

PROBLEM OF LETTER ARCHITECTURE COMPLETION USING A * SEARCH METHOD

Heri Santoso

Program Studi Ilmu Komputer FST
Universitas Islam Negeri Sumatera Utara
Herisantoso@uinsu.ac.id

Abstract. *Problems of architecture beams or commonly called Blocks Architecture is one of the classic problems in the study of Artificial Intelligence (AI), the architectural Problems of beams can be obtained from the problems that are all around us and is one of those cases that could be problems solved by using this beam is the stack, a stack problem conditions using push and pop, one of the methods that can be used is the A * Search algorithm, A * describe the process of problem solving in which one must take any decision on any of the best stride and the use of simulation in resolving the problems of architectural beams are expected to give an idea of how the process of the settlement of these problems, simulation is an attempt to copying features, display, and a real system characteristics, the expected results of the application of the algorithm is A * this is showing the form of settlement in the simulation.*

*Keyword: Simulation, Problems Of Beams, A * Algorithm*

Abstrak. Masalah balok arsitektural atau yang biasa disebut Arsitektur Blok adalah salah satu masalah klasik dalam studi Inteligensi Buatan (AI), masalah arsitektur balok dapat diperoleh dari masalah yang ada di sekitar kita dan merupakan salah satu kasus yang bisa menjadi masalah yang dipecahkan dengan menggunakan balok ini adalah tumpukan, kondisi tumpukan masalah menggunakan push dan pop, salah satu metode yang dapat digunakan adalah algoritma pencarian A *, A * menggambarkan proses pemecahan masalah di mana seseorang harus membuat keputusan tentang salah satu langkah terbaik dan penggunaan simulasi dalam menyelesaikan arsitektur masalah balok diharapkan dapat memberikan gambaran tentang bagaimana proses penyelesaian masalah, simulasi adalah upaya untuk menyalin fitur, penampilan, dan karakteristik dari sistem nyata, hasil yang diharapkan dari aplikasi algoritma adalah A * menunjukkan bentuk penyelesaian dalam simulasi.

Kata Kunci : Simulasi, Masalah Balok, Algoritma A

Latar Belakang

Problema arsitektur balok atau biasa disebut *Blocks Architecture* merupakan salah satu persoalan klasik dalam bidang studi *Artificial Intelligence* (AI). Problema ini dapat diilustrasikan seperti berikut, terdapat 2 atau lebih kotak yang disusun menjadi 2 atau 3 tumpukan balok sebagai kondisi awal (*initial state*). Sasaran (*goal*) dari problema ini adalah mendapatkan suatu tumpukan kotak sesuai dengan keinginan. Operasi yang diperbolehkan dalam proses penyelesaiannya yaitu turunkan(x), yang berarti bahwa kotak x diturunkan dari suatu tumpukan tertentu dan letakkan (x, y), yang berarti bahwa kotak x diletakkan di atas kotak y, dengan persyaratan bahwa kotak x dan kotak y harus berada di urutan paling atas dari suatu tumpukan.

Salah satu metode yang bisa digunakan adalah algoritma A^* Search, A^* mendeskripsikan proses pemecahan masalah dimana seseorang harus mengambil setiap keputusan terbaik pada setiap langkahnya. Optimisasi dengan dynamic programming yang diterapkan di A^* dicapai dengan memilih setiap keputusan terbaik (optimal) dalam setiap tahap.

Penggunaan simulasi dalam menyelesaikan problema arsitektur balok huruf diharapkan mampu memberikan gambaran bagaimana proses penyelesaian problema tersebut, simulasi merupakan sebuah usaha untuk menyalin fitur, tampilan, dan karakteristik sebuah sistem nyata. Dalam model ini, akan ditunjukkan bagaimana mensimulasikan bagian sebuah sistem manajemen operasi dengan mengembangkan model matematika paling dekat yang menggambarkan sistem yang sesungguhnya (Soma Purnama Aji, Penerapan Model Simulasi Antrian Multi Channel Single Phase Pada Antrian Di Apotek Purnama Semarang, Sekolah Tinggi Ilmu Ekonomi Widya Manggala).

Berdasarkan uraian di atas, penulis ingin merancang suatu perangkat lunak yang mampu untuk melakukan simulasi terhadap problema *Blocks Architecture*. Oleh karena itu, peneliti mengambil judul untuk penelitian ini "**Penyelesaian Problema Arsitektur Balok huruf Dengan Metode A^* Search**".

