

SKRIPSI

**MENYELESAIKAN PERMAINAN DOMINO GRID
MENGUNAKAN ALGORITMA HUMAN DAN ODOMETER**



DANIEL

NPM: 2011730102

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2017**

UNDERGRADUATE THESIS

**SOLVING DOMINO GRID USING HUMAN ALGORITHM
AND ODOMETER**



DANIEL

NPM: 2011730102

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND
SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2017**

LEMBAR PENGESAHAN

**MENYELESAIKAN PERMAINAN DOMINO GRID
MENGUNAKAN ALGORITMA HUMAN DAN ODOMETER**

DANIEL

NPM: 2011730102

Bandung, 10 Mei 2017

Menyetujui,

Pembimbing



Dr.rer.nat. Cecilia Esti Nugraheni



Ketua Tim Penguji



Dott. Thomas Anung Basuki

Anggota Tim Penguji



Aditya Bagoes Saputra, M.T.

Mengetahui,

Ketua Program Studi



Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

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

MENYELESAIKAN PERMAINAN DOMINO GRID MENGGUNAKAN ALGORITMA HUMAN DAN ODOMETER

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 10 Mei 2017



Daniel
NPM: 2011730102

ABSTRAK

Kenneth E. Caviness seorang ahli matematika mengajukan masalah *domino grid*. *Domino Grid* sendiri adalah permainan tentang bagaimana cara untuk menutupi papan permainan dengan seluruh kartu domino yang tersedia. Permainan *Domino Grid* ini terdiri dari papan permainan dan *bone*. Papan permainan adalah umumnya memiliki ukuran 7x8 kotak yang berisi dengan angka-angka mulai dari 0 sampai dengan 6. Ada beberapa algoritma yang dapat digunakan untuk menyelesaikan *domino grid* ini. Algoritma yang akan digunakan adalah algoritma *human* dan odometer. Pada skripsi ini, bahasa pemrograman yang akan digunakan adalah java. Perangkat lunak yang dibuat akan mengimplementasi algoritma *human* dan odometer, dimana perangkat lunak akan dapat menampilkan solusi dari permainan ini. Untuk ukuran papan yang cukup besar (7x8) algoritma *human* dapat mendapatkan solusi permainan jauh lebih cepat daripada odometer. Sebaliknya untuk papan yang kecil (3x4) waktu yang dibutuhkan kedua algoritma hampir sama.

Kata-kata kunci: Domino, papan permainan, algoritma human, odometer

ABSTRACT

Kenneth E. Caviness is a mathematician who first encountered domino grid problem. Domino grid is a game about how to cover the puzzle with all the dominoes tile. Domino grid consist of a board and bone tile. The board is a 7x8 square that contains number between 0 and 6. There are several algorithm to solve this puzzle. Algorithm that will be implemented are human algorithm and odometer. Also, in this study, the programming language that used to implement those two algorithms will be java. The software that will be made will implement human algorithm and odometer, where the software will be able show the solution of the board. For big board (ex. 7x8) human algorithm solved the game much faster than odometer. For small board (ex. 3x4) both algorithm solved the game at almost same time.

Keywords: Domino, bone, board, human algorithm, odometer

Dipersembahkan untuk diri sendiri

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas segala bantuan-Nya penulis dapat menyelesaikan tugas akhir yang berjudul "Menyelesaikan Permainan *Domino Grid* Menggunakan Algoritma *Human* dan Odometer". Setelah menerima bantuan dari berbagai pihak, akhirnya tugas akhir ini dapat diselesaikan. Oleh sebab itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada seluruh pihak yang telah berperan pada pembuatan tugas akhir ini. Secara khusus penulis ingin menyampaikan terima kasih kepada:

1. Seluruh keluarga yang memberikan dukungan.
2. Bu Cecilia Esti Nugraheni selaku pembimbing yang telah memberikan banyak bantuan.
3. Seluruh dosen lain yang telah berperan selama penulis kuliah di Informatika UNPAR.
4. Teman-teman yang telah memberikan semangat
5. Rekan-rekan seangkatan di Informatika UNPAR yang telah berbagi ilmu dan pengalaman.

