

# Rancang Bangun Simulasi Langkah Kuda Dalam Papan Catur

**Amri Saputra, ST, Elin Haerani, ST. M.Kom**

Teknik Informatika Fakultas Sains dan Teknologi UIN Sultan Syarif Kasim Riau  
Jl. H.R. Subrantas KM. 15 no. 155 Simpang Baru- Panam  
Pekanbaru – Riau  
Email : elin\_haerani@yahoo.com.sg

## ABSTRAK

Biji catur kuda dalam papan catur memiliki pergerakan menyerupai huruf L dan juga dapat melangkah walaupun terdapat bidak/pion di depannya. Biji catur ini merupakan salah satu biji yang sangat sulit digerakkan dan sering juga merupakan biji yang paling berbahaya apabila tidak diperhatikan secara seksama setiap pergerakannya. Simulasi dari permasalahan ini menyediakan sebuah papan catur berukuran 8 x 8. Sasaran (goal) dari permasalahan ini adalah menggerakkan sebuah biji kuda dari suatu posisi tertentu pada papan catur ke posisi tujuan yang diinginkan dengan mensimulasikan semua solusi pergerakan terpendek yang mungkin untuk menuju ke posisi tujuan tersebut. Penyelesaian permasalahan ini dapat menggunakan bantuan sistem produksi dan menggunakan metode pencarian melebar pertama (Breadth First Search). Aplikasi mensimulasikan semua gerakan dari semua solusi yang ditemukan, sehingga memberikan gambaran yang cukup jelas atas solusi-solusi yang dihasilkan

**Kata Kunci:** Catur, Simulasi, BFS

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Catur adalah permainan mental yang dimainkan oleh dua orang. Pecatur adalah orang yang memainkan catur, baik dalam pertandingan satu lawan satu maupun satu melawan banyak orang (dalam keadaan informal). Kata catur diambil dari bahasa Sanskerta yang berarti "empat", namun kata ini sebenarnya merupakan singkatan dari caturangga yang berarti empat sudut. Di India kuno permainan catur memang dimainkan oleh empat peserta yang berada di empat sudut yang berbeda. Hal ini lain dari permainan catur modern di mana pesertanya hanya dua orang saja, kemudian kata caturangga ini diserap dalam bahasa Persia menjadi shatranj. Kata chess dalam bahasa Inggris diambil dari bahasa Persia shah.

Biji catur terdiri dari Raja, Ratu, Gajah/Rencong, Kuda, Benteng dan Pion. Dari sekian banyak biji catur, biji catur kuda mempunyai langkah yang lain dari biji catur lainnya. Biji catur kuda dalam papan catur memiliki pergerakan menyerupai huruf L dan juga dapat melangkah walaupun terdapat bidak/pion di depannya. Biji catur ini merupakan salah satu biji yang sangat sulit digerakkan dan sering juga merupakan biji yang paling berbahaya apabila tidak diperhatikan secara seksama setiap pergerakannya. Simulasi dari permasalahan ini menyediakan sebuah papan catur berukuran 8 x 8. Sasaran (goal) dari permasalahan ini adalah menggerakkan sebuah biji kuda dari suatu posisi tertentu pada papan catur ke posisi tujuan yang diinginkan dengan mensimulasikan semua solusi pergerakan terpendek yang mungkin untuk menuju ke posisi tujuan tersebut. Permasalahan ini juga merupakan salah satu masalah klasik dalam *artificial intelligence* (AI). Penyelesaian permasalahan ini dapat menggunakan bantuan sistem produksi dan menggunakan metode pencarian melebar pertama (Breadth First Search).

Berdasarkan uraian di atas, penulis ingin merancang suatu aplikasi yang mampu mencari semua solusi pergerakan terpendek sebuah biji kuda dari posisi awal menuju posisi tujuan yang bersifat dinamis dan menampilkan simulasi pergerakan dari kuda dalam papan catur. Oleh karena itu, penulis mengambil penelitian dengan judul “Rancang Bangun Simulasi Langkah Kuda dalam Papan Catur”

### 1.2 Rumusan Masalah

Adapun masalah yang akan dibahas dalam penelitian ini yaitu :

- Mencari dan menampilkan semua solusi pergerakan terpendek yang dapat dilewati biji kuda dari posisi awal ke posisi tujuan.
- Mensimulasikan semua solusi pergerakan yang dihasilkan.

