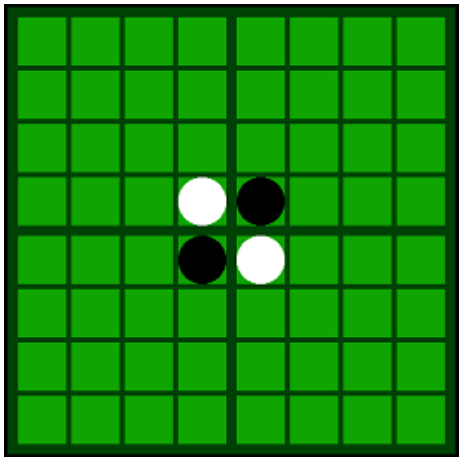
BAB III

OBJEK DAN METODOLOGI PENELITIAN

3.1. Objek Penelitian

3.1.1. Permainan Reversi

Permainan Reversi adalah permainan yang dimainkan oleh dua orang pemain. Permainan ini dimainkan di atas papan persegi yang terdiri dari 8 baris dan 8 kolom kotak-kotak kecil. Peralatan lain yang dibutuhkan adalah koin berwarna gelap dan koin berwarna terang (umumnya warna hitam dan warna putih) masing-masing sebanyak 64 buah. Pada awal permainan akan diletakkan dua koin hitam dan dua koin putih pada tengah-tengah papan. Untuk lebih jelasnya, dapat dilihat pada Gambar 3.1.

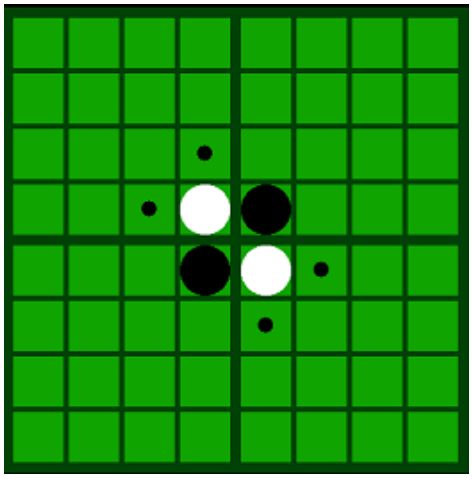


Gambar 3.1. Keadaan awal permainan Reversi

3.1.1.1. Aturan Permainan Reversi

Aturan permainan Reversi secara umum adalah sebagai berikut :

1. Pemain yang menggunakan koin hitam akan bermain terlebih dahulu.
2. Bila pemain koin hitam yang akan bermain, maka koin hitam harus diletakkan di kotak yang dapat dilompati oleh koin hitam lainnya. Dan begitu juga kondisinya untuk pemain yang menggunakan koin putih. Gambar 3.2 memperlihatkan kondisi awal permainan Reversi.



Gambar 3.2. Kotak yang mungkin (giliran langkah hitam)

3. Apabila salah satu pemain tidak dapat bermain karena tidak ada kotak yang sesuai dengan aturan nomor 2, maka pemain yang satunya lagi yang bermain.
4. Apabila kedua pemain sama-sama tidak dapat mengambil langkah lagi, maka permainan berakhir.

Permainan Reversi berakhir dengan beberapa kondisi sebagai berikut :

1. Semua kotak pada papan sudah penuh diisi koin-koin.
2. Belum semua kotak pada papan diisi tetapi koin-koin yang ada pada papan hanya tersisa koin-koin dalam 1 warna saja.
3. Kedua pemain setuju untuk mengakhiri permainan (bisa seri ataupun menyerah).

Pemenang pada permainan Reversi ditentukan dengan jumlah koin-koin yang ada pada papan permainan. Pemain dengan jumlah koin yang lebih banyak adalah pemenangnya. Jadi, pada permainan Reversi ini para pemain diharapkan dapat memikirkan strategi-strategi agar mendapatkan jumlah koin terbanyak pada akhir permainan.

3.1.1.2. Strategi Permainan Reversi

Reversi atau Othello merupakan permainan yang mudah untuk dipelajari, tetapi sangat memerlukan waktu lama untuk menjadi handal. Seperti judul buku yang ditulis mantan juara dunia othello Brian Rose "*Othello : A Minute to Learn – A Lifetime to Master*". Oleh karena itu diperlukan strategi dalam permainan papan ini, berikut hal-hal yang perlu diperhatikan dalam bermain reversi atau othello :

1. Jumlah Keping Koin

Tujuan permainan reversi atau othello adalah untuk memiliki jumlah keping sebanyak-banyaknya di akhir permainan. Karena inilah pemain pemula othello selalu berusaha melangkah di kotak-kotak di mana dia bisa membalik keping lawan sebanyak-banyaknya. Strategi ini disebut *greedy strategy* (strategi rakus).

