

PENERAPAN METODE HEURISTIC DALAM PEMBUATAN GAME PING PONG

Yenny Erwinda (0911009)

Mahasiswa Jurusan Teknik Informatika STMIK Budi Darma Medan
Jl. Sisingamangaraja No. 338 Simpang Limun Medan
www.stmik-budidarma.ac.id // Email: yennyerwinda@gmail.com

ABSTRAK

Semakin majunya perkembangan teknologi khususnya dalam pembuatan game yang telah diciptakan oleh generasi penerus. Diantara kecerdasan buatan (Artificial Intelligence AI) yang didefinisikan sebagai kecerdasan yang di tunjukkan oleh suatu entitas buatan.

Kecerdasan diciptakan dan dimasukkan kedalam suatu mesin (Komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain adalah system pakar dan permainan computer (game).

Dalam skripsi ini penulis mencoba membuat sebuah permainan (game) yang dimasukkan unsure kecerdasan buatan didalamnya. Kecerdasan buatan ini diharapkan dapat membantu generasi penerus dalam pembuatan (game).

Kata Kunci : *Heuristic, Game*

1. Pendahuluan

1.1. Latar Belakang Masalah

Kecerdasan Buatan (*Artificial Intelligence* atau *AI*) didefinisikan sebagai kecerdasan yang ditujukan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (*games*), *Logika fuzzy*, syarat tiruan dan robotika.

Banyak hal yang kelihatan sulit untuk kecerdasan manusia, tetapi untuk Informatika relatif tidak bermasalah. Seperti contoh: mentransformasikan persamaan, menyelesaikan persamaan integral, membuat permainan catur dan Backgammon. Di sisi lain, hal yang bagi manusia kelihatannya menuntut sedikit kecerdasan, sampai sekarang masih sulit untuk direalisasikan dalam Informatika. Seperti contoh pengenalan obyek/muka, bermain sepak bola.

Dan bagi para pemakai komputer, rasanya tidak ada yang tidak mengenal dengan apa yang disebut dengan permainan. Banyak sekali program permainan komputer yang beredar di pasaran mulai dari yang sederhana sampai yang kelas tinggi sehingga harus berfikir untuk melakukan permainan tersebut.

Perkembangan program permainan komputer sudah demikian pesatnya dengan program yang makin lama canggih yang disesuaikan dengan perkembangan perangkat keras dan perangkat lunak komputer.

Metode *heuristic* digunakan sebagai metode untuk pencairian posisi ping-pong pada papan atau area bermain ping pong, dengan diterapkannya

metode ini proses pemeriksaan posisi ping pong apakah masih dalam area permainan atau tidak.

Oleh karena itu penulis mencoba membuat sebuah permainan yang dimasukkan unsur Kecerdasan Buatan didalamnya. Banyak orang setelah bermain dengan komputer, bertanya-tanya bagaimana program permainan itu dapat dibuat. Namun dengan memanfaatkan *Borland Delphi 7*, dengan menggunakan bahasa pemrograman *Borland Delphi* diharapkan bisa membuat berbagai permainan komputer.

1.2. Perumusan Masalah

Berdasarkan uraian pada latar belakang diatas maka yang menjadi perumusan masalah adalah

1. Bagaimana mengenal bentuk permainan ping pong?
2. Bagaimana menerapkan metode heuristic dalam permainan ping pong?
3. Bagaimana merancang aplikasi permainan ping pong?