Rumusan Masalah

Adapun rumusan masalahnya antara lain

1. Bagaimana merancang sebuah sistem simulasi untuk penyelesaian *block architecture* problema ?
2. Bagaimana menyelesaikan suatu problema dengan menggunakan Metode A^* Search?
3. Bagaimana menerapkan metode A^* pada sistem yang dirancang sebagai solusi permasalahan *block architecture* problema?

Batasan Masalah

Karena keterbatasan waktu dan pengetahuan peneliti, maka ruang lingkup permasalahan dalam merancang perangkat lunak ini antara lain :

1. Jumlah kotak dibatasi maksimal 6 buah dan minimal 3 buah dan setiap kotak diberikan *label* berupa karakter tunggal A sampai dengan H.
2. Jumlah tumpukan balok sebanyak 3 buah tumpukan.

3. *Input* dari perangkat lunak berupa :
 - a. Kondisi awal (*initial state*) berupa posisi awal setiap balok dalam tumpukan tertentu.
 - b. Kondisi akhir (*goal state*) berupa tumpukan balok dalam keadaan tertentu pada satu tumpukan.
4. Metode yang digunakan untuk pencarian solusi adalah metode A* Search

Tujuan Penelitian

Sesuai dengan permasalahan yang telah dirumuskan maka tujuan penelitian ini adalah:

1. Untuk merancang sebuah aplikasi simulasi untuk penyelesaian *block architecture* problema
2. Untuk menyelesaikan suatu problema dengan menggunakan Metode A* Search

Manfaat Penelitian

Sesuai dengan tujuan yang telah dirumuskan maka manfaat penelitian ini adalah sebagai berikut:

1. Mengetahui menyelesaikan suatu problema dalam bentuk visualisasi dan langkah-langkah penyelesaiannya
2. Mengetahui penyelesaian suatu problema dengan menerapkan Metode A* Search
3. Manfaat aplikasi bisa digunakan sebagai tool bantu pembelajaran struktur data
4. Penelitian yang dihasilkan dapat dijadikan sebagai salah satu referensi bagi mahasiswa jurusan Ilmu Komputer Universitas Islam Negeri Sumatera Utara.

Problema Balok Huruf

Problema *balok huruf* merupakan salah satu persoalan klasik dalam bidang studi *Artificial Intelligence (AI)*. Problema ini dapat diilustrasikan seperti berikut, terdapat 2 atau lebih kotak yang disusun menjadi 2 atau 3 tumpukan balok sebagai kondisi awal (*initial state*). Sasaran (*goal*) dari problema ini adalah mendapatkan suatu tumpukan kotak sesuai dengan keinginan. Operasi yang diperbolehkan dalam proses penyelesaiannya yaitu turunkan(x), yang berarti bahwa kotak x diturunkan dari suatu tumpukan tertentu dan letakkan (x, y), yang berarti bahwa kotak x diletakkan di atas kotak y, dengan persyaratan bahwa kotak x dan kotak y harus berada di urutan paling atas dari suatu tumpukan.

Secara umum, *pseudocode* prosedur pencarian pada problema *blocks world architecture* dapat dituliskan seperti berikut ini:

```
Function GSearch(Problem, QueuingFn) : Solution | Failure;  
Var nodes:structure;  
Begin  
  While  
    Begin
```

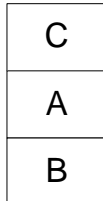
```

Node := remove_front_node(nodes);
IF Goal_test(problem, STATE(node)) succeeds then
    Solution := Node
Else
    Nodes := QueuingFn(node, OPERATOR(problem));
End;
EndWhile
End;

```

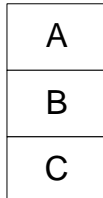
Sebagai contoh, perhatikan kasus sederhana berikut:

a. Keadaan awal:



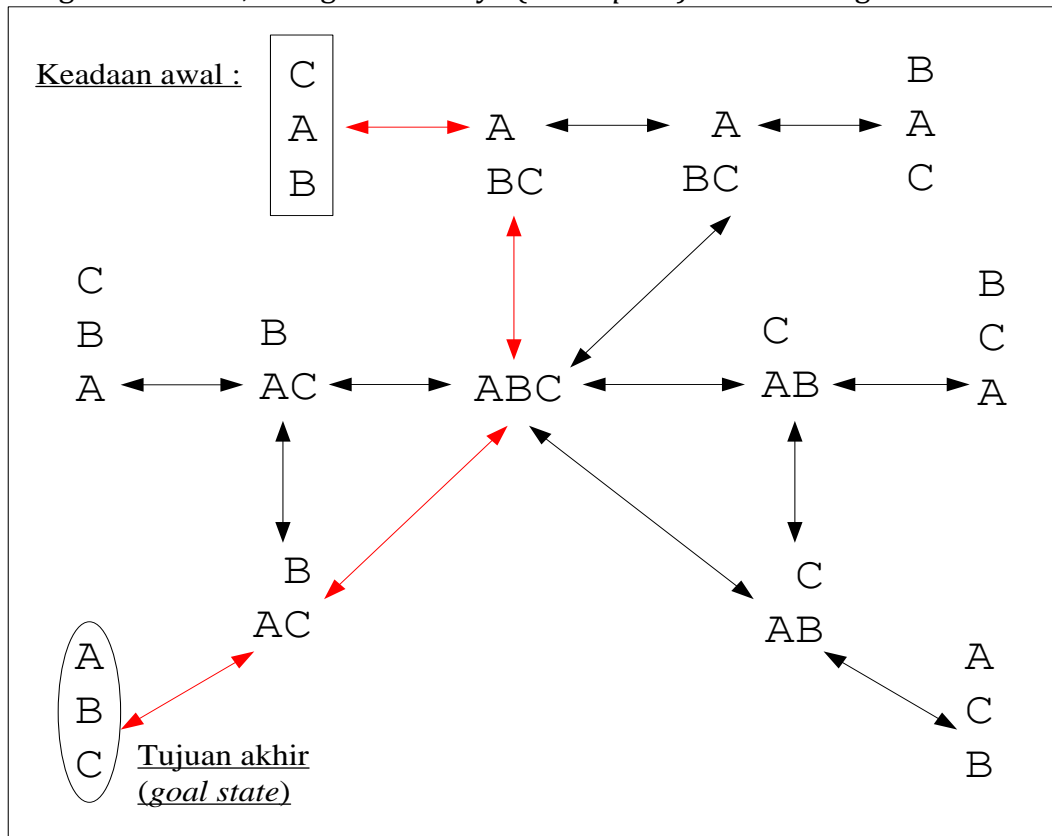
Gambar 2.1 Keadaan awal

b. Tujuan akhir (goal state atau final state) adalah:



Gambar 2.2 Tujuan akhir (goal state)

Dengan demikian, ruang keadaannya (state space) adalah sebagai berikut:



Gambar 2.3 Ruang Keadaan pada Blocks World Architecture

Sumber: T. Sutojo, 2011

Dari gambar 2.3, dapat dilihat bahwa ruang keadaan tersebut mempunyai 13 elemen atau node, dan penyelesaian untuk permasalahan ini adalah anggota dari kumpulan semua lintasan dari keadaan awal hingga tujuan yang lintasannya ditandai dengan garis berwarna merah.

Algoritma A* (A-Star)

Algoritma A* (A-Star) merupakan salah satu dari algoritma yang menerapkan teknik heuristik (teknik pencarian terbimbing). Algoritma A* (A-Star) merupakan salah satu algoritma yang dipelajari untuk menyelesaikan permasalahan. Algoritma A* menyelesaikan masalah yang menggunakan graf untuk perluasan ruang statusnya. Dengan kata lain digunakan untuk menyelesaikan permasalahan yang bisa direpresentasikan dengan graf. Algoritma A* adalah sebuah algoritma yang telah diperkaya dengan menerapkan suatu heuristik, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan.

Algoritma A* membangkitkan simpul yang paling mendekati solusi. Simpul ini kemudian disimpan suksesornya ke dalam list sesuai dengan urutan yang paling mendekati solusi terbaik. Kemudian, simpul pertama pada list diambil, dibangkitkan suksesornya dan kemudian suksesor ini disimpan ke dalam list sesuai dengan urutan yang terbaik untuk solusi. List simpul ini disebut dengan simpul terbuka(open node).

