

BAB III

METODE PENELITIAN

Dalam penyusunan skripsi ini, dirancang suatu sistem pemrosesan paralel pada permainan catur. Adapun metode penelitian yang digunakan pada penyusunan skripsi ini adalah:

3.1 Studi Literatur

Studi literatur berguna untuk memperoleh data dan menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi:

1. Mempelajari literatur yang menjelaskan tentang *chess programming*,
2. Mempelajari algoritma pencarian Shannon Type-A,
3. Pemrosesan paralel,
4. *Cluster Beowulf*,
5. Pemrograman dengan Open MPI.

3.2 Penentuan Spesifikasi Alat

Perangkat yang digunakan untuk implementasi pemrosesan paralel permainan catur diuraikan sebagai berikut:

a) Perangkat Keras:

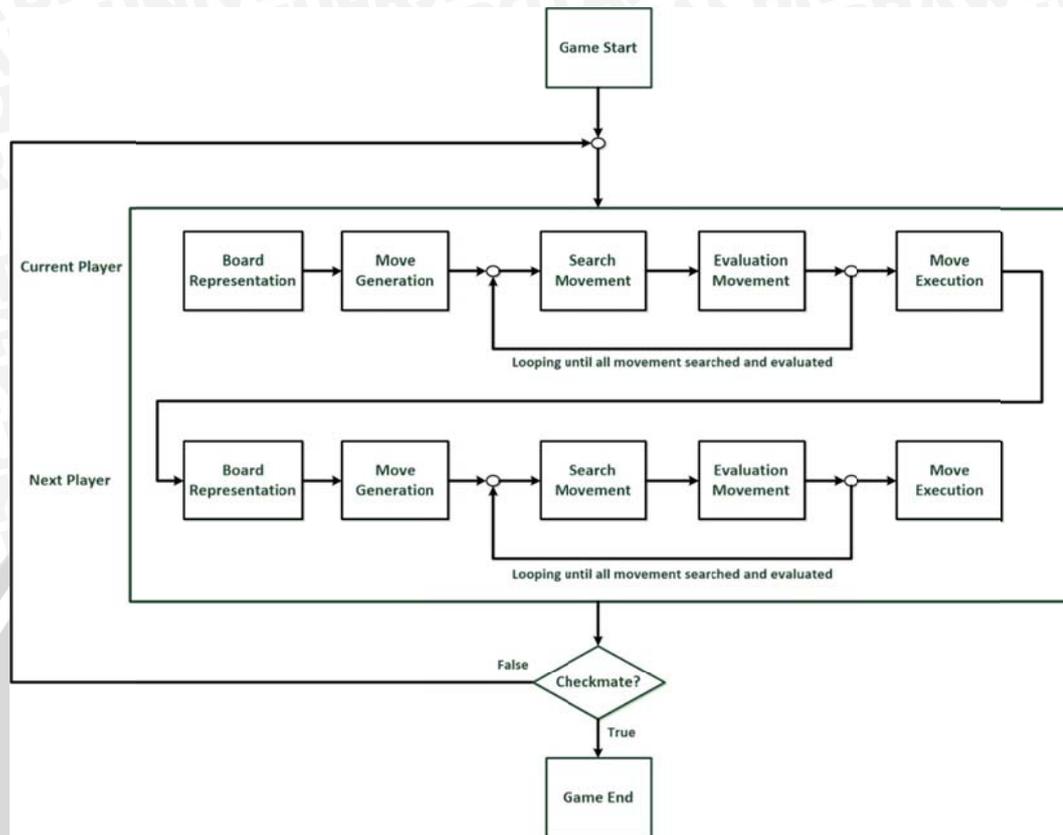
1. 4 unit komputer personal: Intel i3 550, DDR3 2 GB.
2. 1 unit komputer personal: Intel Dual Core E7400, DDR2 2 GB.
3. *Switch TL-SG1024D* dan perkabelan UTP Cat 5e.

b) Perangkat Lunak:

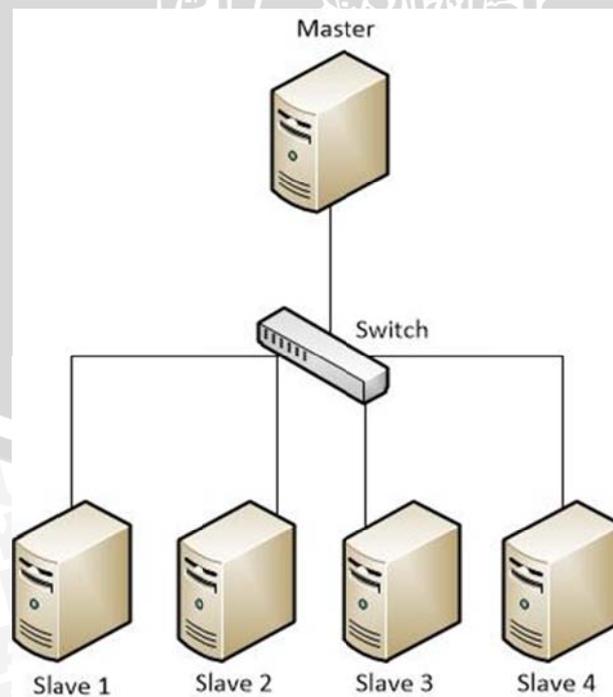
1. GNU/Linux Salix64 14.1: gcc 4.8.2, glibc 2.17, dan kernel Linux 3.10.17 64-bit (komputer *master*).
2. GNU/Linux Salix64 14.0: gcc 4.7.1, glibc 2.15, dan kernel Linux 3.2.29 64 bit (komputer *slave*).
3. Open MPI 1.8.1.

3.3 Diagram Sistem

Sistem pemrosesan paralel dari program catur ini adalah sebagai berikut:

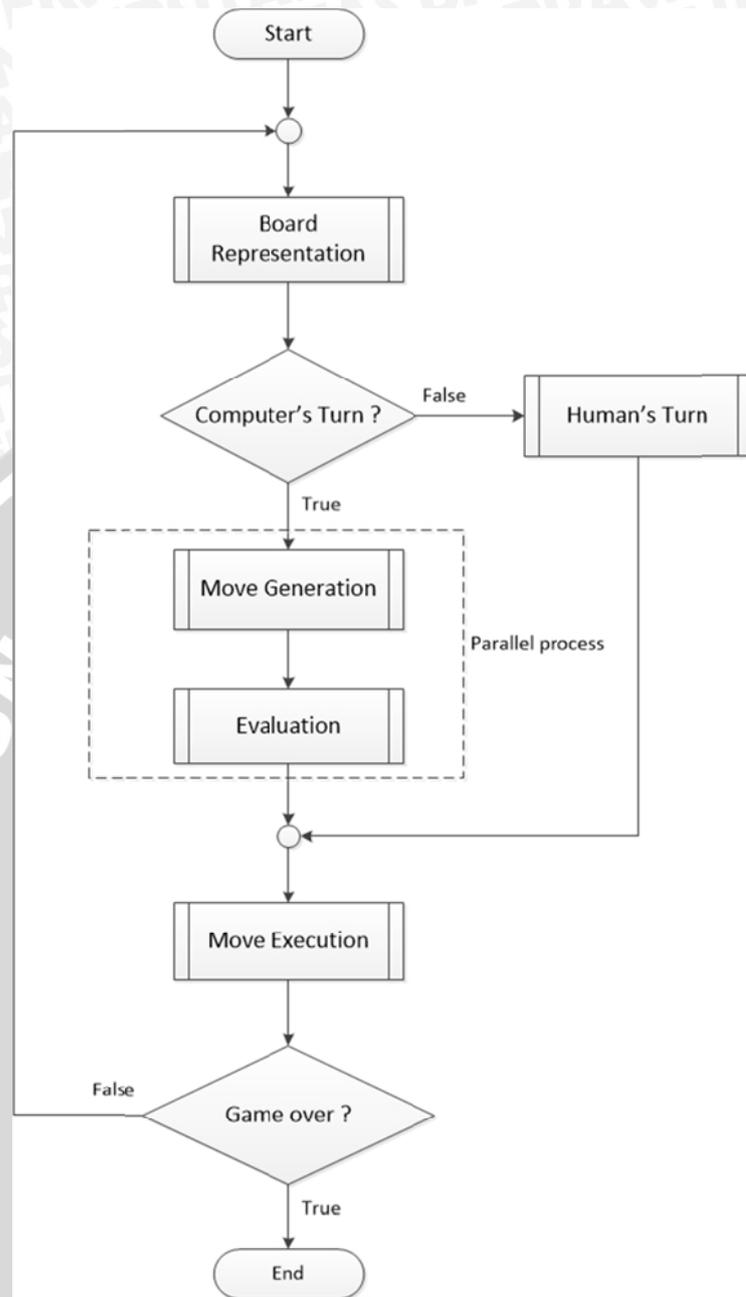


Gambar 3.1 Blok Diagram Sistem Permainan Catur



Gambar 3.2 Gambaran Umum Sistem Pemrosesan Paralel

3.4 Cara Kerja Sistem



Gambar 3.3 Flowchart Cara Kerja Sistem Pemrosesan Paralel pada Permainan Catur

3.4.1 Board Representation

Dasar dari sebuah chess engine adalah *board representation* atau representasi papan, yaitu sebuah sistem yang mencatat semua posisi bidak dalam papan dan menerapkan aturan yang berlaku. Langkah pertama untuk membuat *chess engine* adalah membuat representasi papan yang mengenal *setiap* peraturan dari catur, tidak boleh ada kesalahan sedikitpun, walaupun kadang untuk peraturan yang lebih rumit seperti castling (king-rook swap) dan pendeteksian hasil seri cukup susah untuk diterapkan.

Disamping memodelkan papan catur dengan penempatan bidaknya, informasi tambahan juga diperlukan, seperti giliran (turn) saat ini dan juga beberapa kemungkinan lain.

Implementasi *square centric* merupakan sebuah *inverse association* – apakah satu petak kosong atau sedang diisi oleh suatu bidak. Representasi *square centric* yang paling populer adalah 8*8 Board.

8*8 Board sebagai *basic square-centric board representation* adalah dua dimensi array of bytes (atau integer), yang berisi kode bidak dan petak kosong, atau yang lebih sering adalah array satu dimensi dalam range 0-63.

3.4.2 Move Generation

Setelah representasi papan, satu hal penting yang harus dilakukan adalah pembuatan gerakan (*move generation*). Hal ini penting sebagai *gameplay* dari sebuah program catur, dan hal ini harus sepenuhnya benar. Membuat *move generator* yang baik merupakan langkah dasar pembuatan program catur.

3.4.3 Proses Melakukan Langkah

Setelah pemain manusia melakukan gerakan dan diterima oleh komputer, sekarang adalah giliran dari komputer untuk bergerak: mencari jawaban yang paling tepat. Logika dari program ini menggunakan algoritma Shannon Type-A.

Langkah-langkahnya sebagai berikut:

1. Melakukan *scanning* untuk menemukan posisi tiap bidak.
2. Menemukan seluruh pergerakan yang mungkin untuk setiap bidak.
3. Mencari langkah yang paling baik (prediksi berdasarkan nilai) dari manusia untuk setiap pergerakan di proses(2).
4. Komputer terus berpikir dengan mengulangi proses(2) dan (3) sesuai dengan tingkat kedalaman pemikiran yang telah ditetapkan.

A. Melakukan scan pada papan catur.

Program melakukan proses *scanning* untuk menemukan seluruh posisi bidak. Lalu akan dilakukan proses pengecekan apakah bidak yang ditemukan warnanya sama dengan warna bidak komputer.

B. Menemukan seluruh gerakan yang mungkin

Setelah semua posisi bidak sudah ditemukan, kemudian hal selanjutnya yang dilanjutkan adalah dilakukan untuk menemukan seluruh gerakan yang mungkin untuk dilakukan pada kondisi papan catur saat ini.

