BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya..

3.1.1 Analisis Masalah

Permasalahan ruang keadaan merupakan salah satu permasalahan yang terdapat dalam kehidupan sehari-hari yang dipelajari dalam cabang ilmu kecerdasan buatan. Salah satu metode penyelesaian yang diulas dalam keilmuan kecerdasan buata yaitu metode pencarian buta denga menggunakan *Breadth-first search.*

Permasalahan ini kerap dibahas secara teoritis dalam dunia pendidikan sehingga tidak sedikit dari kalangan pelajar sulit untuk memahami langkah penyelesaian dari permasalahan tersebut, serta dari kalangan pengajar yang kesulitan untuk memvisualisasikan langkah penyelesaian secara simulasi disamping langkah penyelesaian secara teoritis. Permasalahan ini juga banyak dikupas secara teoritis, dan juga divisualisasikan ke dalam bentuk *game.* Namun perpaduan kedua cara tersebut belum ada, bahkan mengupas materi pembelajaran

langkah penyelesaiannya belum tersajikan secara komputasi sistem kecerdasan buatan, sehingga pemahaman tentang metode penyelesaian permasalahan ruang keadaan menggunakan *breadth-first search* kurang terbantukan.

3.1.2 Analisis Data

3.1.2.1 Data Masukan

Data masukan dari aplikasi bantu kecerdasan buatan pembelajaran dan penyelesaian pada permasalahan ruang keadaan yang digunakan terdiri dari data teori dan materi pembelajaran, data *Problem*, data *Variabel*, data *Control*. Rincian data di atas adalah sebagai berikut:

- Data teori dan materi pembelajaran merupakan data-data mengenai pembelajaran serta pemahaman langkah peyelesaian secara teoritis. Data ini merupakan data statis yang tidak dapat dirubah oleh *user.*
- Data *Problem* merupakan data permasalahan yang akan diselesaikan dan disimulasikan. *Problem* yang akan dikaji yaitu permasalahan *farmer problems* yang merupakan salah satu kasus dalam permasalahan ruang keadaan (*State and Space*). Data problem ini akan dimodifikasi sesuai dengan data masukan dari *user*, tentunya *problem* yang memiliki kemiripan pola dan prosedur kasus yang akan diselesaikan yaitu *farmer problem*, seperti *cannibal missionaries, family problems.*
- Data *Variabel* merupakan pendefinisian dari *variabel* apa saja yang terlibat dalam *problem*.
- Data *Control* yaitu merupakan strategi kontrol apa saja yang harus diperhatikan oleh *variabel* untuk menyelesaikan *problem.*

3.1.2.2 Analisis Metode Penyelesaian

1. Searching

Masalah utama dalam membangun sistem berbasis AI adalah bagaimana mengkonversi situasi yang diberikan ke dalam situasi lain yang diinginkan menggunakan sekumpulan operasi tertentu, sehingga permasalahan ruang keadaan dapat diselesaikan.

Salah satu contoh permasalahan ruang keadaan yang dibahas adalah masalah petani, kambing, serigala dan sayur-sayuran yang sering juga disebut dengan *Farmer's Problem*. Ilustrasi dari permasalahan ini adalah sebagai berikut, seorang petani akan menyeberangkan seekor kambing, seekor serigala dan sayur-sayuran dengan sebuah rakit melalui sungai. Rakit hanya bisa memuat petani dan satu penumpang yang lain (kambing, serigala atau sayur-sayuran). Jika ditinggalkan oleh petani tersebut, maka sayur-sayuran akan dimakan oleh kambing dan kambing akan dimakan oleh serigala.

Untuk dapat menyelesaikan permasalahan diatas diperlukan tiga langkah berikut :

a. Definisikan ruang masalah, keadaan awal (*initial state)* dan keadaan tujuan (*goal state).*

Setiap keadaan dari *Farmer's Problem* dapat direpresentasikan dengan (kambing, petani, serigala, sayur-sayuran) dan posisi rakit.

Keadaan awal dan keadaan tujuan dapat direpresentasikan sebagai berikut :

1) Keadaan awal (*start state*):
    a) Daerah kiri = (1,1,1,1).
    b) Daerah kanan = (0,0,0,0)
    c) Posisi rakit berada di kiri.

Artinya, terdapat kambing, petani, serigala dan sayur-sayuran di daerah kiri.

