

SKRIPSI

**PERBANDINGAN KECEPATAN PENCARIAN
MENGUNAKAN ALGORITMA MINIMAX SECARA
SEKUENSIAL DAN SECARA PARALEL DALAM
PERMAINAN CONGKLAK**



Rifky Fatharani Kautsar

NPM: 2012730041

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

UNDERGRADUATE THESIS

**SPEED COMPARISON OF SEQUENTIAL AND PARALLEL
MINIMAX ALGORITHM IN CONGKLAK GAME**



Rifky Fatharani Kautsar

NPM: 2012730041

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

LEMBAR PENGESAHAN



**PERBANDINGAN KECEPATAN PENCARIAN
MENGUNAKAN ALGORITMA MINIMAX SECARA
SEKUENSIAL DAN SECARA PARALEL DALAM PERMAINAN
CONGKLAK**

Rifky Fatharani Kautsar

NPM: 2012730041

Bandung, 19 Desember 2017

Menyetujui,

Pembimbing

Luciana Abednego, M.T.

Ketua Tim Penguji

Aditya Bagoes Saputra, M.T.

Anggota Tim Penguji

Claudio Franciscus, 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:

PERBANDINGAN KECEPATAN PENCARIAN MENGGUNAKAN ALGORITMA MINIMAX SECARA SEKUENSIAL DAN SECARA PARALEL DALAM PERMAINAN CONGKLAK

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 19 Desember 2017



Rifky Fatharani Kautsar
NPM: 2012730041

ABSTRAK

Permainan congklak adalah permainan tradisional yang ada di Indonesia. Pada skripsi ini dibuat perangkat lunak komputer untuk mensimulasikan permainan congklak. Algoritma minimax digunakan untuk mencari langkah optimal pada permainan congklak. Namun pencarian secara sekuensial terbilang sangat lama, maka dibuatlah pencarian secara paralel. Dengan algoritma minimax secara paralel diharapkan dapat mempercepat waktu pencarian langkah optimal. Tujuan dari skripsi ini adalah membandingkan kecepatan algoritma minimax secara sekuensial dan secara paralel.

Berdasarkan hasil pengujian fungsional, algoritma minimax secara sekuensial dan secara paralel pada permainan congklak dapat diimplementasikan dengan baik ke dalam perangkat lunak dan menghasilkan langkah optimal dari kasus yang diberikan. Sedangkan, berdasarkan hasil pengujian eksperimen dengan membandingkan kecepatan algoritma minimax secara sekuensial dan secara paralel dengan kasus yang diberikan, dapat disimpulkan bahwa algoritma minimax secara paralel lebih cepat 44.5% dalam melakukan pencarian langkah optimal. Semakin besar data yang dibangkitkan dan batasan yang diberikan, semakin terlihat pula efisiensi algoritma minimax secara paralel.

Kata-kata kunci: permainan congklak, algoritma minimax, komputasi paralel, *task decomposition*

ABSTRACT

Congklak is one of Indonesian traditional game. In this undergraduate thesis, a computer software is created to simulate congklak game . Minimax algorithm can be used to find an optimal path to win congklak game. However searching using sequential minimax algorithm is fairly slow, thus parallel minimax algorithm is developed. Parallel minimax algorithm expected to speed up this searching time. The purpose of this undergraduate thesis is to compare the speed of sequential and parallel minimax algorithms.

Both sequential and parallel minimax algorithm are implemented for congklak game. Functionality test result shows that the optimal path produced by the two algorithms are the same for each given case. Several experiments is conducted to compare their speed. Results show that the minimax algorithm is 44.5% faster than the sequential algorithm. As the input data and search bounday gets bigger, the parallel algorithm shows more significant improvement compared to the sequential algorithm.

Keywords: congklak game, minimax algorithm, parallel programing, task decomposition

Skripsi ini saya persembahkan untuk keluarga tercinta.

KATA PENGANTAR

Ucapan terima kasih kepada Allah SWT., keluarga tercinta, teh Yani, kang Cecep, Faza, Fahmi, Swakresti, Naupal, Ira, Anita, Ganjar, dan seluruh teman - teman lain yang tidak dapat saya sebutkan satu - persatu.