Akhir kata, penulis menyadari bahwa tugas akhir ini tidak lepas dari kekurangan dan kecerobohan, namun penulis berharap tugas akhir ini dapat memberikan kontribusi baik untuk penelitian atau pembelajaran selanjutnya.

Bandung, Mei 2017

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
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	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	5
2.1 Domino Grid	5
2.1.1 Bone	5
2.1.2 Permainan <i>Domino Grid</i>	5
2.1.3 Representasi Papan Permainan	6
2.2 Algoritma <i>Human</i>	7
2.2.1 Menempatkan <i>bone</i>	7
2.2.2 <i>Forced Location</i>	9
2.3 Algoritma Odometer	10
3 ANALISIS	15
3.1 Analisis Algoritma	15
3.1.1 Algoritma <i>Human</i>	15
3.1.2 Odometer	24
3.2 Analisis Perangkat Lunak	33
3.2.1 Deskripsi Program	33
3.2.2 Diagram Use Case	33
3.2.3 Skenario Use Case	33
3.2.4 Diagram Kelas Secara Garis Besar	35
4 PERANCANGAN	37
4.1 Rancangan <i>User Interface</i>	37
4.2 Diagram Kelas	39
4.2.1 Kelas Bone	41
4.2.2 Kelas Position	42
4.2.3 Kelas Domino	43
4.2.4 Kelas DominoGUI	45

4.3	Spesifikasi Input dan output	45
4.3.1	Spesifikasi Input	45
4.3.2	Spesifikasi Output	46
5	IMPLEMENTASI DAN PENGUJIAN	47
5.1	Implementasi	47
5.1.1	Lingkungan Implementasi	47
5.1.2	Implementasi User Interface	48
5.2	Pengujian	49
5.2.1	Pengujian Kebenaran Algoritma	49
5.2.2	Pengujian Waktu	59
6	KESIMPULAN DAN SARAN	63
6.1	Kesimpulan	63
6.2	Saran	63
	DAFTAR REFERENSI	65
	A KODE PROGRAM	67

DAFTAR GAMBAR

1.1	Papan Permainan	1
2.1	<i>Bone</i>	5
2.2	Papan permainan dengan beberapa bone yang sudah diletakkan	6
2.3	(a) Papan permainan (b) representasi dalam array	6
2.4	(a) Papan permainan (b) banyaknya kemungkinan posisi untuk masing masing bone	7
2.5	Hasil peletakan bone	8
2.6	(a) Papan permainan (b) banyaknya kemungkinan	8
2.7	Hasil Penggandaan Papan dan peletakan bone (0,3)	9
2.8	(a) Contoh forced location untuk bone (4,4) (b) Contoh forced location untuk bone (5,6)[1]	9
2.9	Contoh Odometer	10
2.10	Odometer <i>rollover</i>	10
2.11	Odometer	11
2.12	(a) Odometer (b) Nilai Maksimum Odometer	11
2.13	Contoh Rollover	13
3.1	Papan Permainan	15
3.2	(a) Papan Permainan (b) Kemungkinan Lokasi Untuk Bone (0,0)	16
3.3	(a) Hasil Transpose (b) Kemungkinan Posisi Untuk Bone (0,0)	16
3.4	Hasil Peletakan 4 bone	17
3.5	Hasil Penghapusan	17
3.6	Papan Permainan	17
3.7	Contoh Penggandaan Papan Untuk Bone (0,3)	18
3.8	Kemungkinan Posisi Untuk Posisi [1,1]	18
3.9	Forced Location Pada Posisi [2,0]	18
3.10	Contoh Papan Berukuran 5x4	19
3.11	Kondisi Papan Setelah peletakan (0,0) dan (1,1)	19
3.12	Kondisi Papan Setelah peletakan <i>forced location</i>	20
3.13	Kondisi Papan Setelah peletakan (0,3)	20
3.14	Kondisi Papan Setelah <i>forced location</i>	21
3.15	(a) bone (0,1) kemungkinan pertama (b) bone (0,1) kemungkinan kedua (c) bone (2,2) kemungkinan pertama (d) bone (2,2) kemungkinan kedua	22
3.16	Kondisi Papan Saat Ini	22
3.17	Kondisi Papan Setelah Peletakan <i>bone</i> (3,3)	23
3.18	Kondisi Papan Setelah Peletakan <i>bone</i> (1,3)	24
3.19	(a) bone (1,2) kemungkinan pertama (b) bone (1,2) kemungkinan kedua	25
3.20	Solusi akhir papan	25
3.21	Proses Pengecekan	27
3.22	Contoh Papan Berukuran 5x4	27
3.23	Contoh Kasus	30
3.24	Diagram Use Case	34
3.25	Diagram Kelas secara Garis Besar	35