### 1.3 Batasan Masalah

Ruang lingkup permasalahan dalam merancang aplikasi ini antara lain :

- Ukuran dari papan catur 8 x 8
- Komponen-komponen yang terdapat pada aplikasi, yaitu:
  - Kuda (putih), sebagai simbol dari posisi awal.
  - Raja (hitam), sebagai simbol dari posisi tujuan.
  - Bidak/Pion (hitam), sebagai simbol dari rintangan yang tidak boleh ditempati. Bidak / rintangan dibatasi maksimal sebanyak 8 buah.

3. Metode pencarian yang digunakan adalah dengan menggunakan metode pencarian melebar pertama (*Breadth-First Search*).
4. Pencarian hanya akan menemukan semua solusi pergerakan terpendek.
5. Aplikasi yang akan dirancang bukan suatu permainan/ *game*, hanya memperlihatkan langkah kuda dari posisi awal hingga mencapai posisi tujuan (raja)

#### 1.4 Tujuan Penelitian

Tujuan penyusunan penelitian ini adalah untuk merancang suatu aplikasi yang mampu mencari semua solusi pergerakan terpendek untuk menuju posisi tujuan dengan menggunakan bantuan sistem produksi dan menggunakan metode BFS (*Breadth-First Search*)

## 2. TINJAUAN PUSTAKA

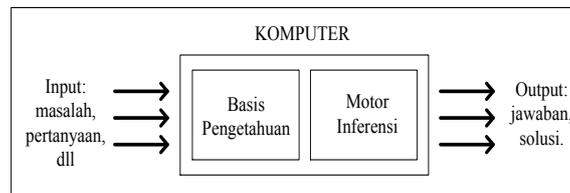
### 2.1 Definisi AI

Kecerdasan buatan atau AI adalah salah satu bagian ilmu komputer yang mempelajari tentang bagaimana cara membuat agar komputer dapat melakukan pekerjaan seperti yang dilakukan oleh manusia. Tujuan dari AI adalah untuk memecahkan persoalan dunia nyata (bersifat praktis) dan memahami inteligensi (bersifat memahami). (Sandi.S, 1993)

AI merupakan salah satu bagian ilmu komputer yang mempelajari tentang bagaimana cara membuat agar komputer dapat melakukan pekerjaan seperti yang dilakukan oleh manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan zaman, maka peran komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia. (Kusumadewi.S, 2002)

Untuk menciptakan aplikasi AI ada 2 bagian utama yang sangat dibutuhkan, yaitu:

1. Basis Pengetahuan (*Knowledge Base*), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan yang lainnya.
2. Motor Inferensi (*Inference Engine*), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman. (Kusumadewi.S, 2002)

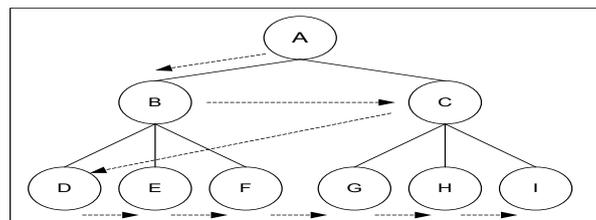


Gambar 2.1 Penerapan konsep AI di dalam computer

### 2.2 Pencarian Melebar Pertama (*Breadth First Search*)

*Breadth First Search* adalah algoritma yang melakukan pencarian secara melebar yang mengunjungi suatu simpul kemudian mengunjungi semua simpul yang bertetangga dengan simpul tersebut terlebih dahulu (Erdi Wijaya, 2005). Algoritma ini memerlukan sebuah antrian  $q$  untuk menyimpan simpul yang telah dikunjungi. Simpul-simpul ini diperlukan sebagai acuan untuk mengunjungi simpul-simpul yang bertetangga dengannya. Tiap simpul yang telah dikunjungi masuk ke dalam antrian hanya satu kali. Algoritma ini juga membutuhkan *table Boolean* untuk menyimpan simpul yang telah dikunjungi sehingga tidak ada simpul yang dikunjungi lebih dari satu kali.

Metode *Breadth-First Search*, semua *node* pada level  $n$  akan dikunjungi terlebih dahulu sebelum mengunjungi *node-node* pada level  $n+1$ . Pencarian dimulai dari *node* akar terus ke level ke-1 dari kiri ke kanan, kemudian berpindah ke level berikutnya. Demikian seterusnya hingga ditemukannya solusi.