1.3. Batasan Masalah

Proses pencarian ruang keadaan tidak cukup untuk mengotomatiskan tingkah laku pemecahan masalah, pada setiap situasi tersebut hanya terdapat sejumlah pilihan langkah yang terbatas yang boleh dilakukan oleh dua orang pemain. Maka dalam penulisan skripsi ini akan dibahas beberapa hal antara lain :

1. Metode yang digunakan adalah heuristic.
2. Penerapan heuristic hanya sebatas pada pemeriksaan posisi ping pong pada papan permainan.
3. Bahasa pemrograman yang digunakan adalah *Borland Delphi*

1.4. Tujuan dan Manfaat Penelitian

Adapun tujuan dari studi ini yang dilakukan pada pembuatan skripsi ini adalah:

1. Untuk mengenal bentuk permainan ping pong
2. Untuk menerapkan metode heuristic dengan BFS pada dalam permainan ping pong
3. Untuk merancang aplikasi permainan ping pong

Manfaat yang diperoleh penulis selama melakukan penelitian dan penyusunan skripsi ini adalah :

1. Mempelajari dan menerapkan ilmu yang didapat selama bangku kuliah.
2. Mampu membuat rancangan algoritma untuk permasalahan pemrograman berdasarkan beberapa strategi rancangan seperti: iteratif, dan *Heuristic*.
3. Untuk mengembangkan metode dan system untuk menyelesaikan masalah, masalah yang biasa diselesaikan melalui aktivitas intelektual manusia.

1.5. Metode Penelitian

Dalam pelaksanaan skripsi ini aktivitas yang dilakukan didalamnya yaitu mengadakan eksplorasi terhadap perangkat dan konsep yang akan digunakan dalam pembangunan sistem ini, melakukan analisis terhadap permasalahan yang ada, melakukan perancangan sistem berdasarkan hasil analisis tersebut, melakukan implementasi sistem tersebut dengan perangkat yang telah ditentukan dan yang terakhir adalah mengadakan testing terhadap sistem tersebut.

Langkah-langkah yang digunakan dalam penelitian ini adalah:

1. Eksplorasi. Pada tahap ini dilakukan eksplorasi terhadap beberapa perangkat dan konsep yang akan digunakan dalam membuat skripsi ini. Eksplorasi dilakukan pada beberapa perangkat yang akan digunakan untuk membangun sistem dalam skripsi ini seperti *Borland Delphi 7*, Eksplorasi konsep dilakukan dengan cara studi literatur yaitu dengan studi dari berbagai macam buku teks, diktat kuliah, jurnal, karya tulis ilmiah, skripsi dan tesis yang berkaitan dengan masalah yang akan dibahas.
2. Analisis Sistem. Pada tahap ini dilakukan analisis terhadap rumusan masalah dan batasan yang ada dalam skripsi ini. Analisis ini juga dilakukan untuk melakukan analisis spesifikasi sistem yang akan dibuat sesuai dengan batasan yang ada.
3. Perancangan Sistem. Pada tahap ini dilakukan proses perancangan sesuai hasil analisis. Pada tahap perancangan ini dilakukan beberapa perancangan yaitu perancangan arsitektur sistem, perancangan antarmuka, perancangan modul lainnya yang akan berintegrasi dalam suatu sistem.

4. Implementasi Sistem. Pada tahap ini dilakukan implementasi sesuai dengan hasil perancangan. Implementasi ini dilakukan dengan menggunakan perangkat yang sudah dieksplorasi pada tahap sebelumnya. Pada proses implementasi ini dilakukan pembuatan modul-modul dalam bahasa pemrograman tertentu.
5. Testing Sistem. Pada tahap ini dilakukan beberapa tes terhadap sistem yang telah diimplementasikan. Testing dilakukan dengan memasukkan data pengujian tertentu, untuk melihat kesiapan sistem di dunia nyata.

2. Mengatasi *Combinatorial Explosion*

Game ping pong akan dimainkan dalam papan yang berupa kotak-kotak dengan ordo $X \times Y$, supaya pengguna bisa menentukan sendiri orde yang diinginkan. Pada ordo $X \times Y$, 2 kotak akan dipakai oleh titik awal dan titik tujuan. Maksimal penghalang pada tiap ordo dapat kita tentukan yaitu dengan rumus pada tabel 3.1.