2) Keadaan tujuan (*goal state*):
    a) Daerah kiri = (0,0,0,0).
    b) Daerah kanan = (1,1,1,1)
    c) Posisi rakit berada di kanan.

Artinya, terdapat kambing, petani, serigala dan sayur-sayuran di daerah kanan.

b. Definisikan aturan produksi

Setelah keadaan awal dan keadaan tujuan sudah didefinisikan pada langkah pertama, langkah kedua yaitu mendefinisikan aturan produksi yang berlaku, mendefinisikan Variabel, Aturan (*rule*), dan strategi kontrol.

Tabel 3.1. Variabel Farmer's Problem

Variabel ID	Nama Variabel	Bisa mengemudikan rakit (Driver)
K	Kambing	Tidak

P	Petani	Ya
SR	Serigala	Tidak
SY	Sayur-sayuran	Tidak

Tabel 3.2. *Control Farmer's Problem*

No	Control
1	Kambing akan memakan sayur-sayuran apabila Petani tidak berada di tempat.
2	Serigala akan memakan Kambing apabila Petani tidak berada di tempat.

Dari variabel dan control yang terdapat pada *Farmer's Problem*, kita dapat menyimpulkan aksi-aksi apa saja yang dapat dilakukan dengan tetap mematuhi kontrol yang ada. Aksi-aksi yang dapat dilakukan dapat dilihat pada Tabel 3.3.

Tabel 3.3. Tabel Aksi-aksi yang dapat terjadi pada *Farmer's Problem*

Aksi ke-	Aturan
1	Petani dan kambing menyeberang.
2	Petani dan serigala menyeberang.
3	Petani dan sayur-sayuran menyeberang.
4	Petani dan kambing kembali.
5	Petani dan serigala kembali.
6	Petani dan sayur-sayuran kembali.
7	Petani kembali.

Pencarian solusi dari permasalahan di atas adalah dengan menggunakan bantuan pohon pelacakan dengan penjelasan sebagai berikut :

1. Akar dari pohon merupakan keadaan awal.
2. Akar pohon kemudian membentuk cabang-cabang baru dengan melakukan semua aksi-aksi yang dapat terjadi pada keadaan awal.
3. Keadaan-keadaan baru yang dihasilkan merupakan cabang baru dari pohon.
4. Proses pengembangan dan pencarian dilanjutkan dengan membentuk keadaan baru (cabang baru) dari cabang-cabang yang sudah ada hingga didapatkan solusi.
5. Proses pembentukan keadaan baru juga melakukan prosedur pengecekan apakah keadaan tersebut sudah pernah dibentuk sebelumnya pada pohon.
6. Apabila sudah ada, maka cabang baru tidak dibentuk dan apabila tidak, maka cabang baru dibentuk.
7. Apabila tidak ada cabang yang dapat dikembangkan lagi dan ternyata belum ditemukan solusi, maka permasalahan yang sedang diselesaikan tidak memiliki solusi.