Gambar 2.3 Metode *Breadth First Search*

### 2.3 Asal Usul Permainan Catur

Catur adalah permainan mental yang dimainkan oleh dua orang. Pecatur adalah orang yang memainkan catur, baik dalam pertandingan satu lawan satu maupun satu melawan banyak orang (dalam keadaan informal). Sebelum bertanding, pecatur memilih biji catur yang akan ia mainkan. Terdapat dua

warna yang membedakan bidak atau biji catur, yaitu hitam dan putih. Pemegang buah putih memulai langkah pertama, yang selanjutnya diikuti oleh pemegang buah hitam secara bergantian sampai permainan selesai. Kata catur diambil dari bahasa Sanskerta yang berarti "empat". Namun kata ini sebenarnya merupakan singkatan dari *caturangga* yang berarti empat sudut. Di India kuno permainan catur memang dimainkan oleh empat peserta yang berada di empat sudut yang berbeda. Hal ini lain dari permainan catur modern di mana pesertanya hanya dua orang saja. Kemudian kata *caturangga* ini diserap dalam bahasa Persia menjadi *shatranj*. Kata *chess* dalam bahasa Inggris diambil dari bahasa Persia *shah*.

### 3. METODOLOGI PENELITIAN

Metodologi penelitian merupakan pedoman dalam pelaksanaan penelitian sehingga yang dicapai tidak menyimpang dari tujuan yang telah ditentukan sebelumnya. Berikut gambaran langkah-langkah secara garis besar dalam penelitian ini :

- a. Pembelajaran literatur
- b. Pengumpulan data
- c. Analisa dan perancangan
- d. Implementasi dan pengujian
- e. Kesimpulan dan saran

### 4. ANALISA DAN PERANCANGAN

#### 4.1 Analisa

Permasalahan pergerakan langkah kuda dalam papan catur merupakan salah satu masalah klasik dalam AI. Dalam permainan catur, biji kuda memiliki pergerakan menyerupai huruf L. Biji catur ini merupakan salah satu biji yang paling sulit digerakkan dan sering juga merupakan biji yang paling berbahaya apabila tidak diperhatikan secara seksama setiap pergerakannya. Permasalahan ini menyediakan sebuah papan catur berukuran 8 x 8. Sasaran (*goal*) dari permasalahan ini adalah menggerakkan sebuah biji kuda dari suatu posisi tertentu pada papan catur ke posisi tujuan yang diinginkan dengan mensimulasikan semua pergerakan yang mungkin untuk menuju ke posisi tujuan tanpa melanggar aturan-aturan yang telah ditetapkan.

Variabel-variabel yang terdapat pada perangkat lunak adalah:

1. Kuda (putih): merupakan simbol posisi awal.



Gambar 4.1 Simbol Posisi Awal

2. Raja (hitam): merupakan simbol posisi tujuan.



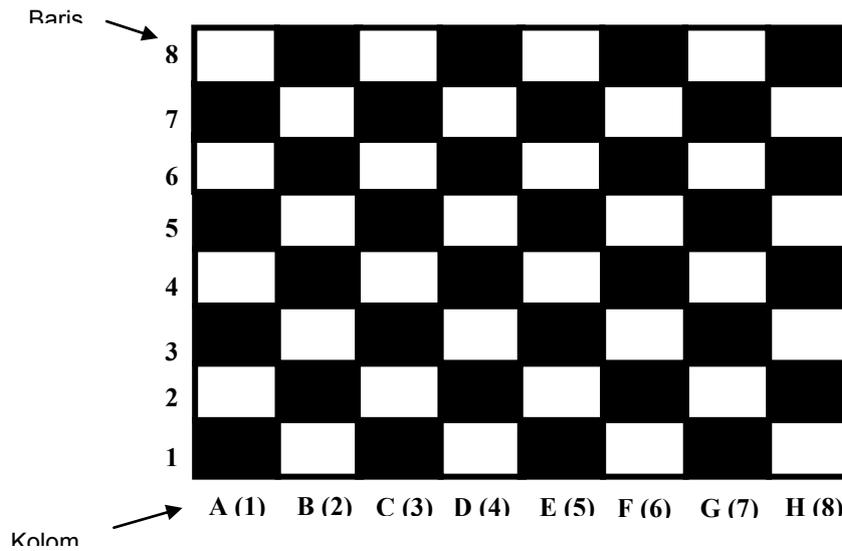
Gambar 4.2 Simbol Posisi Tujuan

3. Bidak/pion (hitam): merupakan penghalang yang tidak boleh ditempati oleh kuda .



Gambar 4.3 Simbol Penghalang

Wadah tempat pergerakan kuda adalah sebuah papan catur dengan ukuran 8 x 8 seperti terlihat pada gambar 4.4.