Bandung, Desember 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 Penelitian | 2 |
| 1.6 Sistematika Penulisan | 3 |
| 2 DASAR TEORI | 5 |
| 2.1 Permainan Congklak | 5 |
| 2.2 Algoritma Minimax [1] | 5 |
| 2.3 Komputasi Paralel [2] | 8 |
| 2.3.1 <i>Thread</i> | 8 |
| 2.3.2 <i>ExecutorService, Future, dan Callable</i> | 9 |
| 2.3.3 Metode - Metode pada Komputasi Paralel | 9 |
| 2.4 Model-View-Controller (MVC) [3] | 10 |
| 2.5 <i>Library</i> Apache POI-XSSF [4] | 11 |
| 2.5.1 XSSFWorkbook | 11 |
| 2.5.2 XSSFSheet | 11 |
| 2.5.3 XSSFRow | 11 |
| 2.5.4 XSSFCell | 12 |
| 3 ANALISIS | 13 |
| 3.1 Analisis Algoritma MiniMax Pada Permainan Congklak | 13 |
| 3.2 Analisis Algoritma MiniMax Paralel pada Permainan Congklak | 15 |
| 3.3 Analisis Kelas | 16 |
| 4 PERANCANGAN | 19 |
| 4.1 Kebutuhan Masukan dan Keluaran | 19 |
| 4.2 Perancangan Antarmuka | 19 |
| 4.2.1 Antarmuka Manual | 19 |
| 4.2.2 Antarmuka Otomatis | 20 |
| 4.3 Diagram Kelas Rinci | 21 |
| 4.3.1 UI | 21 |
| 4.3.2 Controller | 24 |

| | | |
|----------|---|-----------|
| 4.3.3 | Engine | 26 |
| 4.3.4 | SavedTable | 28 |
| 4.3.5 | TheMiniMax | 31 |
| 4.3.6 | ParallelMiniMax | 33 |
| 4.3.7 | CallableMiniMax | 34 |
| 4.4 | Rincian <i>Method</i> | 35 |
| 4.4.1 | Kelas Engine | 37 |
| 4.4.2 | Kelas TheMiniMax | 37 |
| 4.4.3 | Kelas MinimaxParallel | 43 |
| 5 | IMPLEMENTASI DAN PENGUJIAN | 45 |
| 5.1 | Implementasi Antarmuka | 45 |
| 5.1.1 | Antarmuka Manual | 45 |
| 5.1.2 | Antarmuka Otomatis | 48 |
| 5.2 | Pengujian Fungsional | 49 |
| 5.2.1 | Pengujian Permainan Congklak | 49 |
| 5.2.2 | Pencarian Langkah Menggunakan Algoritma Minimax | 50 |
| 5.2.3 | Membaca dan Menulis Ke Dalam Berkas XLSX | 51 |
| 5.3 | Pengujian Experimental | 51 |
| 5.4 | Kesimpulan Pengujian Perangkat Lunak | 57 |
| 6 | KESIMPULAN DAN SARAN | 59 |
| 6.1 | Kesimpulan | 59 |
| 6.2 | Saran | 59 |
| | DAFTAR REFERENSI | 61 |
| | A KODE SUMBER | 63 |
| | B DATA LENGKAP KASUS YANG DIUJIKAN | 87 |

DAFTAR GAMBAR

| | | |
|------|--|----|
| 1.1 | Papan Congklak. | 1 |
| 2.1 | Contoh pohon permainan minimax dengan dua <i>ply</i> | 7 |
| 2.2 | Contoh pohon permainan minimax pada tic-tac-toe. | 8 |
| 2.3 | Contoh diagram pengerjaan tugas <i>Task Decomposition</i> | 10 |
| 3.1 | Sebuah pohon permaian dari pembangkitan algoritma minimax pada permainan congklak. <i>initial node</i> atau <i>state</i> awal berada di tingkat <i>max</i> (yang dianggap pemain pertama) dan menghasilkan banyak <i>node</i> baru yang berada di tingkat <i>min</i> (yang dianggap pemain kedua) dan seterusnya bergantian bermain sampai berakhir pada <i>terminal</i> | 14 |
| 3.2 | <i>node</i> yang berwarna merah akan dibangkitkan berdasarkan giliran pemain pertama dan <i>node</i> yang berwarna biru akan dibangkitkan berdasarkan giliran pemain kedua. | 15 |
| 3.3 | Contoh pohon permainan minimax yang diparalelkan. | 16 |
| 3.4 | Diagram Kelas analisis perangkat lunak permainan congklak. | 17 |
| 4.1 | Antarmuka Manual. | 20 |
| 4.2 | Antarmuka Otomatis. | 21 |
| 4.3 | Diagram kelas UI rinci perangkat lunak permainan congklak. | 22 |
| 4.4 | Diagram kelas Controller rinci perangkat lunak permainan congklak. | 25 |
| 4.5 | Diagram kelas Engine rinci perangkat lunak permainan congklak. | 27 |
| 4.6 | Diagram kelas SavedTable rinci perangkat lunak permainan congklak. | 29 |
| 4.7 | Diagram kelas TheMiniMax rinci perangkat lunak permainan congklak. | 31 |
| 4.8 | Diagram kelas ParallelMiniMax rinci perangkat lunak permainan congklak. | 33 |
| 4.9 | Diagram kelas CallableMiniMax rinci perangkat lunak permainan congklak. | 34 |
| 5.1 | Tampilan awal manual pada perangkat lunak. | 46 |
| 5.2 | Tampilan perangkat lunak setelah memilih lubang kecil. | 46 |
| 5.3 | Tampilan hasil menjalankan algoritma minimax secara sekuensial. | 47 |
| 5.4 | Tampilan hasil menjalankan algoritma minimax secara paralel. | 47 |
| 5.5 | Tampilan awal otomatis pada perangkat lunak. | 48 |
| 5.6 | Tampilan setelah pengujian dijalankan. | 49 |
| 5.7 | Perubahan kondisi papan permainan dari lubang kecil yang dipilih pengguna. | 50 |
| 5.8 | Hasil yang didapat menggunakan algoritma minimax secara sekuensial. | 50 |
| 5.9 | Hasil yang didapat menggunakan algoritma minimax secara paralel. | 51 |
| 5.10 | Masukan kasus berupa berkas input.xlsx yang ditampilkan dalam tabel <i>input</i> | 52 |
| 5.11 | Hasil yang didapat dari kasus - kasus yang diuji dimasukkan ke dalam berkas output.xlsx. | 53 |
| 5.12 | Rata - rata waktu (dalam detik) yang didapat dari setiap kasus menggunakan algoritma minimax secara sekuensial. | 54 |
| 5.13 | Rata - rata waktu (dalam detik) yang didapat dari setiap kasus menggunakan algoritma minimax secara paralel. | 54 |