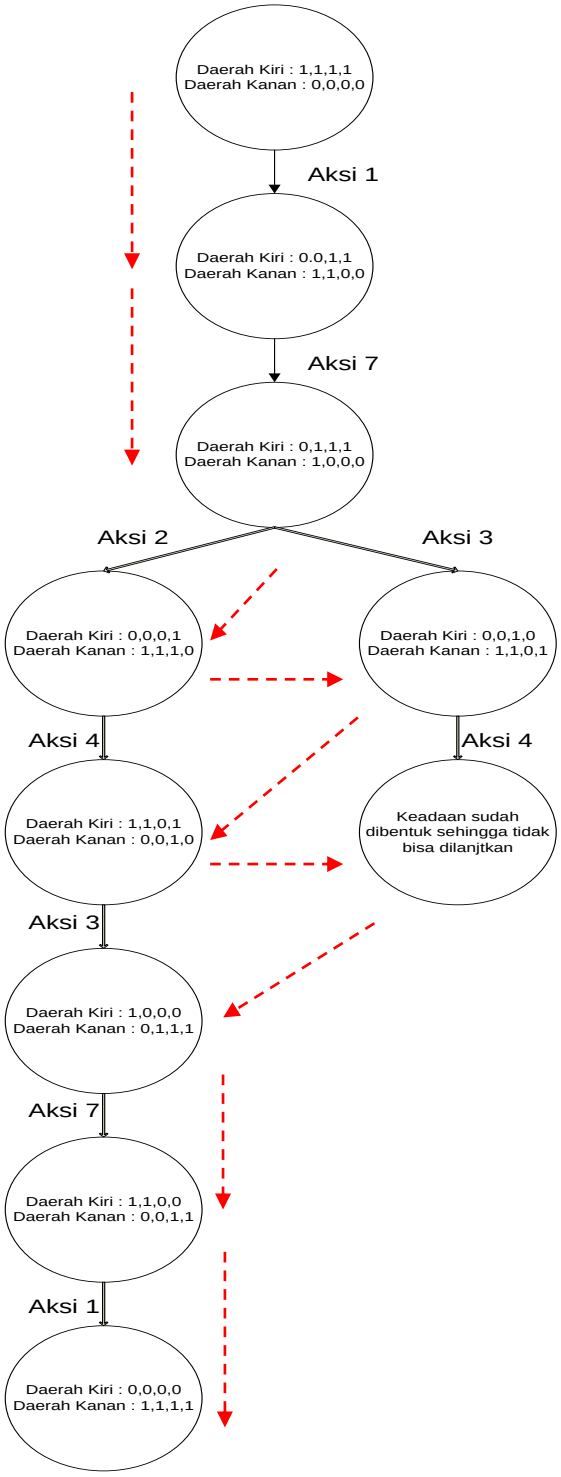
c. Memilih metode pencarian yang tepat

Metode pencarian yang dipergunakan yaitu metode pencarian buta (*Blind Search*) yaitu metode *Breadth-First Search (BFS)*.

2. Metode *Breadth-First Search (BFS)*

Metode pencarian *Breadth-First Search (BFS)* dipilih karena memiliki kelebihan menemukan solusi dan solusi yang diambil yaitu solusi minimum yang terpendek.

Berikut ini merupakan struktur pohon pelacakan untuk mencari solusi *farmer's problem* menggunakan *Breadth First Search* :



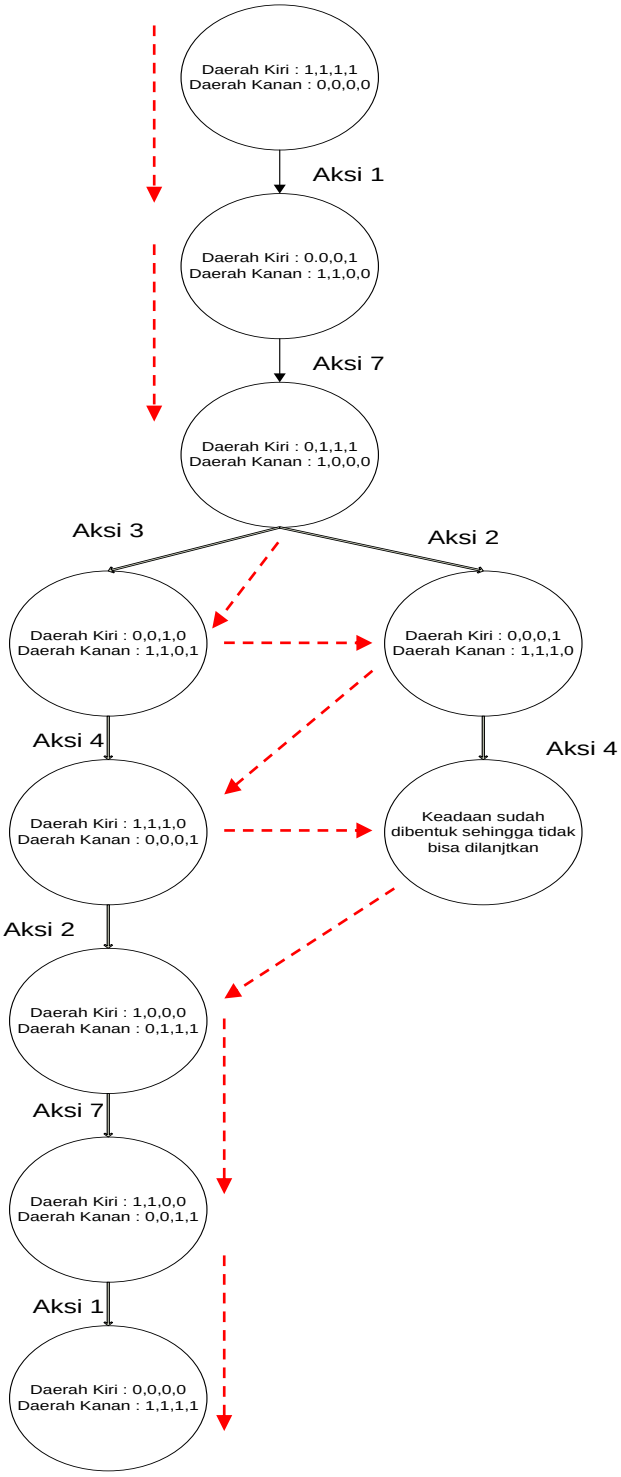
Gambar 3.1. Struktur pohon pelacakan solusi 1 untuk Farmer's Problem dengan menggunakan Breadth First search