Gambar 4.4 Papan Catur 8 x 8

Permasalahan yang dihadapi adalah kuda harus digerakkan dari posisinya ke posisi raja dalam sebuah papan catur 8 x 8. Adapun aturan pergerakan biji kuda yang mungkin adalah:

1. Kuda dapat digeser ke atas sebelah kiri.

Operasi: Baris = baris + 2, Kolom = kolom - 1.

Aturan: Nilai variabel kolom setelah operasi harus > 0, nilai variabel baris <= m, dan tidak terdapat biji bidak beserta langkah aktif bidak pada posisi tersebut

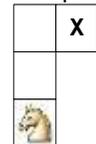


Gambar 4.5 Aturan Pergerakan-1

2. Kuda dapat digeser ke atas sebelah kanan.

Operasi: Baris = baris + 2, Kolom = kolom + 1.

Aturan: Nilai variabel kolom setelah operasi harus <= n dan baris setelah operasi harus <= m dan tidak terdapat biji bidak beserta langkah aktif bidak pada posisi tersebut.



Gambar 4.6 Aturan Pergerakan-2

3. Kuda dapat digeser ke kiri sebelah atas.

Operasi: Baris = baris + 1, Kolom = kolom - 2.

Aturan: Nilai variabel kolom setelah operasi harus > 0, nilai variabel baris <= m, dan tidak terdapat biji bidak beserta langkah aktif bidak pada posisi tersebut.

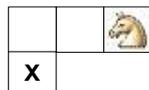


Gambar 4.7 Aturan Pergerakan-3

4. Kuda dapat digeser ke kiri sebelah bawah.

Operasi: Baris = baris - 1, Kolom = kolom - 2.

Aturan: Nilai variabel kolom dan baris setelah operasi harus > 0 dan tidak terdapat biji bidak beserta langkah aktif bidak pada posisi tersebut.



Gambar 4.8 Aturan Pergerakan-4

5. Kuda dapat digeser ke kanan sebelah atas.

Operasi: Baris = baris + 1, Kolom = kolom + 2.

Aturan: Nilai variabel kolom setelah operasi harus  $\leq m$  dan baris setelah operasi harus  $\leq n$  dan tidak terdapat biji bidak beserta langkah aktif bidak pada posisi tersebut.



Gambar 4.9 Aturan Pergerakan-5

6. Kuda dapat digeser ke kanan sebelah bawah.

Operasi: Baris = baris - 1, Kolom = kolom + 2.

Aturan: Nilai variabel kolom setelah operasi harus  $\leq n$ , nilai variabel baris harus  $> 0$ , dan tidak terdapat biji bidak beserta langkah aktif bidak pada posisi tersebut.

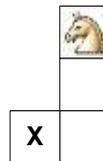


Gambar 4.10 Aturan Pergerakan-6

7. Kuda dapat digeser ke bawah sebelah kiri.

Operasi: Baris = baris - 2, Kolom = kolom - 1.

Aturan: Nilai variabel kolom dan baris setelah operasi harus  $> 0$  dan tidak terdapat biji bidak beserta langkah aktif bidak pada posisi tersebut.



Gambar 4.11 Aturan Pergerakan-7

8. Kuda dapat digeser ke bawah sebelah kanan.

Operasi: Baris = baris - 2, Kolom = kolom + 1.

Aturan: Nilai variabel kolom setelah operasi harus  $\leq n$ , nilai variabel baris harus  $> 0$ , dan tidak terdapat biji bidak beserta langkah aktif bidak pada posisi tersebut.