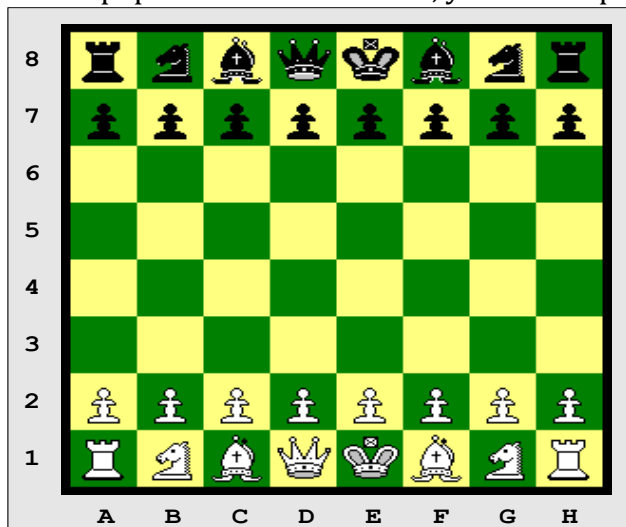
Algoritma A* (A-Star) merupakan perkembangan dari algoritma Best First

Mendefinisikan Masalah Sebagai Suatu Ruang Keadaan

Misalkan permasalahan yang akan dihadapi adalah "Permainan Catur", maka harus ditentukan:

1. Posisi awal pada papan catur;

Posisi awal setiap permainan catur selalu sama, yaitu semua bidak diletakkan di atas papan catur dalam 2 sisi, yaitu kubu putih dan kubu hitam.

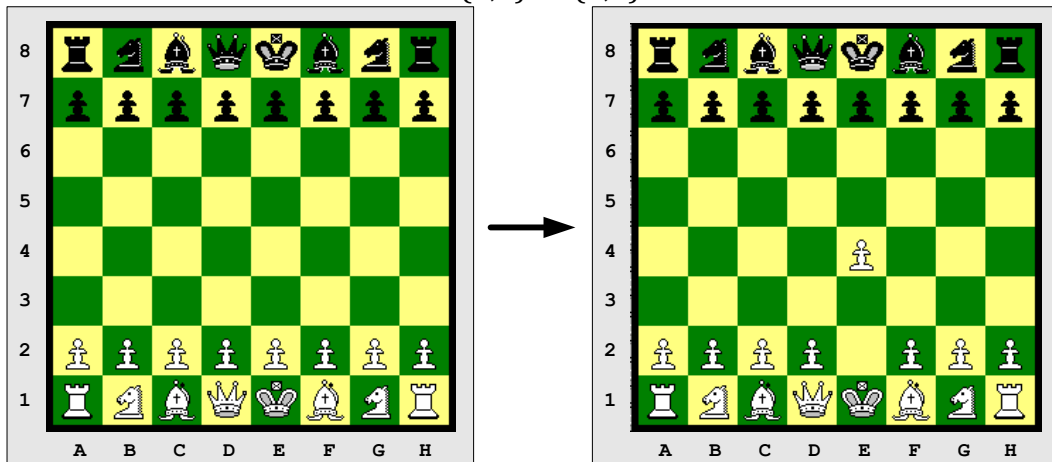


Gambar 2.5 Keadaan awal permainan catur

Sumber: T. Sutojo, 2011

2. Aturan-aturan untuk melakukan gerakan secara legal;
Aturan-aturan ini sangat berguna untuk menentukan gerakan suatu bidak, yaitu melangkah dari satu keadaan ke keadaan lain. Misalkan untuk mempermudah menunjukkan posisi bidak, setiap kotak harus ditunjukkan dalam huruf (A, B, C, D, E, F, G, H) pada arah horizontal dan angka (1, 2, 3, 4, 5, 6, 7, 8) pada arah vertikal. Suatu aturan untuk menggerakkan bidak dari posisi (E,2) ke (E,4) dapat ditunjukkan dengan aturan:

IF Bidak putih pada Kotak(E,2),
And Kotak(E,3) Kosong,
And Kotak(E,4) Kosong,
Then Gerakkan bidak dari (E,2) ke (E,4)