Dari struktur pohon pelacakan yang dikembangkan pada gambar 3.1, didapatkan solusi seperti terlihat pada tabel 3.4.

Tabel 3.4. Tabel Solusi Farmer's Problem

Daerah Kiri	Aksi	Posisi Rakit	Daerah Kanan
(1,1,1,1)	-	KIRI	(0,0,0,0)
(0,0,1,1)	Aksi ke-1 (Petani dan kambing menyeberang)	KANAN	(1,1,0,0)
(0,1,1,1)	Aksi ke-7 (Petani kembali)	KIRI	(1,0,0,0)
(0,0,0,1)	Aksi ke-2 (Petani dan serigala menyeberang)	KANAN	(1,1,1,0)
(1,1,0,1)	Aksi ke-4 (Petani dan kambing kembali)	KIRI	(0,0,1,0)
(1,0,0,0)	Aksi ke-3 (Petani dan sayuran menyeberang)	KANAN	(0,1,1,1)
(1,1,0,0)	Aksi ke-7 (Petani kembali)	KIRI	(0,0,1,1)
(0,0,0,0)	Aksi ke-1 (Petani dan kambing menyeberang)	KANAN	(1,1,1,1)

Pencarian solusi pada struktur pohon pelacakan menggunakan metode pencarian melebar pertama (*breadth-first search*), sehingga solusi yang didapatkan adalah merupakan solusi terpendek.



Gambar 3.2. Struktur pohon pelacakan solusi 2 untuk Farmer's Problem dengan menggunakan Breadth First Search

Dari struktur pohon pelacakan yang dikembangkan pada Gambar 3.2 didapatkan solusi seperti terlihat pada tabel 3.5.

Tabel 3.5. Tabel Solusi Farmer's Problem

Daerah Kiri	Aksi	Posisi Rakit	Daerah Kanan
(1,1,1,1)	-	KIRI	(0,0,0,0)
(0,0,1,1)	Aksi ke-1 (Petani dan kambing menyeberang)	KANAN	(1,1,0,0)
(0,1,1,1)	Aksi ke-7 (Petani kembali)	KIRI	(1,0,0,0)
(0,0,0,1)	Aksi ke-3 (Petani dan sayuran menyeberang)	KANAN	(1,1,1,0)
(1,1,0,1)	Aksi ke-4 (Petani dan kambing kembali)	KIRI	(0,0,1,0)
(1,0,0,0)	Aksi ke-2 (Petani dan serigala menyeberang)	KANAN	(0,1,1,1)
(1,1,0,0)	Aksi ke-7 (Petani kembali)	KIRI	(0,0,1,1)
(0,0,0,0)	Aksi ke-1 (Petani dan kambing menyeberang)	KANAN	(1,1,1,1)

Pencarian solusi pada struktur pohon pelacakan menggunakan metode pencarian melebar pertama (*breadth-first search*), sehingga solusi yang didapatkan adalah merupakan solusi terpendek.