Gambar 4.12 Aturan Pergerakan-8

Algoritma yang dibutuhkan dalam merancang Aplikasi Simulasi Langkah Kuda Dalam Papan Catur dapat dibagi menjadi 2 (dua) bagian, yaitu :

1. Algoritma pencarian solusi.
2. Algoritma fungsi pendukung.

#### 4.1.1 Algoritma Pencarian Solusi

Seperti telah dijelaskan pada bab 2 bahwa pencarian solusi menggunakan metode *Breadth-First Search* (BFS). Algoritma pencarian BFS akan melakukan penelusuran *node* per level, artinya semua *node* pada level yang sama akan dikunjungi terlebih dahulu sebelum penelusuran dilanjutkan ke *node* pada level berikutnya. Karena bisa terdapat lebih dari satu solusi, maka penelusuran atau pengembangan *node* dilakukan hingga tidak ada *node* yang dapat dikembangkan lagi dan *node* yang berisi posisi tujuan merupakan solusi.

Terdapat 8 buah operasi untuk mengembangkan setiap *node* dalam pohon pencarian, yaitu:

1. Geser biji kuda ke atas sebelah kiri (Baris = baris + 2, Kolom = kolom - 1).
2. Geser biji kuda ke atas sebelah kanan (Baris = baris + 2, Kolom = kolom + 1).
3. Geser biji kuda ke kiri sebelah atas (Baris = baris + 1, Kolom = kolom - 2).
4. Geser biji kuda ke kiri sebelah bawah (Baris = baris - 1, Kolom = kolom - 2).
5. Geser kuda ke kanan sebelah atas (Baris = baris + 1, Kolom = kolom + 2).
6. Geser kuda ke kanan sebelah bawah (Baris = baris - 1, Kolom = kolom + 2).
7. Geser kuda ke bawah sebelah kiri (Baris = baris - 2, Kolom = kolom - 1).
8. Geser kuda ke bawah sebelah kanan (Baris = baris - 2, Kolom = kolom + 1).

Operasi ini harus mematuhi aturan sebagai berikut:

1. Posisi setelah pergeseran tidak boleh berada di luar papan catur.
2. Posisi setelah pergeseran tidak boleh sama dengan posisi biji bidak/pion (rintangan)
3. Posisi setelah pergeseran belum pernah dilintasi sebelumnya (pada satu jalur yang sama dalam pohon pencarian). Ini menjamin bahwa biji kuda tidak akan menempati posisi yang sudah pernah ditempati sebelumnya.

#### 4.1.2 Algoritma Fungsi Pendukung

Beberapa algoritma fungsi pendukung pada aplikasi ini adalah:

1. Fungsi 'ConvIndexToPosisi'.

Fungsi ini mengkonversi nilai indeks kotak menjadi posisi pada papan catur.

Untuk lebih memahami cara kerja fungsi, perhatikan gambar 4.13 berikut.

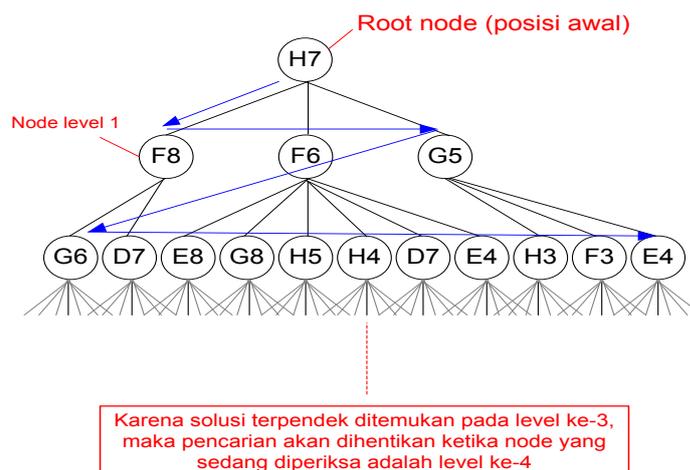
BARIS 8 7 6 5 4 3 2 1	Index = 57	Index = 58	Index = 59	Index = 60	Index = 61	Index = 62	Index = 63	Index = 64
	Index = 49	Index = 50	Index = 51	Index = 52	Index = 53	Index = 54	Index = 55	Index = 56
	Index = 41	Index = 42	Index = 43	Index = 44	Index = 45	Index = 46	Index = 47	Index = 48
	Index = 33	Index = 34	Index = 35	Index = 36	Index = 37	Index = 38	Index = 39	Index = 40
	Index = 25	Index = 26	Index = 27	Index = 28	Index = 29	Index = 30	Index = 31	Index = 32
	Index = 17	Index = 18	Index = 19	Index = 20	Index = 21	Index = 22	Index = 23	Index = 24
	Index = 9	Index = 10	Index = 11	Index = 12	Index = 13	Index = 14	Index = 15	Index = 16
	Index = 1	Index = 2	Index = 3	Index = 4	Index = 5	Index = 6	Index = 7	Index = 8
KOLOM	A (1)	B (2)	C (3)	D (4)	E (5)	F (6)	G (7)	H (8)