4.1	Rancangan <i>User Interface</i>	37
4.2	Rancangan <i>User Interface</i> (2)	38
4.3	Rancangan <i>User Interface</i> (3)	39
4.4	Diagram Kelas	40
4.5	Kelas Bone	41
4.6	Kelas Position	42
4.7	Kelas Domino	43
4.8	Kelas DominoGUI	45
5.1	User Interface	48
5.2	User Interface (2)	49
5.3	User Interface (3)	50
5.4	Contoh Kasus 1	51
5.5	Hasil Algoritma Human Kasus 1	51
5.6	Hasil Algoritma Odometer Kasus 1	52
5.7	Contoh Kasus 2	52
5.8	Hasil Algoritma Human Kasus 2	53
5.9	Hasil Algoritma Odometer Kasus 2	53
5.10	Contoh Kasus 3	54
5.11	Hasil Algoritma Human Kasus 3	54
5.12	Hasil Algoritma Odometer Kasus 3	55
5.13	Contoh Kasus 4	55
5.14	Hasil Algoritma Human Kasus 4	56
5.15	Hasil Algoritma Human Kasus 4	56
5.16	Contoh Kasus 5	57
5.17	Hasil Algoritma Human Kasus 5	57
5.18	Hasil Algoritma Odometer Kasus 5	58
5.19	Contoh Kasus 6	58
5.20	Hasil Algoritma Human Kasus 6	59
5.21	Hasil Algoritma Odometer Kasus 6	59
5.22	Contoh Kasus 7	60
5.23	Hasil Algoritma Human Kasus 7	60
5.24	Hasil Algoritma Human Kasus 7	61
5.25	Contoh Kasus 8	61
5.26	Contoh Kasus 9	61
5.27	Contoh Kasus 10	61
5.28	Contoh Kasus 11	61
5.29	Contoh Kasus 12	62
5.30	Contoh Kasus 1	62
5.31	Contoh Kasus 2	62
5.32	Contoh Kasus 3	62
5.33	Contoh Kasus 4	62

DAFTAR TABEL

2.1	Contoh Kemungkinan Setiap Posisi	12
3.1	Tabel Frekuensi	19
3.2	Tabel Frekuensi(2)	20
3.3	Tabel Frekuensi(3)	21
3.4	Tabel Frekuensi(4)	23
3.5	Tabel Frekuensi(4)	24
3.6	Contoh Kemungkinan Setiap Posisi	26
3.7	Frekuensi setiap bone	28
3.8	Frekuensi setiap bone setelah diurutkan	28
3.9	Frekuensi setiap bone	30
3.10	Frekuensi setiap bone	30
3.11	Rollover	31
3.12	Setelah rollover digit kesembilan	31
3.13	Setelah rollover digit kedelapan	31
3.14	Setelah rollover digit ketujuh	31
3.15	Setelah rollover digit keenam	31
3.16	Setelah rollover digit kelima	32
5.1	Hasil Perbandingan	60

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Domino adalah sebuah permainan yang menggunakan sekumpulan kartu yang biasa disebut *bone* atau *tile*. Pada umumnya *bone* terdiri dari kumpulan 28 kartu. Setiap kartu domino dibagi menjadi 2 bagian, setiap bagian akan ditandai oleh satu atau lebih titik atau tidak ditandai sama sekali. Setiap *bone* dinamai berdasarkan nilai pada kedua sisinya, sebagai contoh (2,5) atau (5,2) adalah penamaan untuk bone bernilai 2 dan 5.