Greedy Strategy memang diperlukan di akhir permainan untuk memperbanyak jumlah keping, tetapi di awal dan di tengah permainan strategi ini akan memberikan kondisi buruk dengan alasan *mobility* (dijelaskan di point 4). Untuk itu disarankan supaya menjaga jumlah keping sedikit di awal dan tengah permainan, tetapi tentu saja harus memperbanyak keping di akhir permainan.

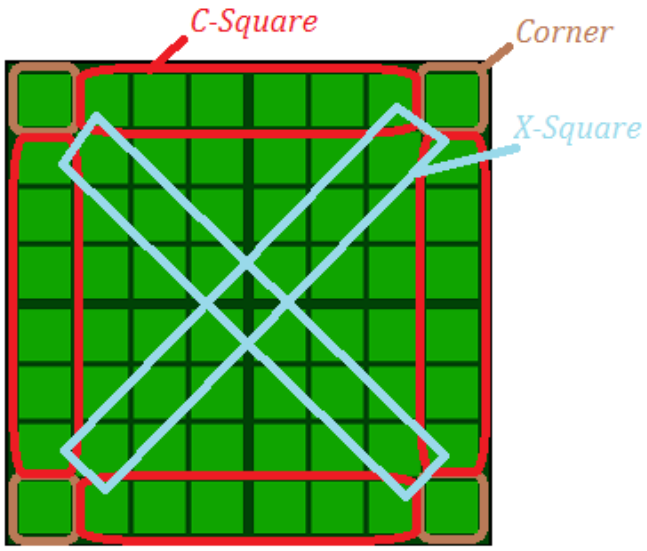
2. Kotak Sudut (*corners*), *X-Squares*, dan *C-Squares*

Jika pemain berhasil meletakan keping di kotak sudut (*corner*), maka keping itu tidak akan pernah bisa dibalik oleh lawan, karena tidak ada keping yang dapat mengapitnya. Bahkan bermula dari keping sudut ini, seorang pemain dapat menggunakannya untuk membalik keping-keping lawan di sekitarnya. Itulah sebabnya kotak sudut adalah kotak paling penting pada permainan othello. Untuk itu pemain disarankan supaya berusaha mendapatkan kotak sudut di awal dan di tengah permainan.

Kotak *X-Square* adalah kotak di sebelah diagonal kotak sudut. Kotak ini dipandang sebagai kotak yang paling berbahaya, karena ketika pemain meletakan keping di kotak ini, lawan dapat segera memanfaatkannya untuk mendapatkan kotak sudut di sebelahnya. Karena itulah disarankan untuk tidak melangkah di kotak *X-Square* pada awal dan tengah permainan.

Kotak *C-Square* adalah kotak di sebelah kotak sudut baik secara horizontal atau vertikal. Kotak ini juga dipandang berbahaya setelah *X-Squares*, karena pemain lawan dapat memanfaatkannya untuk mendapatkan sudut dengan trik-trik

tertentu. Untuk itu disarankan supaya tidak melangkah ke kotak *C-Square* di awal dan tengah permainan.