Gambar 4.13 Posisi dalam papan catur

Tata cara penomoran posisi papan catur dalam aplikasi seperti terlihat pada gambar 4.13 Apabila index = 30, maka fungsi ConvIndexToPosisi akan mengembalikan nilai variabel .Baris = 4, .Kolom = 6 dan .Posisi = "F4".

2. Fungsi 'IsAllNodeCek'.  
Fungsi ini melakukan pemeriksaan apakah semua *node* sudah ditelusuri. Fungsi mengembalikan nilai *boolean*. 'True' jika semua *node* sudah ditelusuri dan 'False' jika terdapat *node* yang belum ditelusuri.
3. Fungsi 'IsValidPosisi'.  
Fungsi melakukan pemeriksaan apakah suatu posisi *valid* atau tidak. Fungsi mengembalikan nilai *boolean*. 'True' jika posisi *valid* dan 'False' jika posisi tidak *valid*.

Pencarian solusi problema kuda dengan menggunakan metode BFS seperti terlihat pada gambar 4.14 berikut.



Gambar 4.14 Pencarian BFS pada problema kuda

Solusi terpendek dari permasalahan pada gambar 4.15 dapat dilihat pada tabel berikut.

**Tabel 4.1** Solusi Permasalahan Kuda

No.	Solusi	Langkah-Langkah	Panjang
1.	Solusi-1	F8, D7, E5	3 langkah
2.	Solusi-2	F8, G6, E5	3 langkah
3.	Solusi-3	F6, D7, E5	3 langkah
4.	Solusi-4	F6, G4, E5	3 langkah
5.	Solusi-5	G5, F3, E5	3 langkah

**4.2 Perancangan**

Perancangan aplikasi menggunakan bahasa pemrograman *Microsoft Visual Basic 6.0* dengan menggunakan beberapa komponen *standard visual basic*, seperti:

1. *Command Button*, sebagai tombol.
2. *Text Box*, sebagai tempat *input*.
3. *Image & PictureBox*, untuk menampilkan gambar.
4. *MSFlexGrid*, sebagai tabel dan komponen lainnya.