Kenneth E. Caviness seorang ahli matematika mengajukan masalah *Domino Grid*. *Domino Grid* sendiri adalah permainan tentang bagaimana cara untuk menutupi papan permainan dengan seluruh kartu domino yang tersedia. Permainan *Domino Grid* ini terdiri dari papan permainan dan *bone*. Papan permainan adalah bidang dengan ukuran 7x8 kotak yang berisi dengan angka-angka mulai dari 0 sampai dengan 6.

1	4	4	4	4	4	0	0
1	2	1	6	6	2	2	4
1	2	0	0	0	6	6	6
5	2	0	2	0	0	2	2
1	3	3	3	3	5	5	5
4	3	3	3	6	6	5	5
4	5	1	1	1	6	5	3

Gambar 1.1: Papan Permainan

Masalah dari permainan ini adalah bagaimana caranya untuk dapat menemukan solusi permainan ini . Cara yang paling mudah adalah menggunakan algoritma *brute force* yaitu dengan mencoba segala kemungkinan yang dapat dilakukan. Algoritma *brute force* ini bekerja dengan mencoba seluruh kemungkinan *bone* yang dapat diletakkan pada papan. Walaupun cara tersebut dapat menemukan jawaban dari permainan tersebut tapi waktu yang digunakan untuk mencapai jawaban tersebut sangat lama sehingga algoritma ini tidak dapat diandalkan.

Pada skripsi ini akan menggunakan algoritma *human* dan juga algoritma odometer untuk menemukan solusi dari permainan *domino grid*. Setelah melakukan pembahasan kedua algoritma, akan dibuat program yang mengimplementasikan kedua algoritma tersebut, dan akan dilakukan perbandingan antara kedua algoritma tersebut. Kedua algoritma ini dipilih untuk menyelesaikan masalah ini karena kedua algoritma ini pasti dapat menemukan solusi selama papan yang digunakan memiliki minimal satu kemungkinan solusi. Selain itu juga waktu *run* kedua algoritma jauh lebih cepat daripada algoritma *brute force*.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang sudah dibahas, berikut merupakan rumusan masalah yang akan dibahas pada penelitian ini

- Bagaimana cara kerja dari *human algorithm* dan *odometer*?
- Bagaimana cara mengimplementasikan algoritma *human* dan *odometer*?
- Bagaimana waktu *run* algoritma *human* dan *odometer*?

1.3 Tujuan

- Memahami cara kerja dari *human algorithm* dan *odometer*
- Membuat perangkat lunak yang mengimplementasikan algoritma *human* dan *odometer*
- Membandingkan waktu *run* untuk algoritma *human* dan *odometer*

1.4 Batasan Masalah

- Masukan papan permainan diasumsikan memiliki minimal satu buah solusi.
- Nilai maksimum *bone* yang dapat dimasukkan adalah 4 sampai 6.

1.5 Metodologi

Metodologi yang akan dilakukan dalam penelitian ini adalah sebagai berikut:

- Melakukan studi pustaka untuk mempelajari algoritma *human* dan *odometer*.
- Mengimplementasikan algoritma *human* dan *odometer*.
- Melakukan pengujian terhadap perangkat lunak yang telah dibuat .
- Mendokumentasikan keseluruhan proses selama penelitian berlangsung.

1.6 Sistematika Pembahasan

1. Bab Pendahuluan

Bab 1 berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan.

2. Bab Dasar Teori

Bab 2 berisi studi pustaka mengenai *human algorithm* dan *odometer*.

3. Bab Analisis

Bab 3 berisi tentang cara kerja *human algorithm*, cara kerja *odometer*, diagram kelas secara garis besar, dan analisis perangkat lunak.

4. Bab Perancangan

Bab 4 berisi rancangan user interface, diagram kelas *human algorithm* dan *odometer*.

5. Bab Implementasi dan Pengujian

Bab 5 berisi lingkungan implementasi, hasil implementasi, rancangan pengujian, lingkungan pengujian dan hasil dari pengujian tersebut.

6. Bab Kesimpulan dan Saran

Bab 6 berisi kesimpulan dari penelitian yang telah dilakukan dan saran yang ditujukan untuk penelitian lebih lanjut.