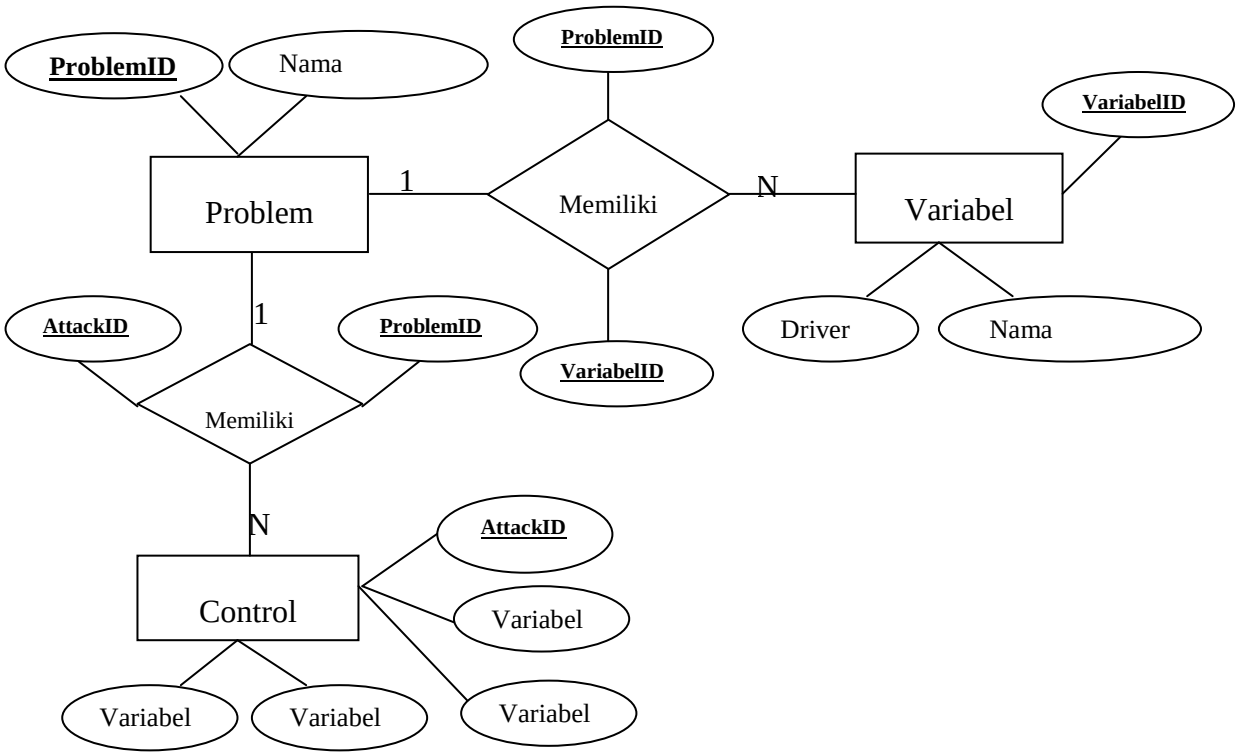
3.1.3 Analisis Informasi

Output yang akan diperoleh dari hasil *input*an data yang telah dijalankan yaitu berupa materi pembelajaran dan pemahaman tentang permasalahan ruang keadaan *(state and space),* serta solusi dari langkah penyelesaian yang dijalankan oleh sistem baik secara manual maupun secara otomatis melalui aplikasi simulator.

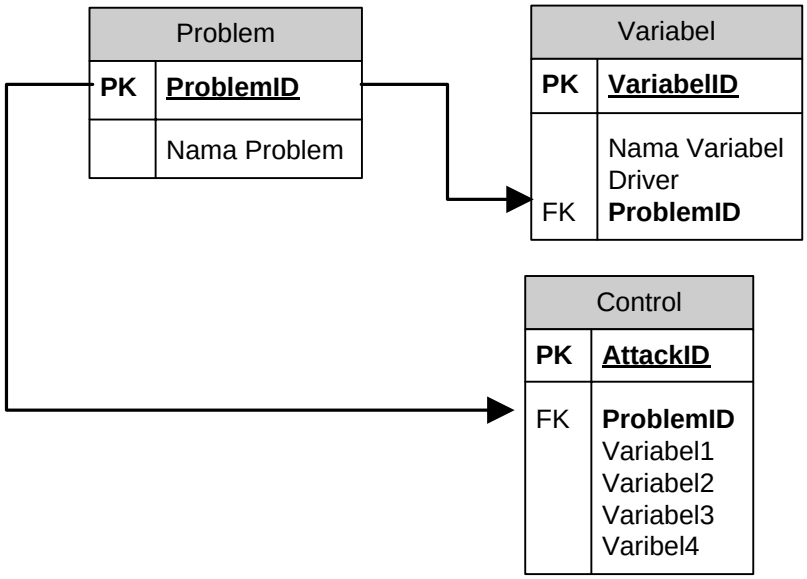
3.1.4 Analisis Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik.