| | | |
|------|--|----|
| 5.14 | Perbandingan waktu rata - rata setiap putaran secara sekuensial dan paralel pada kasus 1 | 55 |
| 5.15 | Perbandingan waktu rata - rata setiap putaran secara sekuensial dan paralel pada kasus 2 | 55 |
| 5.16 | Perbandingan waktu rata - rata setiap putaran secara sekuensial dan paralel pada kasus 3 | 56 |
| 5.17 | Perbandingan waktu rata - rata setiap putaran secara sekuensial dan paralel pada kasus 4 | 56 |
| 5.18 | Perbandingan waktu rata - rata setiap putaran secara sekuensial dan paralel pada kasus 5 | 56 |
| 5.19 | Perbandingan waktu rata - rata setiap putaran secara sekuensial dan paralel pada kasus 6 | 57 |
| 5.20 | Perbandingan waktu rata - rata setiap putaran secara sekuensial dan paralel pada semua kasus | 57 |

DAFTAR TABEL

| | |
|--|----|
| B.1 Kasus yang diuji | 87 |
| B.2 Hasil sekuensial dari kasus yang diuji | 88 |
| B.3 Hasil paralel dari kasus yang diuji | 89 |
| B.4 Hasil waktu rata - rata sekuensial dari kasus yang diuji | 90 |
| B.5 Hasil waktu rata - rata paralel dari kasus yang diuji | 90 |
| B.6 Hasil waktu rata - rata setiap putaran dari semua kasus yang diuji | 90 |

BAB 1

PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, tujuan, metode penelitian, dan sistematika penulisan yang digunakan untuk menyusun skripsi ini.

1.1 Latar Belakang

Congklak adalah permainan tradisional Indonesia yang dimainkan oleh dua orang. Permainan ini memakai papan berbentuk persegi panjang yang memiliki 16 lubang terdiri dari 14 lubang kecil atau anak dan dua lubang besar atau lumbung (Gambar 1.1). Masing - Masing pemain memiliki tujuh lubang kecil dan satu lubang besar yang berada di sebelah kiri pemain. Pemenang permainan congklak ini ditentukan dari banyaknya biji pada lubang besar dan pemain yang memiliki biji terbanyak pada lubang besar miliknya dialah pemenangnya.

Implementasi permainan congklak dalam perangkat lunak ini untuk membuat sistem cerdas yang dapat memainkan permainan congklak serta dapat menunjukkan perbandingan kecerdasan sistem yang menggunakan secara sekuensial dengan secara paralel. Sistem cerdas memiliki pengetahuan yang luas sehingga dapat memilih langkah terbaik yang dapat diambil berdasarkan pengetahuan yang dimiliki. Dalam permainan congklak pengetahuan yang dimaksud adalah kondisi - kondisi apa saja yang mungkin terjadi pada permainan congklak. Salah satu sistem cerdas yang dapat digunakan pada permainan congklak ini adalah algoritma minimax.