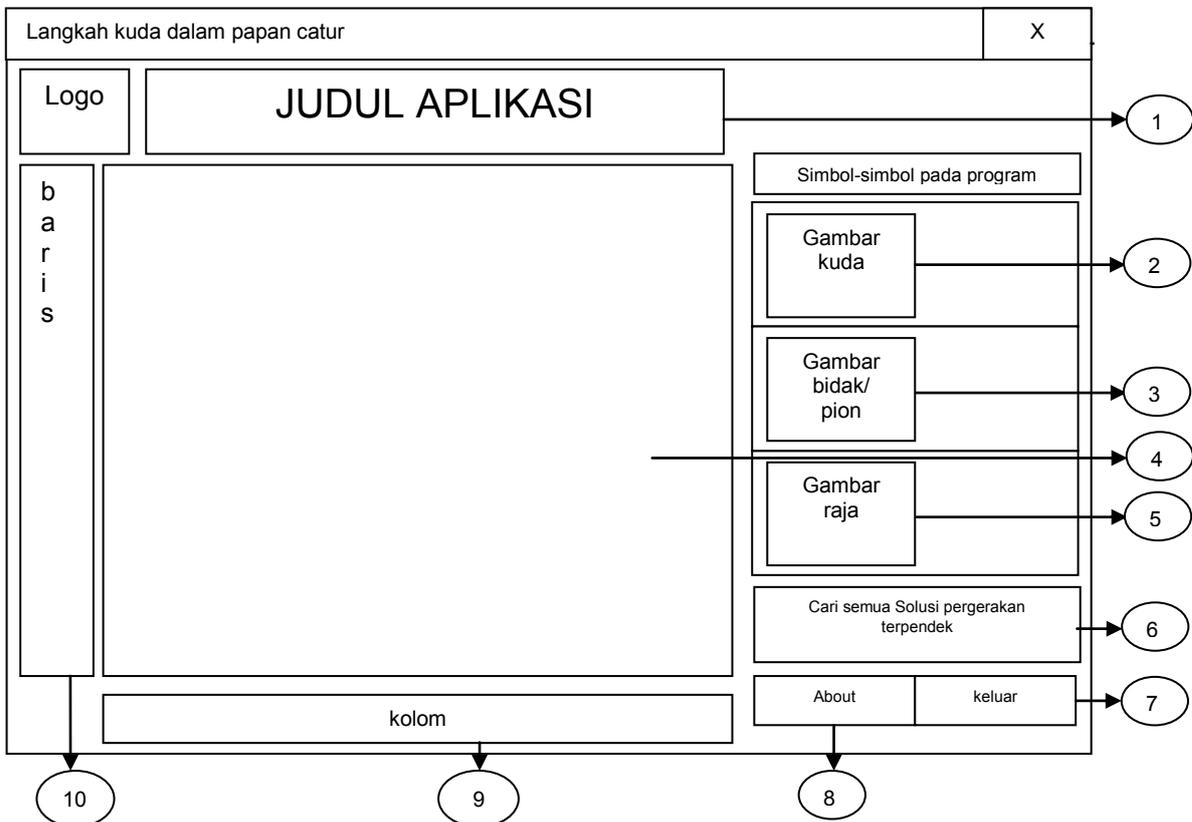
Aplikasi memiliki 3 (tiga) buah *form*, antara lain:

1. *Form Input*.
2. *Form Solusi*.
3. *Form About*.

Penjelasan perancangan dan fungsi masing-masing *form* akan disertakan pada subbab-subbab berikutnya.

**4.2.1 Form Input**

*Form Input* berfungsi untuk mengatur posisi awal, posisi tujuan dan posisi rintangan dalam permasalahan kuda.



**Gambar 4.15** Rancangan Form Input

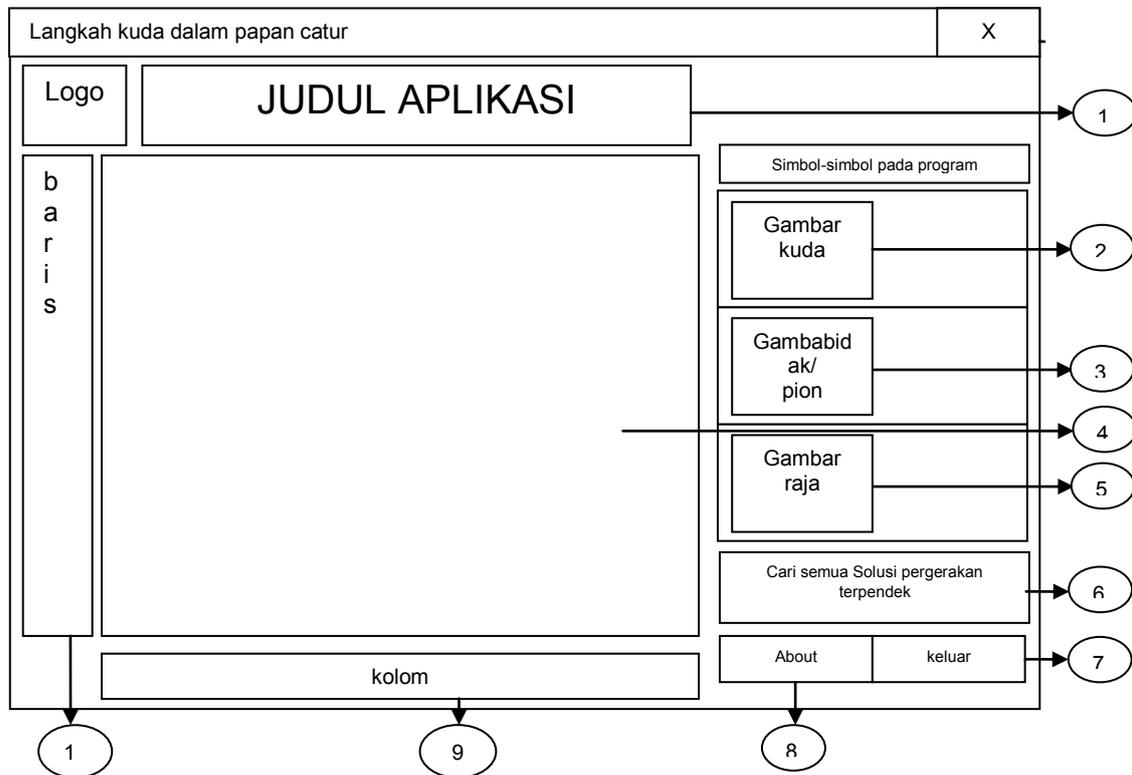
Keterangan:

**Tabel 4.2** Rancangan Form Input

No	Nama objek	Keterangan
1	TextBox	Judul aplikasi
2	Image kuda	Posisi awal
3	Image bidak/pion	Posisi rintangan
4	Image raja	Posisi tujuan
5	PictureBox	Daerah papan catur
6	CommandButton	Tombol cari semua solusi pergerakan terpendek
7	CommandButton	Keluar dari aplikasi
8	CommandButton	Menampilkan form About
9	TextBox	Kolom
10	TextBox	Baris

#### 4.2.2 Form Solusi

Form Solusi berfungsi untuk menampilkan semua solusi yang dihasilkan dan mensimulasikan gerakan kuda berdasarkan solusi yang dipilih.



**Gambar 4.16** Rancangan Form Solusi

Keterangan:

**Tabel 4.3** Rancangan Form Solusi

No	Nama objek	Keterangan
1	TextBox	Judul aplikasi
2	MSFlexGrid	Tabel untuk menampilkan semua solusi yang ada
3	MSFlexGrid	Tabel untuk menampilkan langkah-langkah pada solusi yang dipilih pada tabel no.2
4	PictureBox	Daerah papan catur
5	CommandButton	Tombol untuk memulai simulasi gerakan kuda.
6	CommandButton	Tombol keluar dari form simulasi
7	CommandButton	Tombol untuk mengulangi simulasi gerakan kuda dari awal.
8	TextBox	Kolom
9	TextBox	Baris

## 5. IMPLEMENTASI

Tampilan *form input error* pada saat posisi kuda (posisi awal) belum ditentukan terlihat pada gambar 5.1 berikut.



Gambar 5.1 Form input peringatan untuk posisi awal

Tampilan *form input error* pada saat posisi tujuan (posisi raja) belum ditentukan terlihat pada gambar 5.2 berikut.



Gambar 5.2 Form input peringatan untuk posisi tujuan

## 6. KESIMPULAN

1. Berhasil menemukan pencarian solusi terpendek dalam aplikasi menggunakan metode pencarian *Breadth First Search* (BFS).
2. Aplikasi ini merupakan implementasi (penerapan) nyata pohon pencarian dalam memecahkan suatu permasalahan berdasarkan konsep *Artificial Intelligence* (AI).
3. Aplikasi ini mensimulasikan semua gerakan dari semua solusi yang ditemukan, sehingga memberikan gambaran yang cukup jelas atas solusi-solusi yang dihasilkan.

## 7. DAFTAR PUSTAKA

- [1] Erdi Wijaya, SPK Untuk Penjadwalan Kuliah Dengan Teknik *Breadth First Search* (studi kasus Jurusan TIF UIN SUSKA RIAU). Universitas Islam Negeri Sultan Sarif Kasim. 2005
- [2] Kusumadewi.S, *Artificial Intelligence (Teknik dan Aplikasinya)*, Edisi 2, Penerbit Graha Ilmu, 2002.
- [3] Sandi.S, *Artificial Inteligencia*, Andi Offset Yogyakarta, 1993.
- [4] Suyoto, *Intelegensi Buatan Teori Dan Pemograman*. Penerbit Gava Media. 2004
- [5] <http://datahardisk.blogspot.com/2010/07/asal-usul-permainan-catur.html>. Diakses tanggal 13 Mei 2011.
- [6] <http://omkicau.com/2010/05/04/aneka-suara-binatang-liar-binatang-gurun-dan-suara-burung/> Diakses tanggal 13 Juni 2011
- [7] <http://www.google.co.id/picture-white-horse-chess-type-icon-file>. Diakses tanggal 13 juni 2011