Tabel 1 : Maksimal Penghalang

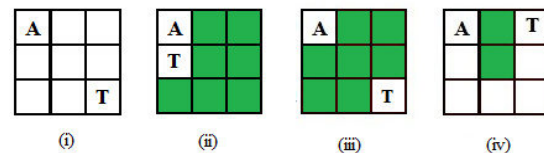
Ordo	Maksimal Penghalang
$X \times Y$	$(M \times N) - 2$
3×3	$(3 \times 3) - 2 = 7$
4×4	$(4 \times 4) - 2 = 14$
5×5	$(5 \times 5) - 2 = 23$
$\dots \times \dots$	$(\dots \times \dots) - 2 = \dots$

Keterangan :

X : Nilai baris

Y : Nilai kolom

Dengan maksimal penghalang $(X \times Y) - 2$, maka akan menghasilkan jalan yang tidak terlihat disebabkan jarak antara titik awal dan titik tujuan saling berdekatan, tapi dengan kondisi titik awal dan titik tujuan saling berjauhan maka akan menghasilkan jalan tidak akan ditemukan, implementasinya dapat dilihat dengan contoh ordo 3×3 pada gambar 3



Gambar 3 : Ruang peta ordo 3×3 dengan tiga kondisi

Maksimal penghalang yang akan dipasang pada ordo lainnya adalah sebanyak ordo tersebut dikurangi dua untuk menempatkan titik awal dan titik tujuan. Pada gambar 3.1 menunjukkan suatu ruang (*map*) dengan ordo 3×3 di dalam *game* yang akan dibangun. Setiap kotak mempresentasikan simpul (*node*). Setiap kotak terhubung ke delapan

kotak yang paling dekat, artinya setiap simpul (*node*) terhubung ke simpul lain yang berada di sebelah kanan, kiri, atas-kanan, bawah-kanan, bawah-kiri, dan atas-kiri dari simpul tersebut. Kotak warna *orange* dan warna hijau diimplementasikan sebagai penghalang, yaitu kotak yang tidak dapat dilalui oleh titik awal.

2.1. Solusi Paling Optimal

Terdapat beberapa hal yang perlu didefinisikan terlebih dahulu dalam kasus *game pingpong* dengan penerapan algoritma heuristik. Adapun istilah-istilah yang akan dibahas yaitu *path*, *open list*, *closed list*, nilai *f*, *g* dan *n*.

Algoritma *heuristic* menggunakan dua senarai yaitu *OPEN* dan *CLOSED OPEN* adalah senarai (*list*) yang digunakan untuk menyimpan simpul-simpul yang pernah dibangkitkan dan nilai heuristiknya telah dihitung tetapi belum terpilih sebagai simpul terbaik (*best node*) dengan kata lain, *OPEN* berisi simpul-simpul masih memiliki peluang untuk terpilih sebagai simpul terbaik, sedangkan *CLOSED* adalah senarai untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah terpilih sebagai simpul terbaik. Artinya, *CLOSED* berisi simpul-simpul yang tidak mungkin terpilih sebagai simpul terbaik (peluang untuk terpilih sudah tertutup).