Algoritma Minimax adalah algoritma yang biasanya digunakan untuk membuat sebuah sistem cerdas yang digunakan pada permainan *multiplayer* seperti pada permainan congklak ini. Algoritma ini membangkitkan kondisi yang terjadi pada permainan congklak dan memilih salah satu langkah yang terbaik dari setiap kondisi yang dibangkitkan. Karena pada permainan congklak terlalu banyak kondisi jika harus dibangkitkan semua, maka kondisi yang dibangkitkan akan dibatasi. Namun algoritma ini membutuhkan waktu yang cukup lama untuk mencari langkah terbaiknya jika dilakukan pembangkitan dan pencarian satu per satu. Maka dibuatlah algoritma minimax secara paralel.

Komputasi paralel digunakan untuk mempercepat waktu pengerjaan perangkat lunak dengan mengerjakan langkah - langkah komputasi yang independen secara bersamaan. Sehingga langkah-langkah komputasi tersebut tidak harus menunggu langkah komputasi lain selesai dikerjakan.



Gambar 1.1: Papan Congklak.

Diharapkan dengan komputasi paralel ini algoritma minimax dapat mempercepat pembangkitan dan pencarian dibandingkan dengan algoritma minimax secara sekuensial.

1.2 Rumusan Masalah

Rumusan dari masalah yang akan dibahas pada skripsi ini sebagai berikut:

1. Bagaimana mengimplementasikan sistem cerdas permainan congklak menggunakan algoritma MiniMax secara sekuensial?
2. Bagaimana mengimplementasikan sistem cerdas permainan congklak menggunakan algoritma MiniMax secara paralel?
3. Bagaimana perbandingan kinerja yang dihasilkan sistem cerdas algoritma MiniMax sekuensial dengan sistem cerdas algoritma MiniMax paralel?

1.3 Tujuan

Tujuan-tujuan yang hendak dicapai melalui pengerjaan skripsi ini sebagai berikut:

1. Mengimplementasikan sistem cerdas permainan congklak menggunakan algoritma MiniMax secara sekuensial.
2. Mengimplementasikan sistem cerdas permainan congklak menggunakan algoritma MiniMax secara paralel.
3. Membandingkan kinerja sistem cerdas algoritma MiniMax sekuensial dan sistem cerdas algoritma MiniMax paralel.

1.4 Batasan Masalah

Beberapa batasan masalah yang terkait dengan pengerjaan skripsi ini sebagai berikut:

1. Prosesor yang digunakan untuk pengujian hanya empat prosesor.

1.5 Metodologi Penelitian

Metodologi penelitian yang dikerjakan pada skripsi ini adalah sebagai berikut :

1. Melakukan studi mengenai peraturan permainan congklak.
2. Melakukan studi mengenai algoritma MiniMax.
3. Melakukan studi mengenai komputasi paralel.
4. Membuat perangkat lunak permainan congklak sederhana, yang dapat menampilkan jumlah biji pada setiap lubang dan menampilkan waktu pencarian solusi terbaik.
5. Mengimplementasikan sistem cerdas permainan congklak menggunakan algoritma MiniMax secara sekuensial pada perangkat lunak.
6. Mengimplementasikan sistem cerdas permainan congklak menggunakan algoritma MiniMax secara paralel pada perangkat lunak.
7. Melakukan eksperimen dan pengujian.
8. Menulis dokumen skripsi.

1.6 Sistematika Penulisan

1. Bab Pendahuluan

Bab 1 berisi latar belakang, rumusan masalah, tujuan, metode penelitian, dan sistematika penulisan yang digunakan untuk menyusun skripsi ini.

2. Bab Dasar Teori

Bab 2 berisi teori-teori yang digunakan dalam pembuatan skripsi ini. Teori yang digunakan yaitu permainan congklak, Algoritma Minimax dan komputasi paralel.

3. Bab Analisis

Bab 3 berisikan analisis yang dilakukan pada skripsi ini, meliputi analisis perangkat lunak secara sekuensial dan paralel serta analisis kelas.

4. Bab Perancangan

Bab 4 berisikan perancangan perangkat lunak, meliputi diagram kelas rinci beserta deskripsi kelas dan fungsinya dan perancangan antarmuka perangkat lunak.

5. Bab Implementasi dan Pengujian

Bab 5 berisikan implementasi dan pengujian perangkat lunak, meliputi implementasi antarmuka, pengujian fungsional, dan pengujian eksperimental.

6. Bab Kesimpulan dan Saran

Bab 6 berisikan kesimpulan dari hasil pembangunan perangkat lunak beserta saran untuk pengembangan berikutnya.