3.1.4.1 ERD dan Skema Relasi



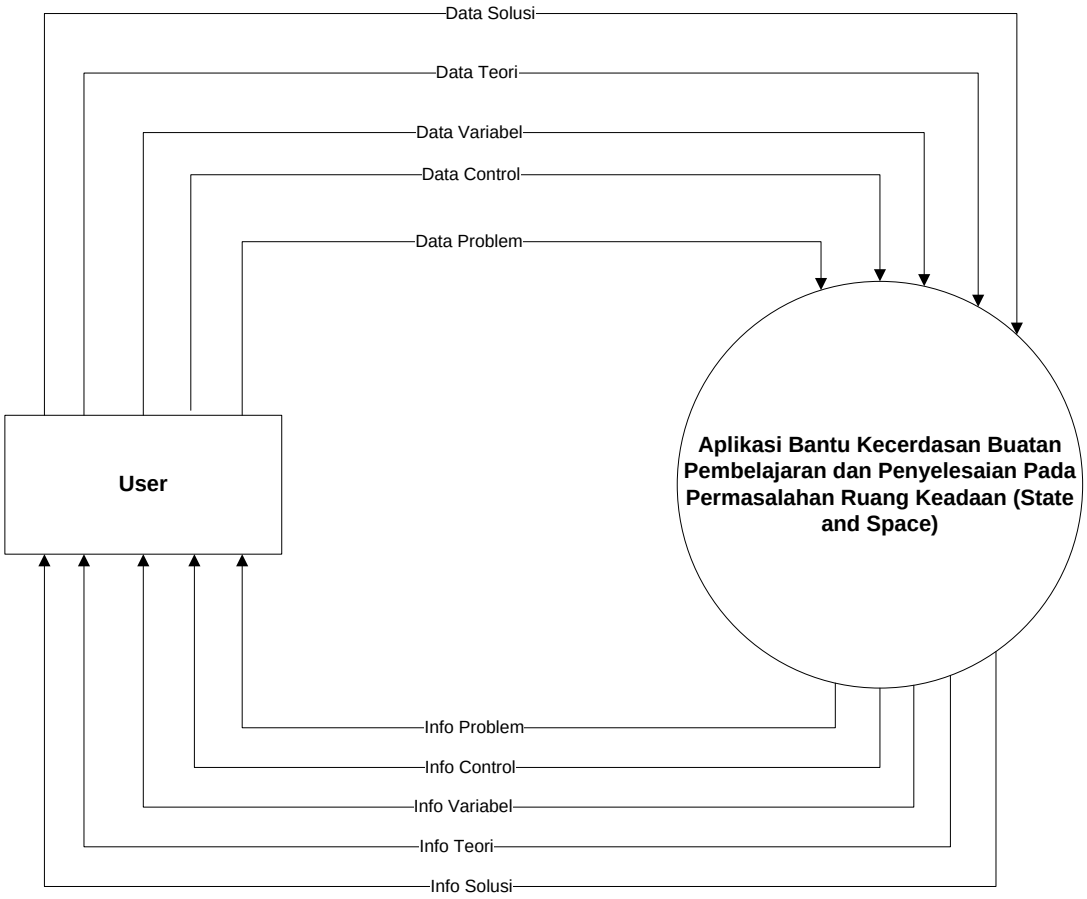
Gambar 3.3. ERD



Gambar 3.4. Skema Relasi

3.1.4.2 Diagram Konteks

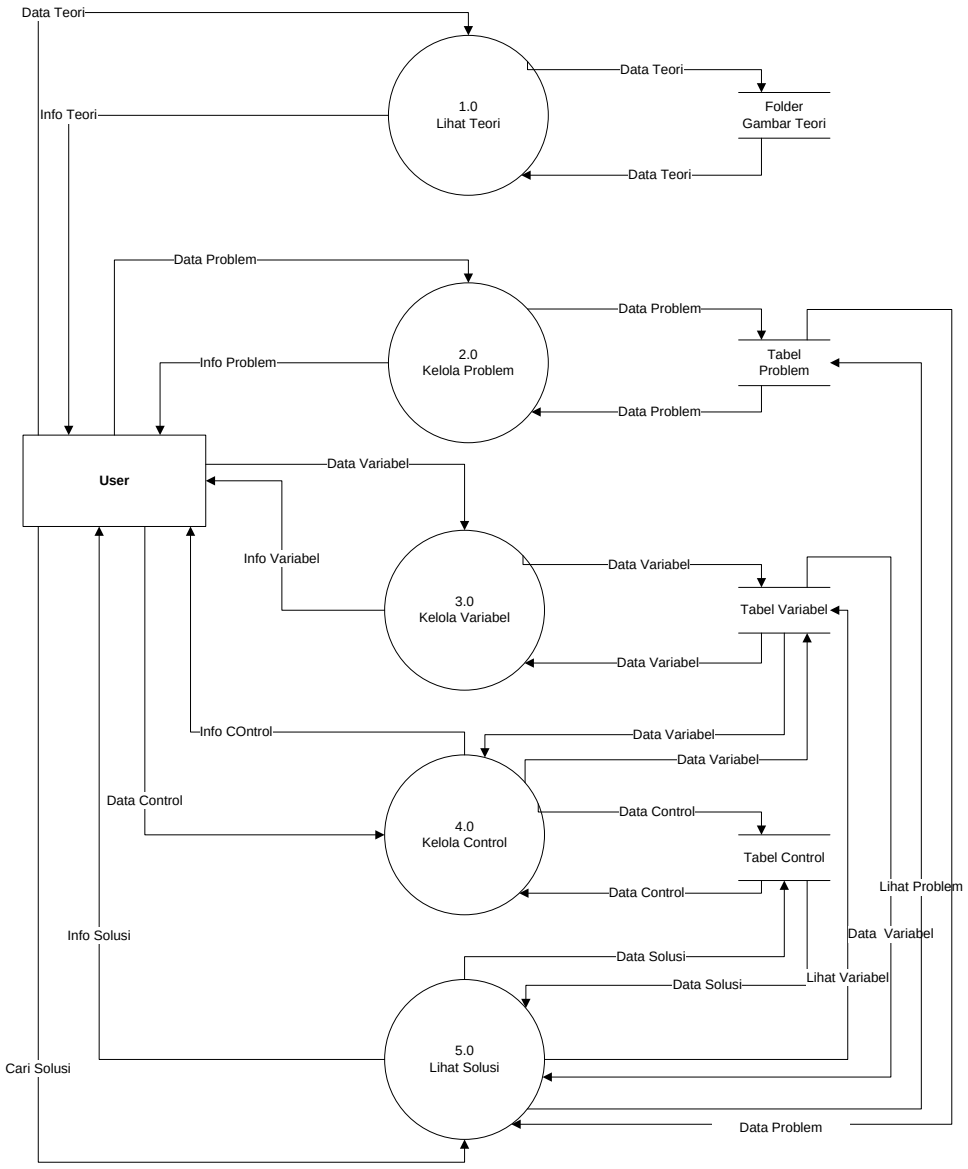
Diagram konteks merupakan alat untuk struktur analisis. pendekatan struktur ini untuk menggambarkan sistem secara garis besar atau secara keseluruhan.