1. *OPEN LIST* adalah *list* yang menyimpan kemungkinan *path* yang akan diperiksa. *OPEN LIST* dibuat terurut berdasarkan nilai *f*. *OPEN LIST* digunakan untuk menentukan secara selektif (berdasarkan nilai *f*) jalan yang dikira lebih dekat menuju pada *path* tujuan. *OPEN* berisi simpul-simpul yang masih memiliki peluang untuk terpilih sebagai simpul terbaik (*best node*).
2. *CLOSED* adalah senarai (*list*) untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah terpilih sebagai simpul terbaik (*best node*) atau senarai yang menyimpan jalan yang sudah diperiksa dari *open list*. Artinya, *CLOSED* berisi simpul-simpul yang tidak mungkin terpilih sebagai simpul terbaik (peluang untuk terpilih sudah tertutup). Kedua *list* (*OPEN LIST* dan *CLOSED LIST*) ini bertujuan juga untuk menghindari penelusuran berkali-kali jalan (*route*) yang memang sudah diidentifikasi agar tidak masuk kembali ke dalam *OPEN LIST*.
3. Nilai *F* adalah *cost* perkiraan suatu *path* yang teridentifikasi. Nilai *F* merupakan hasil dari *f(n)*.
4. Nilai *G* hasil dari fungsi *g(n)*, adalah banyaknya langkah yang diperlukan untuk menuju ke *path* sekarang.
5. Setiap simpul (*node*) harus memiliki informasi nilai *h(n)*, yaitu estimasi harga simpul tersebut dihitung dari simpul tujuan yang hasilnya menjadi nilai *H*. Fungsi *f* sebagai estimasi fungsi evaluasi

terhadap *node n*, dapat dituliskan :

$$f(n) = g(n) + h(n)$$

dengan :

f(n) = fungsi evaluasi (jumlah *g(n)* dengan *h(n)*)

g(n) = biaya (*cost*) yang dikeluarkan dari keadaan awal sampai keadaan *n*

h(n) = estimasi biaya untuk sampai pada suatu tujuan mulai dari *n*

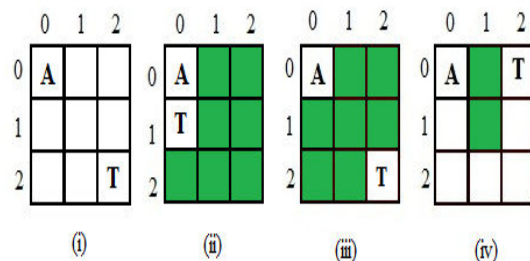
Pergerakan diagonal pada *map* diperbolehkan, maka digunakan fungsi *heuristic Non-Manhattan Distance*. Maka fungsi *heuristic* yang digunakan adalah sebagai berikut :

$$h_diagonal(n) = \min(\text{abs}(n.x - \text{goal}.x) + \text{abs}(n.y - \text{goal}.y))$$

$$h_orthogonal(n) = (\text{abs}(n.x - \text{goal}.x) + \text{abs}(n.y - \text{goal}.y))$$

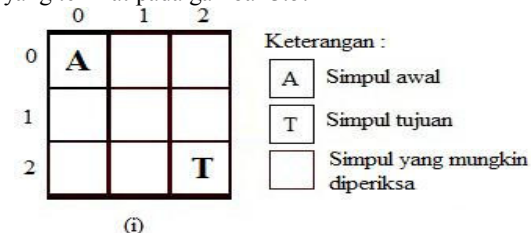
$$h(n) = h_diagonal(n) + (h_orthogonal(n) - (2 * h_diagonal(n)))$$

Sudah dijelaskan pada analisis masalah bahwa ordo dapat disesuaikan dengan rentang *X x Y*, salah satu contoh perhitungan pada *game pingpong* ini akan dijelaskan dengan ordo minimal yaitu *3x3* dikarenakan perhitungan pada ordo berapa pun akan sama. Pada penentuan bobot setiap *node* akan diberikan nilai sesuai dengan jarak terdekat ke tujuan, misal *node* yang terjauh dari tujuan maka diberi bobot yang kecil sedangkan *node* yang terdekat dengan tujuan diberi bobot yang lebih besar. Contoh perhitungannya seperti pada gambar 3.4 yang diberi nilai bobot yang terkecil yaitu 1 dan bobot selanjutnya ditambahkan 1.



Gambar 4 : . Contoh kondisi ruang map yang akan dihitung dengan Heuristik

Perhitungan yang dilakukan dengan algoritma *Heuristik* dengan kondisi tanpa penghalang yang terlihat pada gambar 3.5.