Gambar 3.3. Posisi *Corner*, *C-Square*, dan *X-Square* pada Papan Reversi

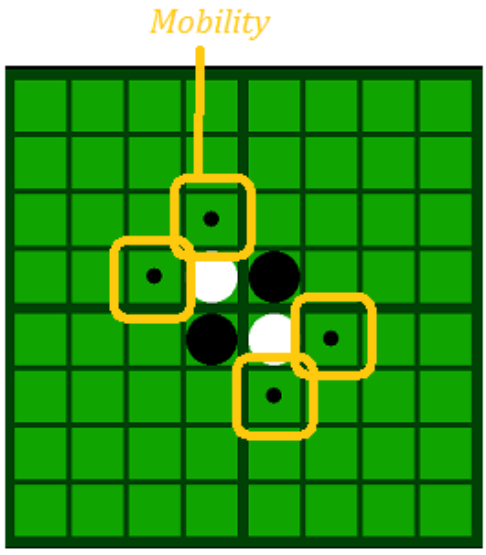
3. Keping Stabil

Keping yang tidak dapat diapit oleh keping lawan, sehingga tidak dapat dibalik disebut keping stabil. Selain itu keping di sisi papan yang bersambung dengan keping sudut juga merupakan keping stabil, atau keping yang semua baris horizontal, vertikal, dan diagonalnya bersambung dengan keping stabil lainnya juga merupakan keping stabil. Keping stabil ini menjadi penentu kemenangan secara mutlak karena tidak bisa dibalik lagi menjadi keping lawan. Untuk itu disarankan supaya pemain memperbanyak jumlah keping stabil di semua tahapan permainan.

4. *Mobility*

Di permulaan permainan hitam dapat melangkah di empat buah kotak. Selanjutnya putih dapat melangkah di tiga buah kotak, demikian seterusnya jumlah kotak di mana pemain dapat melangkah, yaitu dapat mengapit keping lawan, berubah-ubah tergantung langkah lawan sebelumnya. Jumlah kotak di mana pemain dapat melangkah ini disebut *mobility*.

Semakin banyak *mobility*, maka semakin banyak kemungkinan pilihan langkah yang bagus, sebaliknya semakin sedikit *mobility* maka semakin sedikit pilihan langkahnya sehingga semakin besar kemungkinan hanya tersisa langkah-langkah buruk. Karena itu disarankan untuk memperbanyak *mobility* di semua tahapan permainan. *Mobility* berkaitan erat dengan jumlah keping. Apabila jumlah keping pemain banyak, maka semakin susah pemain mengapit keping lawan, sehingga *mobility* menjadi sedikit. Sebaiknya ketika jumlah keping pemain sedikit, akan semakin mudah pemain mengapit keping lawan sehingga *mobility* meningkat.



Gambar 3.4. *Mobility* pada Permainan Reversi

5. Keping Tepi (*Frontier Disc*)

Ini adalah keping yang terletak di tepi kotak kosong. Ketika jumlah keping tepi yang dimiliki banyak, maka semakin besar kemungkinan lawan dapat melangkah di tepi kotak kosong sampingnya, sehingga mobility lawan meningkat. Untuk itu disarankan supaya pemain mempersedikit keping tepi di semua tahapan permainan. Strategi mempersedikit keping tepi ini juga berarti pemain harus berusaha supaya kepingnya selalu tersambung satu sama lainnya dan berada di dalam kepungan keping lawan.

6. *Parity* (Ganjil – Genap)

Di akhir permainan, kotak kosong di papan sering terbagi menjadi beberapa kelompok, dan *parity* adalah kesempatan untuk melangkah terakhir di sebuah kelompok kotak kosong. Pemain yang dapat melangkah terakhir di sebuah kelompok kotak kosong akan diuntungkan karena mendapat kesempatan terkhir untuk membalik keping-keping lawan menjadi kepingnya. Karena itulah disarankan supaya pemain menjadi pemain terakhir yang melangkah di semua kotak kosong di akhir permainan.

Caranya adalah dengan menghitung jumlah kotak kosong di sebuah kelompok. Ketika jumlahnya genap maka pemain jangan melangkah kesana, biarkan lawan yang melangkah kesana. Sebaiknya ketika jumlahnya ganjil, maka usahakan untuk melangkah kesana sebelum lawan melangkah kesana. Dengan demikian pemain akan menjadi yang terakhir untuk melangkah di semua kelompok kotak kosong.

3.1.2. Algoritma Minimax pada Permainan Reversi

3.1.2.1. Analisa Masalah

Pada permainan reversi ini terdapat beberapa masalah yang teranalisis oleh penulis terkait bagaimana permainan reversi pada aplikasi diantaranya yaitu :