Cara paling mudah adalah dengan mencoba menggerakkan seluruh bidak pada semua petak yang tersedia lalu mengecek apakah gerakan tersebut sesuai dengan peraturan atau tidak.

C. Mengevaluasi papan jika suatu gerakan dilakukan

Setiap gerakan yang mungkin harus dianalisa dan dinilai berdasarkan perubahan papan yang terjadi akibat langkah tersebut. Setelah seluruh evaluasi selesai, dipilih gerakan yang paling bagus, yaitu gerakan yang akan menghasilkan hasil evaluasi paling tinggi.

Relasi Antara Jenis Bidak dan Evaluasinya

Sebelumnya harus ditentukan dulu nilai untuk setiap bidak, contohnya:

Pion	100
Kuda	300
Gajah	325
Benteng	500
Menteri	900
Raja	30000

Raja bernilai sangat besar karena jika Raja sampai tertangkap berarti permainan akan berakhir, karena itu dalam setiap evaluasi, hal tertinggi yang akan dilakukan adalah “mencoba menangkap Raja musuh” dan hal yang tidak akan dilakukan adalah “membiarkan Raja sendiri tertangkap”.

Penilaian dilakukan dengan cara melakukan perulangan dan melakukan perbandingan dengan bidak lawan yang ada di langkah tujuan, lalu mengurangi nilai bidak sendiri dengan bidak lawan sesuai nilai diatas. Perulangan ini akan diproses secara tunggal dan paralel untuk mengetahui perbedaan waktu pemrosesannya.

Contoh Evaluasi



Gambar 3.4 Contoh Proses Evaluasi

Anggaplah saat ini giliran putih untuk bermain, saat proses iterasi seluruh gerakan untuk seluruh bidak yang mungkin dilakukan tiba pada giliran Ratu (Queen). Algoritma ini bertujuan untuk memperbesar nilai (gain) pada **setiap** kesempatan.

Kondisi Evaluasi saat ini:

$$\text{Nilai Evaluasi Putih} = \text{Jumlah Nilai Bidak Putih} - \text{Jumlah Nilai Bidak Hitam}$$

$$\begin{aligned} \text{Jumlah Nilai Bidak Putih} &= [2 \text{ Rook} + 2 \text{ Bishop} + 2 \text{ Knight} + 8 \text{ Pawn} + \text{Queen} + \text{King}] \\ &= [2*500 + 2*325 + 2*300 + 8*100 + 900 + 30000] \\ &= 1000 + 650 + 600 + 800 + 900 + 30000 \\ &= 33950 \end{aligned}$$

$$\begin{aligned} \text{Jumlah Nilai Bidak Hitam} &= [2 \text{ Rook} + 2 \text{ Bishop} + \text{Knight} + 8 \text{ Pawn} + \text{Queen} + \text{King}] \\ &= [2*500 + 2*325 + 300 + 8*100 + 900 + 30000] \\ &= 1000 + 650 + 300 + 800 + 900 + 30000 \\ &= 33650 \end{aligned}$$



$$\begin{aligned}
 \text{Nilai Evaluasi Putih} &= \text{Jumlah Nilai Bidak Putih} - \text{Jumlah Nilai Bidak Hitam} \\
 &= 33950 - 33650 \\
 &= \mathbf{300}
 \end{aligned}$$

Nilai Evaluasi saat ini untuk putih adalah Positif **300** (menang).

Dalam keadaan seperti diatas ratu akan mencoba memakan bidak lawan, dengan dua tingkat kedalaman sebagai berikut:

- **H5 -> H6 (Panah Merah)**

Depth 1

Ratu di posisi H5 memakan pion lawan di H6:

$$[2*500 + 2*325 + 2*300 + 8*100 + 900 + 30000] - [2*500 + 2*325 + 300 + 7*100 + 900 + 30000] = 33950 - 33550 = 400 \text{ (Positif, menang)}$$

Depth 2

Prediksi balasan hitam yang paling menguntungkan adalah gajah memakan ratu yang baru saja maju ke H6 (F8 -> H6)

$$[2*500 + 2*325 + 2*300 + 8*100 + 30000] - [2*500 + 2*325 + 300 + 7*100 + 900 + 30000] = 33050 - 33550 = -500 \text{ (Negatif, kalah)}$$

Kesimpulan: Langkah yang buruk, karena kita mengorbankan ratu hanya untuk memakan satu pion.