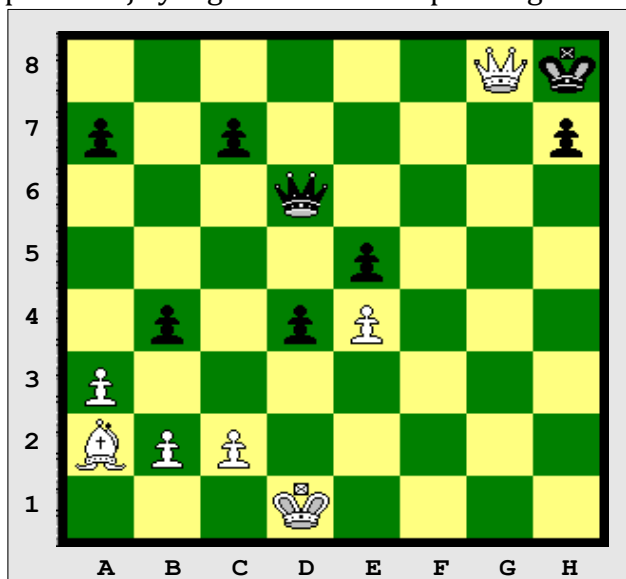


Gambar 2.6 Gerakan bidak catur

Sumber: T. Sutojo, 2011

3. Tujuan (*goal*)

Tujuan yang ingin dicapai adalah posisi pada papan catur yang menunjukkan kemenangan seseorang terhadap lawannya. Kemenangan ini ditandai dengan posisi Raja yang sudah tidak dapat bergerak lagi.



Gambar 2.7 Tujuan (*goal*) pada permainan catur

Sumber: T. Sutojo, 2011

Contoh di atas menunjukkan representasi masalah dalam Ruang Keadaan (*State Space*), yaitu suatu ruang yang berisi semua keadaan yang mungkin. Kita dapat memulai bermain catur dengan menempatkan diri pada keadaan awal, kemudian bergerak dari satu keadaan ke keadaan yang lain sesuai dengan aturan yang ada, dan mengakhiri permainan jika salah satu pemain telah mencapai tujuan.

Sehingga secara umum, untuk mendekripsikan masalah dengan baik, harus:


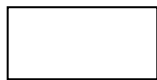

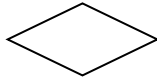

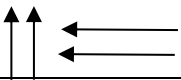

1. Mendefinisikan suatu ruang keadaan.
2. Menetapkan satu atau lebih keadaan awal.
3. Menetapkan satu atau lebih tujuan.
4. Menetapkan kumpulan aturan.

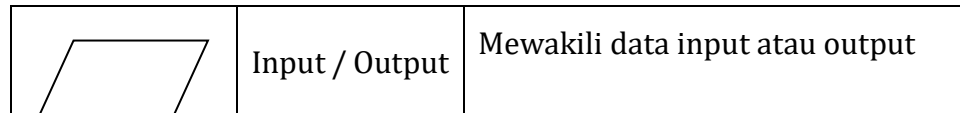
Bagan Alir Program (Flowchart)

Suatu metode untuk menggambarkan tahap – tahap pemecahan masalah adalah dengan mempresentasikan simbol – simbol tertentu yang mudah di mengerti dan standart. Program flowchart adalah salah satu yang dapat didefinisikan sebagai bagan yang menjelaskan secara rinci langkah – langkah dari proses program

(*Jogiyanto, 2007*).

Tabel 2.3 Simbol – Simbol Flowchart Diagram

Simbol	Nama simbol	Keterangan
	Terminal	Untuk menggambarkan awal dan akhir proses aliran dokumen
	Processing	Dipakai untuk pengolahan aritmatika dan pemindahan data
	Preparation	Dipakai untuk memberikan nilai awal dari suatu variable atau counter
	Decision	Dipakai untuk mewakili operasi perbandingan logika
	Predefined process	Dipakai untuk proses yang detailnya djelaskan secara terpisah , misalnya dalam bentuk subroutine
	Garis alir	Dipakai untuk menunjukkan aliran dari program
	Penghubung	Menunjukkan penghubung di halaman yang masih sama atau dihalaman lainnya