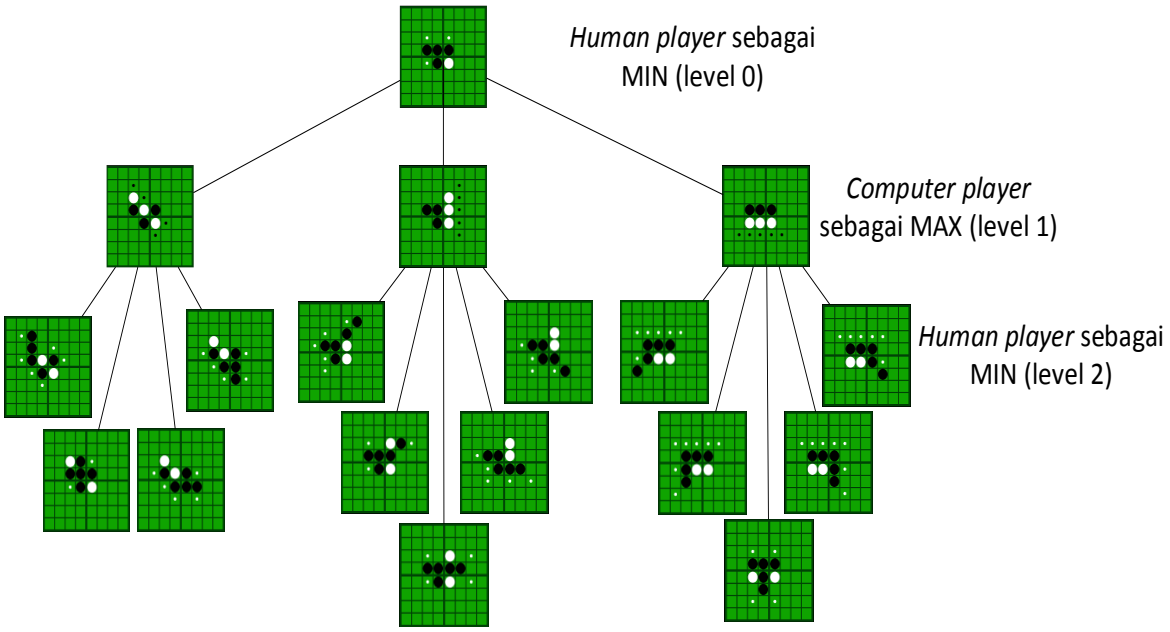
1. Penentuan *mobility* dari berbagai posisi bidak yang terjadi dengan mencari bidak lawan yang dapat diapit atau dibalik oleh bidak pemain secara horizontal, vertikal, dan diagonal.
2. Penentuan langkah yang terbaik dari beberapa *mobility* yang ada dengan memberikan nilai evaluasi pada setiap *mobility*.
3. Pemberian nilai evaluasi pada setiap mobility ditentukan dengan memperhitungkan beberapa faktor diantaranya adalah jumlah bidak, posisi *corner*, posisi x-*corner*, posisi c-*corner*, *mobility*, dan bidak stabil (artinya bidak yang tidak dapat dibalik oleh bidak lawan).
4. Pemakaian algoritma minimax tidak dapat terlepas dari metode pencarian yang dalam hal ini metode *depth-first search* sehinga akan digunakan untuk melakukan pencarian nilai evaluasi terbaik berdasarkan posisi MIN (*rival player*) atau MAX (*current player*) dari algoritma minimax.
5. Tidak mungkin dilakukan penelusuran secara menyeluruh pada setiap kemungkinan langkah sehingga harus dilakukan optimalisasi langkah menggunakan prosedur *alpha-beta pruning* sehingga akan memotong langkah yang tidak perlu.

3.1.2.2. Analisa Algoritma

3.1.2.2.1. Algoritma Minimax

Sebagaimana diketahui sebelumnya algoritma minimax merupakan algoritma yang sangat cocok untuk *type game playing* dengan dua player bergantian. Dalam minimax, MAX adalah sebutan bagi *computer player* yang bertujuan untuk mendapatkan nilai maksimal dan MIN adalah sebutan bagi *human player* yang bertujuan untuk mendapatkan nilai minimal.

Pada permainan reversi kali ini MAX akan ditetapkan sebagai *computer player* yang memiliki bidak berwarna putih dan MIN ditetapkan sebagai *human player* yang memiliki bidak berwarna hitam. Sebagai contoh langkah permainan reversi yang menggunakan konsep minimax dapat dilihat pada Gambar 3.3.



Gambar 3.5. Pohon Pencarian pada Permainan Reversi (2 kedalaman)

Pada keadaan sekarang (*root*) dapat dilihat *human player* (bidak hitam) sudah mendapatkan giliran bermain dan *computer player* (bidak putih) akan melakukan langkah berikutnya. Berdasarkan keadaan ini, *computer player* membuat pohon pencarian yang berisi kemungkinan langkah yang bisa diambil oleh *computer player* dan *human player* dengan kedalaman tertentu.