- **H5 -> G6 (Panah Biru)**

Depth 1

Ratu di posisi H5 memakan pion lawan di G6:

$$[2*500 + 2*325 + 2*300 + 8*100 + 900 + 30000] - [2*500 + 2*325 + 300 + 7*100 + 900 + 30000] = 33950 - 33550 = 400 \text{ (Positif, menang)}$$

Depth 2

Prediksi balasan hitam yang paling menguntungkan adalah benteng memakan ratu yang baru saja maju ke G6 (G8 -> G6)

$$[2*500 + 2*325 + 2*300 + 8*100 + 30000] - [2*500 + 2*325 + 300 + 7*100 + 900 + 30000] = 33050 - 33550 = -500 \text{ (Negatif, kalah)}$$

Kesimpulan: Langkah yang buruk, walaupun kita bisa melakukan skak tetapi dalam langkah selanjutnya kita akan kehilangan ratu sekaligus posisi skak tersebut.

- **H5 -> D5 (Panah Hijau)**

Depth 1

Ratu di posisi H5 memakan pion lawan di D5:

$$[2*500 + 2*325 + 2*300 + 8*100 + 900 + 30000] - [2*500 + 2*325 + 300 + 7*100 + 900 + 30000] = 33950 - 33550 = 400 \text{ (Positif, menang)}$$

Dan juga menempatkan ratu dalam posisi yang bagus untuk memakan benteng musuh di G8 pada giliran selanjutnya.

Depth 2

karena tidak ada bidak putih yang bisa dimakan, maka hitam akan memilih satu gerakan bidak hitam dengan posisi indeks terkecil. Tentu dengan mengecek aturan dari gerakan setiap bidak.

Kesimpulan: Langkah yang bagus, karena selain bisa memperbesar skor untuk putih, juga menempatkan pada posisi yang bagus untuk giliran selanjutnya.

Catatan: Jika kita menggunakan tiga depth, gerakan yang akan dilakukan hitam adalah memajukan pion E7 ke E6 untuk **meminimalisasi** kerugian, yaitu kehilangan benteng pada giliran selanjutnya.

Dari ketiga prediksi diatas (-500, -500, 400) maka diambil gerakan yang akan menghasilkan evaluasi terbaik untuk putih, yaitu H5 -> D5 (Panah Hijau).

D. Melakukan gerakan

Setelah semua gerakan selesai dianalisa dan dinilai, maka dilakukan proses pemilihan gerakan yang paling baik, yaitu gerakan dengan nilai evaluasi yang paling menguntungkan untuk giliran pemain saat ini.

Jika ternyata ditemukan ada lebih dari satu gerakan yang nilai evaluasinya sama besar, maka proses pemikiran akan berlanjut. Namun jika sudah mencapai batas *depth* maksimum, dipilih node gerakan paling kiri (posisi indeks terkecil) dari node gerakan-gerakan yang nilai evaluasinya sama besar.

3.5 Pengujian Sistem

Pengujian dilakukan untuk mengukur performansi sistem yang meliputi: waktu proses komputasi dan peningkatan kecepatan proses komputasi.

Waktu proses komputasi diukur dengan menghitung waktu proses komputasi terhadap jumlah komputer *slave* yang digunakan.

Peningkatan kecepatan diukur dengan membandingkan waktu eksekusi proses komputasi tunggal dengan waktu eksekusi proses komputasi paralel.

3.6 Pengambilan Kesimpulan dan Saran

Pada tahap ini, kesimpulan dan analisis dari pengujian dipaparkan. Tahap berikutnya adalah pembuatan saran untuk perbaikan dan pengembangan penelitian selanjutnya.