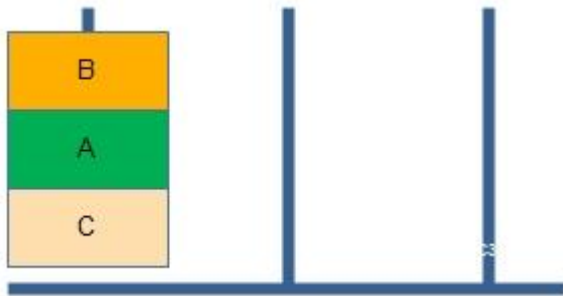
Sumber :Abdul Kadir, 2014

ANALISA DAN PERANCANGAN

Analisa

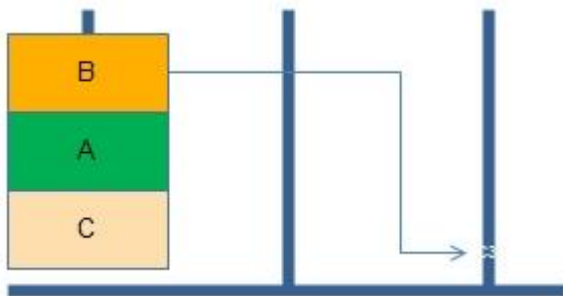
Problema balok huruf merupakan salah satu persoalan klasik dalam bidang studi *Artificial Intelligence* (AI). Problema ini dapat diilustrasikan seperti berikut, terdapat 3 atau lebih kotak yang disusun menjadi 3 tumpukan balok sebagai kondisi awal (*initial state*). Sasaran (*goal*) dari problema ini adalah mendapatkan suatu tumpukan kotak sesuai dengan keinginan. Operasi, aksi dan aturan yang terdapat di dalam problema ini adalah:

1. Kondisi awal dari balok, terdapat 3 buah balok yang diberi nama C,A,B, C merupakan tumpukan paling bawah dan A merupakan tumpukan atas, berikut gambarnya



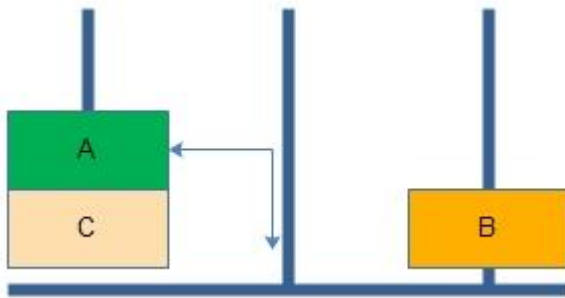
Gambar 3.1 Kondisi Awal

2. Pindahkan balok B dari tiang-1 ke tiang-3. Aturannya adalah balok harus merupakan balok paling atas dari tiang-1 dan menempati posisi paling atas pada tiang-3.



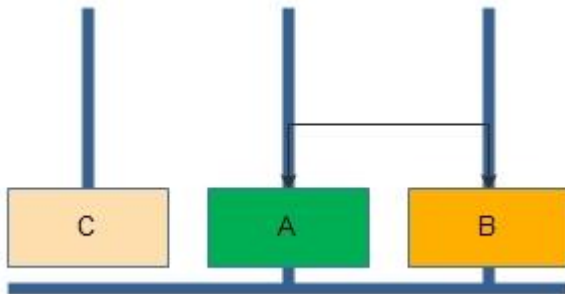
Gambar 3.2 Operasi-1 Pemindahan Balok dari tiang-1 ke tiang-3

3. Pindahkan balok A dari tiang-1 ke tiang-2. Aturannya adalah balok harus merupakan balok paling atas dari tiang-1 dan balok A menempati posisi paling atas pada tiang-2.



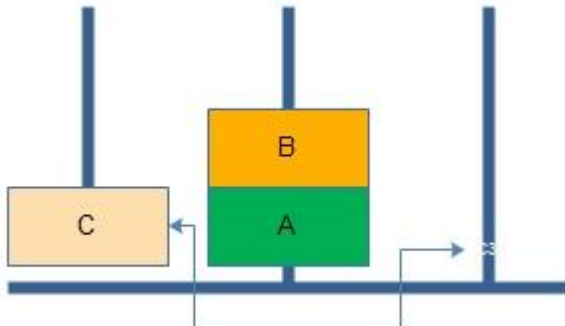
Gambar 3.3 Operasi-2 Pemindahan Balok dari tiang-1 ke tiang-2

4. Pindahkan balok B dari tiang-3 ke tiang-2. Aturannya adalah balok B harus merupakan balok paling atas dari tiang-3 dan balok B menempati posisi paling atas pada tiang-2.