Kemudian *computer player* menganalisis keadaan tersebut berdasarkan prosedur minimax. *Node root* (akar) menunjukan keadaan papan sekarang, sedangkan *node-node* anak menunjukan kemungkinan langkah-langkah yang dapat dijalankan.

3.1.2.2.2. Fungsi Evaluasi pada permainan Reversi

Berdasarkan informasi yang dikutip dari sebuah situs atau blog milik Bowo Prasetyo (http://genethello.blogspot.com/2010/02/genethello-kembara-panjang-menuju-batas.html) yang diakses pada tanggal 23 Maret 2013. Secara umum, fungsi evaluasi yang sering digunakan pada aplikasi othello atau reversi mempertimbangkan beberapa variabel utama, antara lain :

1. *Discs* yaitu selisih jumlah keping yang dimiliki pemain dan yang dimiliki lawan main setelah dilakukannya sebuah langkah. Pada umumnya semakin tinggi nilai *discs* maka semakin baik langkah tersebut.
2. *Mobility* yaitu jumlah langkah yang dimiliki lawan main setelah dilakukannya sebuah langkah. Ketika jumlah langkah yang dimiliki lawan main banyak, maka memungkinkannya memilih langkah-

langkah yang bagus sehingga perlu memperkecil nilai *mobility* ini supaya lawan tidak mempunyai pilihan langkah yang bagus.

3. *Corners* yaitu jumlah sudut yang ditempati pemain dikurangi jumlah sudut yang ditempati lawan main setelah dilakukannya sebuah langkah. Posisi sudut pada permainan othello sangat penting, karena keping pada posisi ini bersifat permanen, yaitu tidak bisa dibalik lagi oleh lawan sampai permainan berakhir. Untuk itu perlu memperbesar nilai *corners* ini sehingga jumlah sudut yang kita tempati lebih banyak daripada yang ditempati lawan.
4. *Stables* yaitu jumlah keping stabil, artinya keping yang tidak bisa dibalik ole lawan sampai akhir permainan.
5. *Frontier* yaitu jumlah keping tepi yang ada, keping yang terletak di tepi kotak kosong. Ketika jumlah keping tepi yang kita miliki banyak, maka semakin besar kemungkinan lawan dapat melangkah di tepi kotak kosong sampingnya, sehingga mobility lawan meningkat.

Dengan demikian perhitungan score sebuah langkah pada permainan othello pada umumnya akan mengikuti rumus :

score = wd*discs + wc*corner + ws*stables + wm*mobility + wf*frontier

Keterangan :

wd = -3 (bobot untuk discs)

wc = 5 (bobot untuk corners)

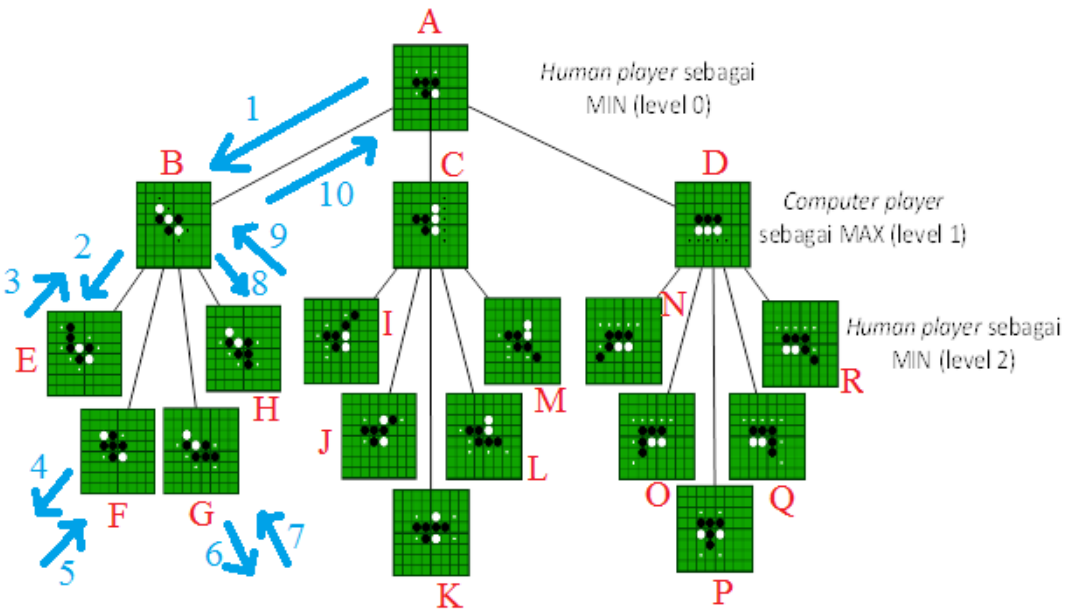
ws = 9 (bobot untuk stables)

wm = 7 (bobot untuk mobility)

wf = -5 (bobot untuk frontier)