Gambar 5 : Contoh kondisi (i) tanpa penghalang dengan pencarian Heuristik

4. Algoritma

Algoritma untuk merancang perangkat lunak penyelesaian permainan ini menggunakan metode pencarian *heuristic*, fungsi *heuristic* yang digunakan merupakan prakiraan (estimasi) *cost* dari *initial state* ke *goal*, tersebut di bawah ini:

1. Pencarian yang kesuksesannya didasarkan pada pertimbangan harga (didapat dari fungsi *heuristic* tertentu) dari setiap langkah.
2. Pencarian dilakukan dengan melihat satu lintasan, dan memungkinkan untuk berpindah ke lintasan lain.

akan dijelaskan bagaimana menerapkan algoritma pencarian (*heuristic*) dalam sebuah aplikasi. Adapun langkah-langkah perancangan adalah sebagai berikut:

Input

Player_1,
Player_2,
Bola,
Papan_permainan

Output

Skor,
Bola_pingpong

Proses

```
ForeColor = 0
CurrentX = 225
CurrentY = 5
Print "Player 2
"; Format(oldP2, "00")
'print new
ForeColor = FORECOLOR
CurrentX = 225
CurrentY = 5
```

i = 0

While (i < 6000) And (Not entered)

```
i = i + 1
'reset time if key pressed
If iKey <> 0 Then
i = 0
End If
```

If X + BALL_SIZE > BX2 Then

If (Y < p2y0 - MISS_TOLERANCE) Or (Y > p2y1 + MISS_TOLERANCE) Then

```
CheckBallX = MISS1_P2
missedBall = True
```

Else

```
If Y > p2Center + 16 Then
pHitOfs = 6
```

End If

```
If Y <= p2Center + 16 Then
pHitOfs = 5
```

End If

```
If Y <= p2Center + 8 Then
pHitOfs = 4
```

End If

```
If Y <= p2Center Then
pHitOfs = 3
```

End If

If Y <= p2Center - 8 Then

pHitOfs = 2

End If

If Y <= p2Center - 16 Then

pHitOfs = 1

End If

5. Kesimpulan dan Saran

5.1. Kesimpulan

Setelah menyelesaikan aplikasi game Ping Pong dengan menggunakan metode *heuristic*, penulis menarik kesimpulan sebagai berikut:

1. Bentuk permainan ping pong dirancang dengan menggunakan bahasa pemrograman Borland Delphi
2. Permainan ping pong yang dirancang memiliki *interface* yang mirip dengan permainan ping pong yang sebenarnya
3. Penerapan metode *heuristic* dalam permainan ping pong diterapkan pada proses bola yang dipukul oleh pemain

5.2. Saran

Penulis ingin memberikan beberapa saran yang mungkin dapat membantu dalam pengembangan perangkat lunak ini yaitu:

1. Aplikasi dapat dikembangkan dengan menambahkan beberapa metode pencarian lain yang terdapat di dalam ruang lingkup *Artificial Intelligence* (AI).
2. Aplikasi dapat dikembangkan dengan animasi pemain yang digerakkan. Untuk animasi yang lebih baik, perangkat lunak dapat dibangun dengan menggunakan aplikasi *Macromedia Flash*.
3. Aplikasi dapat dikembangkan dengan menambahkan pergerakan yang lebih kompleks ketika tidak menemukan lintasan yang paling menjanjikan.

DAFTAR PUSAKA

1. Kusumadewi.S, “**Teknik Dan Aplikasi**” Edisi Kedua, Graha Ilmu Jakarta.
2. Antony Pranata “**Pemrograman Borland Delphi**” Penerbit ANDI, Yogyakarta, Edisi Ketiga.
3. Budi Sutedjo Dharma Oetomo, “**Simbol Flowchart Dan Diagram Aliran Data**” Penerbit ANDI, Yogyakarta, Edisi 2002
4. Zamrony P. Juhara “**Pemrograman Game 3D dengan Directx**” Penerbit ANDI, Yogyakarta.