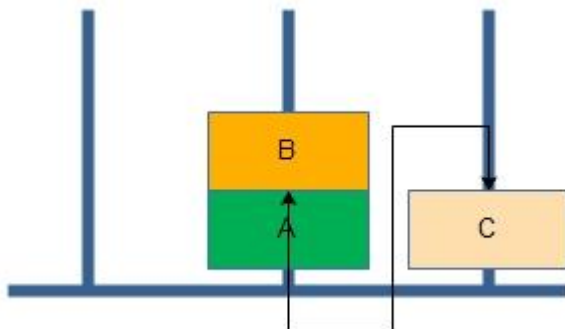
Gambar 3.4 Operasi-3 Pemindahan Balok dari tiang-3 ke tiang-2

5. Pindahkan balok C dari tiang-1 ke tiang-3. Aturannya adalah balok C harus merupakan balok paling atas dari tiang-1 dan balok C menempati posisi paling atas pada tiang-3.



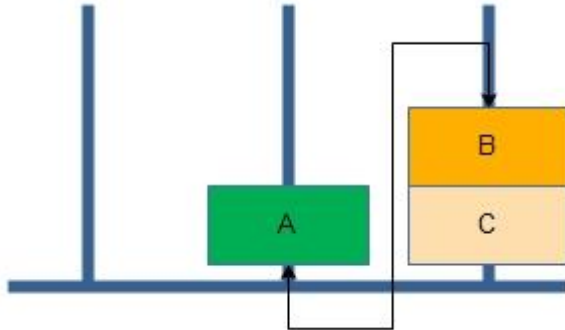
Gambar 3.5 Operasi-5 atau Aksi-5

6. Pindahkan balok B dari tiang-2 ke tiang-1. Aturannya adalah balok B harus merupakan balok paling atas dari tiang-2 dan balok B menempati posisi paling atas pada tiang-1.



Gambar 3.6 Operasi-6 atau Aksi-6

7. Pindahkan balok B dari tiang-2 ke tiang-3. Aturannya adalah balok B harus merupakan balok paling atas dari tiang-2 dan balok A menempati posisi paling atas pada tiang-3.



Gambar 3.7 Operasi-7 atau Aksi-7

Setelah dipindahkan maka hasilnya adalah seperti berikut ini



Gambar 3.8 Hasil

Analisis Pemecahan Masalah dengan Algoritma A*

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

Algoritma A* 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).

Sudah dijelaskan pada analisis masalah bahwa ordo dapat disesuaikan dengan rentang X x Y. 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.

Prosedur pencarian A* merupakan pencarian yang dilakukan dengan mengunjungi tiap-tiap *node* secara sistematis pada setiap level hingga keadaan tujuan (*goal state*) ditemukan. Atau dengan kata lain, penelusuran dilakukan

dengan mengunjungi *node-node* per level hingga ditemukan *goal state*. Apabila terdapat solusi, pencarian A^* menjamin ditemukannya solusi dengan lintasan terpendek (*shortest path*). Penerapan algoritma A^* dapat dilihat pada contoh berikut. $C1=C$, $C2=A$, $C3=B$

1. State awal I:
 - i. Tumpukan-1: C, A,B (Balok B berada di atas balok A dan C).
 - ii. Tumpukan-2: -
 - iii. Tumpukan-3: -
2. State awal II:
 - i. Tumpukan-1: C, A(Balok A berada di atas balok C)
 - ii. Tumpukan-2: -
 - iii. Tumpukan-3: B
3. State awal III:
 - i. Tumpukan-1: C.
 - ii. Tumpukan-2: A.
 - iii. Tumpukan-3: B.
4. State awal IV:
 - i. Tumpukan-1: C.
 - ii. Tumpukan-2: A,B(Balok B berada di atas balok A)
 - iii. Tumpukan-3: -
5. State awal V:
 - i. Tumpukan-1: -.
 - ii. Tumpukan-2: A,B(Balok B berada di atas balok A)
 - iii. Tumpukan-3: C.
6. State awal VI:
 - i. Tumpukan-1: -.
 - ii. Tumpukan-2: A
 - iii. Tumpukan-3: B,C (Balok B berada di atas balok C)
7. State awal VII:
 - i. Tumpukan-1: -
 - ii. Tumpukan-2: -
 - iii. Tumpukan-3: A,B,C(Balok A berada di atas balok B dan C)