3.1.2.2.3. Metode Pencarian *Depth-First Search* (DFS)

Adapun untuk proses penelusuran pada pohon pencarian yang diperlihatkan Gambar 3.5 diatas dilakukan menggunakan metode pencarian kedalaman pertama (*depth-first search*). Metode ini dikatakan pada beberapa literatur merupakan metode yang paling cocok digunakan pada algoritma minimax karena *depth-first search* pada awalnya menelusuri pada kedalaman tertentu (*leaf node*) terlebih dahulu.



Gambar 3.6. Metode DFS pada Pohon Permainan Reversi

Pada gambar 3.6 dapat dilihat langkah awal penelusuran dimulai dari *node root* (*node* A) ke *node* B kemudian ke *node* E yang merupakan *leaf node* (karena dalam hal ini hanya 2 kedalaman), dari *node* E ini mulai ada perhitungan nilai evaluasi pada kondisi tersebut dan akan menjadi bahan pertimbangan bagi *node* B sebelum melangkah ke *node* F, G, dan H. Jika ternyata *node* F, G, atau H memiliki nilai evaluasi lebih baik dari *node* E (dalam hal ini dicari nilai evaluasi terbesar atau MAX) maka *node* itulah yang menjadi tetapan untuk *node* B yang tentu akan menjadi bahan pertimbangan juga oleh *node* A dari *node* C dan D. Tanda panah pada gambar menunjukan alur proses penelusuran dan nomor menunjukan *step by step* yang dilakukan. Panah nomor 3, 5, 7, 9, 10 merupakan proses *backtrack* yang berarti kembali pada *node* sebelumnya. Demikian seterusnya hal yang sama juga dilakukan pada *node* C beserta *child node*-nya dan *node* D beserta *child node*-nya.

3.1.2.2.4. Penelusuran *Level* Permainan

Dalam hal ini metode pencarian atau metode penelusuran DFS (*Depth-First Search*) yang telah dijelaskan diatas akan dimanfaatkan untuk menciptakan perbedaan *level* permainan. Berikut beberapa *level* permainan yang akan dibuat berdasarkan perbedaan penggunaan metode DFS (*Depth-First Search*) :

a. Beginner

Beginner (pemula) adalah tingkatan pada permainan dengan memanfaatkan metode DFS (*Depth-First Search*) terhadap 3 kedalaman penelusuran suatu pohon pencarian yang terbentuk.

b. Intermediate

Intermediate (menengah) adalah tingkatan pada permainan dengan memanfaatkan metode penelusuran DFS (*Depth-First Search*) terhadap 6 kedalaman penelusuran suatu pohon pencarian yang terbentuk.

c. Expert

Expert (mahir) adalah tingkatan pada permainan dengan memanfaatkan metode penelusuran DFS (*Depth-First Search*) terhadap 9 kedalaman penelusuran suatu pohon pencarian yang terbentuk.

Jadi suatu tingkatan permainan dapat mempengaruhi langkah dari *Artificial Intelligence* (AI) karena suatu kecerdasan dalam bermain suatu *game playing* ditentukan oleh seberapa dalam memperhitungkan kemungkinan-kemungkinan

langkah yang terjadi dan dapat mengambil langkah terbaik dari kemungkinan tersebut.

3.2. Metodologi Penelitian

Metode penelitian merupakan suatu mekanisme dan teknik untuk mencari, memperoleh, mengumpulkan, atau mencatat data yang dapat digunakan untuk keperluan menyusun penelitian.

3.2.1. Metode Penelitian

Pada penelitian kali ini penulis akan menggunakan metode penelitian tindakan (*Action Research*).

Metode penelitian tindakan merupakan metode penelitian yang bertujuan untuk mengembangkan keterampilan-keterampilan baru, cara pendekatan baru, atau produk pengetahuan yang baru dan untuk memecahkan masalah dengan penerapan langsung di lapangan (Narimawati, 2008).

