bab 2 berisi dasar - dasar teori dari metode algoritma ketetangaan, *Depth-first search*(DFS), dan React.js.
3. Bab 3 Analisis
Bab 3 berisi analisis masalah dan solusi yang digunakan untuk menyelesaikan masalah tersebut, analisis tentang latar belakang masalah dan implementasi perangkat lunak.
4. Bab 4 Perancangan
Bab 4 berisi penjelasan yang meliputi perancangan kelas - kelas pada perangkat lunak, metode algoritma ketetangaan dan *Depth-first search*(DFS), dan tampilan perangkat lunak.
5. Bab 5 Implementasi dan Pengujian
Bab 5 berisi penjelasan yang meliputi hasil implementasi tampilan, kode kode penting dan pengujian metode algoritma ketetangaan dan *Depth-first search*(DFS) pada perangkat lunak.
6. Bab 6 Kesimpulan
Bab 6 berisi kesimpulan dari keseluruhan implementasi dan pengujian yang dilakukan pada penelitian ini.