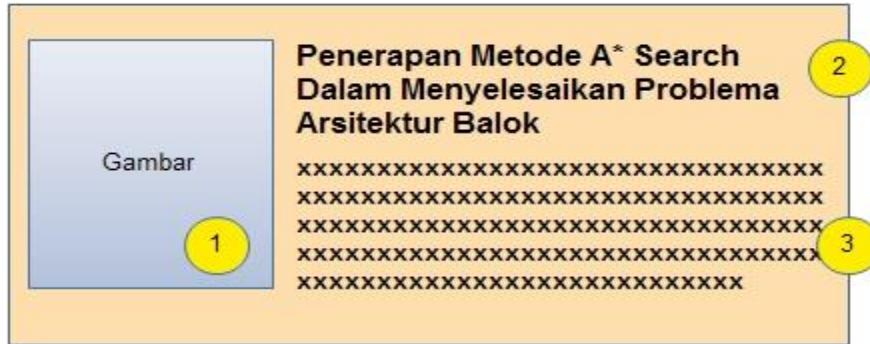
Perancangan

Perancangan perangkat lunak ini menggunakan bahasa pemrograman *Microsoft Visual Basic Net*. Perangkat lunak ini memiliki beberapa *form*, antara lain:

1. *Form Splash Screen*
2. *Form Utama*

Form Splash Screen

Form ini berisi nama perangkat lunak, nama pembuat perangkat lunak, nama kampus, jurusan dan program studi. *Form* ini muncul pertama kali ketika program dieksekusi.



Gambar 3.10 Form Splash Screen

Keterangannya:

1. Menampilkan gambar dari objek balok
2. Menampilkan informasi judul
3. Menampilkan informasi penulis

Form Utama

Fungsi dari form ini adalah untuk menampilkan posisi awal balok dan posisi tujuan dari balok, berikut adalah desainnya



Gambar 3.11 Form Utama

Keterangannya:

1. frame yang digunakan untuk menampilkan posisi awal dari susunan balok
2. frame yang digunakan untuk menentukan posisi akhir dari susunan balok
3. tombol yang digunakan untuk mencari solusi penyelesaian
4. tombol keluar

KESIMPULAN DAN SARAN

Kesimpulan

Setelah menyelesaikan perangkat lunak problema balok huruf menggunakan algoritma A*, penulis menarik kesimpulan sebagai berikut:

1. Persoalan penyelesaian problema balok huruf dengan menggunakan algoritma A* menjamin ditemukannya solusi untuk setiap *state* yang ada.
2. Sistem yang dirancang merupakan penerapan dari algoritma A*
3. Pada balok huruf yang menjadi *goal state*, posisi balok huruf bebas ditentukan pada tumpukan 1, tumpukan 2 dan tumpukan 3 tidak hanya pada satu tumpukan saja.
4. Penerapan bahasa pemrograman visual basic.net dalam pembuatan aplikasi menunjang pembuatan aplikasi lebih cepat.

DAFTAR PUSTAKA

- [1] Firman Harianja, *Penerapan Algoritma A* Pada Permasalahan Optimalisasi Pencarian Solusi Dynamic Water Jug*, Jurnal Pelita Informatika, ISSN: 2301—9425
- [2] Rengga Dionata Putra, *Pencarian Rute Terdekat Pada Labirin Menggunakan Metode A**, Jurnal EECCIS Vol. 6, No. 2, Desember 2012
- [3] Rudy Adipranata, *Aplikasi Pencari Rute Optimum Pada Peta Guna Meningkatkan Efisiensi Waktu Tempuh Pengguna Jalan Dengan Metode A* Dan Best First Search*, PUSLIT Petra
- [4] Soma Purnama Aji, *Penerapan Model Simulasi Antrian Multi Channel Single Phase Pada Antrian Di Apotek Purnama Semarang*, 2011, Jurnal Widya Manggala
- [5] T. Sutojo, S.Si.,M.Kom, *Kecerdasan Buatan*, Penerbit Andi, 2011, Yogyakarta
- [6] Veronica Mutiana, *Optimasi Pencarian Jalur dengan Metode A-Star*, ULTIMATICS, Vol. V, No. 2 Desember 2013, ISSN 2085-4552
- [7] Wahana Komputer, *Shortcourse Series : Visual Basic 2010 Programming*, Penerbit Andi, 2012
- [8] Yulikus Partono, S.Kom, *Pengantar Logika dan Algoritma*, Penerbit Andi, 2010