Oleh karena definisi di atas, penulis memperoleh gambaran mengenai perancangan aplikasi berbasis *mobile* yang dapat menerapkan algoritma minimax pada permainan reversi.

3.2.2. Jenis dan Metode Pengumpulan Data

Jenis data adalah mengenai dari mana data diperoleh. Apakah data diperoleh dari sumber langsung (data primer) atau data diperoleh dari sumber tidak langsung (data sekunder). Sedangkan Metode Pengumpulan Data merupakan teknik atau cara yang dilakukan untuk mengumpulkan data. Metode menunjuk suatu cara sehingga dapat diperlihatkan penggunaannya melalui angket, wawancara, pengamatan, tes, dkoumentasi dan sebagainya.

3.2.2.1. Jenis Data

Jenis data pada penelitian kali ini hanya berupa data sekunder yaitu berupa data-data hasil penelitian yang telah ada sebelumnya, berupa fakta-fakta, dan data-data hasil studi pustaka dari berbagai buku mengenai objek penelitian.

3.2.2.2. Metode Pengumpulan Data

Untuk metode pengumpulan data yang dilakukan pada penelitian ini penulis hanya menggunakan metode pengumpulan data berupa studi pustaka dari beberapa literatur, referensi, dan buku-buku terkait dengan objek penelitian diatas.

3.2.3. Metode Pendekatan dan Pengembangan Perangkat Lunak

Pendekatan yang dilakukan untuk pengembangan dilakukan untuk menggantikan atau membuat sistem / perangkat lunak / aplikasi. Dalam hal ini terdapat metode yang digunakan untuk melakukan pendekatan dan pengembangan terhadap penelitian ini.

3.2.3.1. Metode Pendekatan Perangkat Lunak

Metode pendekatan perangkat lunak yang digunakan penulis dalam membangun perangkat lunak di sini yaitu menggunakan metode pendekatan evolusioner. Alasannya karena dengan metode pendekatan ini penulis dapat fokus menentukan *rules* atau aturan-aturan pada *artificial intelligence* yang akan dibuat dan kemudian baru dilakukan pembuatan aplikasi *artificial intelligence* berdasarkan *rules* tersebut.

3.2.3.2. Metode Pengembangan Perangkat Lunak

Ada salah satu tipe atau jenis yang sering dipakai pada metode pengembangan evolusioner yaitu metode pengembangan pemrograman evolusioner. Dalam tipe ini tujuan proses adalah untuk mengetahui kebutuhan-kebutuhan dan mengembangkan definisi kebutuhan yang lebih baik untuk sistem. Pemrograman evolusioner sulit untuk membuat spesifikasi sistem secara rinci, oleh karena itu pemrograman evolusioner banyak digunakan dalam pengembangan sistem kecerdasan buatan yang menyamai kemampuan manusia.



Gambar 3.7. Model Pengembangan Evolusioner

(Sumber : Adi Kristianto, 2003)

Berikut penjelasan dari gambar 3.7 diatas :

Pada tahapan awal yang dilakukan yaitu dengan membuat deskripsi secara garis besar atau penjelasan secara garis besar tentang sistem kecerdasan buatan

yang akan dibuat, pada tahapan ini mencakup di dalamnya penentuan *rules* pada sistem kecerdasan buatan.

Kemudian tahapan yang paling penting yaitu spesifikasi, pengembangan, dan validasi yang merupakan aktivitas berulang sampai tercapainya perangkat lunak yang menjawab kebutuhan *user*.

3.2.3.3. Alat Bantu Analisis dan Perancangan

3.2.3.3.1. *Flow Chart*

Flow Chart merupakan penggambaran alur proses suatu program yang diwakili dengan bagan-bagan atau *shapes* tertentu. Dalam kasus ini *Flow Chart* digunakan untuk menggambarkan alur algoritma yang diterapkan pada aplikasi reversi dan alur kerja dari aplikasi secara keseluruhan.

3.2.4. Pengujian Software

Adapun teknik pengujian yang dilakukan terhadap perangkat lunak yaitu menggunakan teknik pengujian *Black Box* yaitu dengan menguji fungsi-fungsi pada perangkat lunak, sedangkan untuk pengujian algoritma dalam hal penerapannya pada perangkat lunak menggunakan teknik pengujian *White Box*.