Gambar 3.5. Diagram Konteks

3.1.4.3 Data Flow Diagram (DFD)

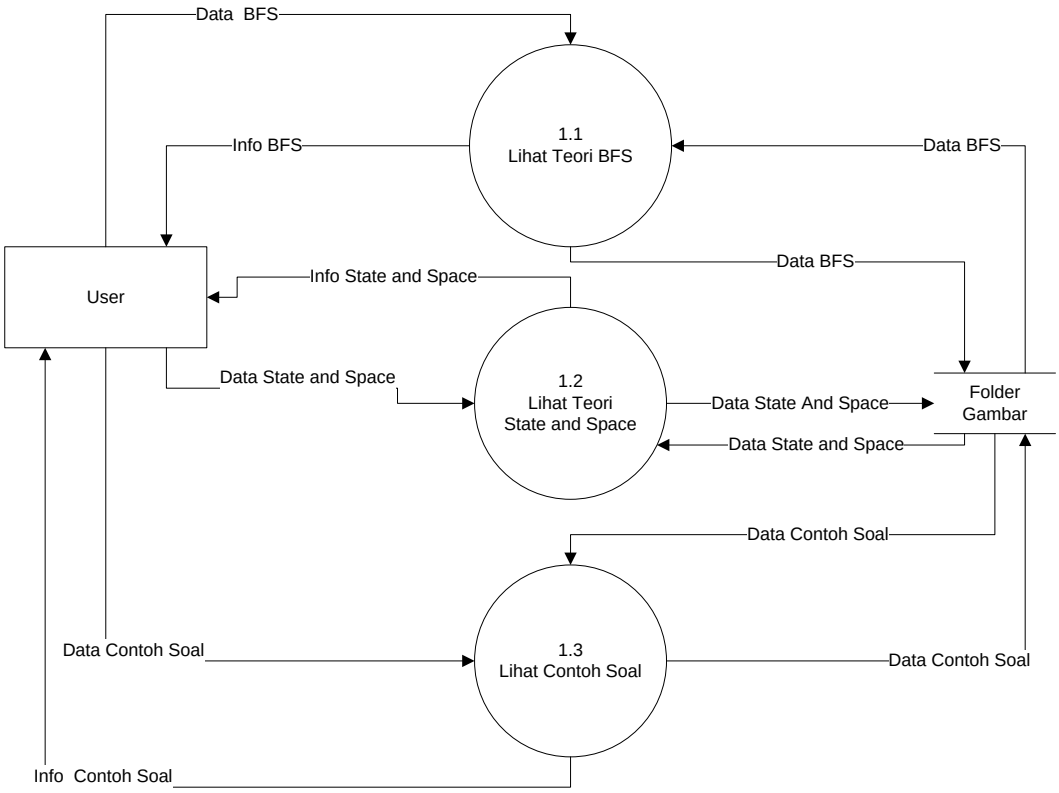
Data Flow Diagram merupakan suatu media yang digunakan untuk menggambarkan aliran proses aplikasi. DFD aplikasi bantu kecerdasan buatan pembelajaran dan penyelesaian permasalahan ruang keadaan (*state and space*) terdiri dari :



Gambar 3.4. DFD Level 1

1. DFD level 2 Proses 1 Lihat Teori

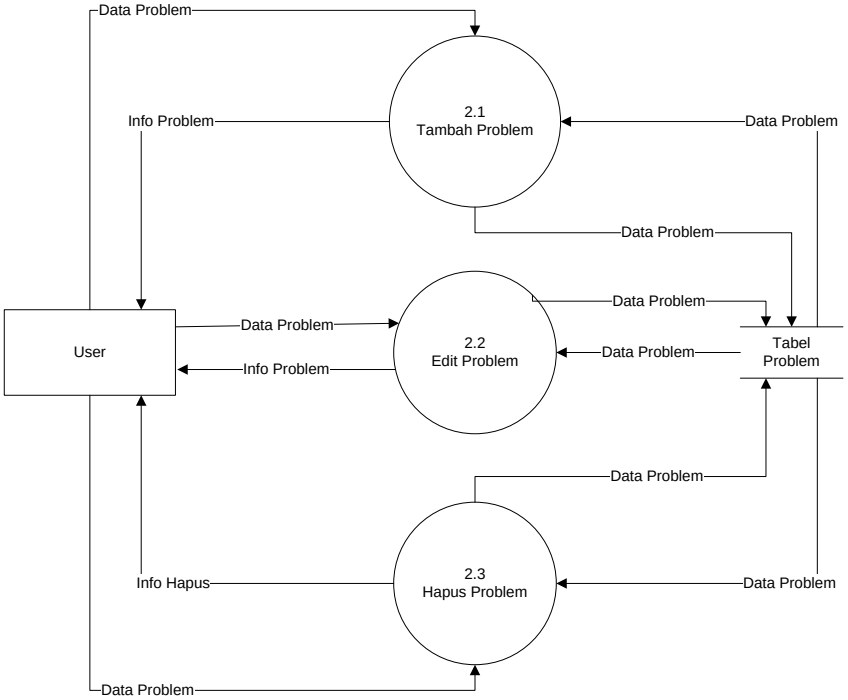
   DFD level 2 Proses 1 merupakan proses pengguna untuk melihat materi dan teori tentang *Breadth-First Search* dan Teori *State and Space*. Materi dan teori tidak disimpan didalam database melainkan disimpan di *folder* dalam bentuk gambar.



Gambar 3.7. DFD Level 2 Proses 1 Lihat Teori

2. DFD Level 2 Proses 2 Kelola Problem

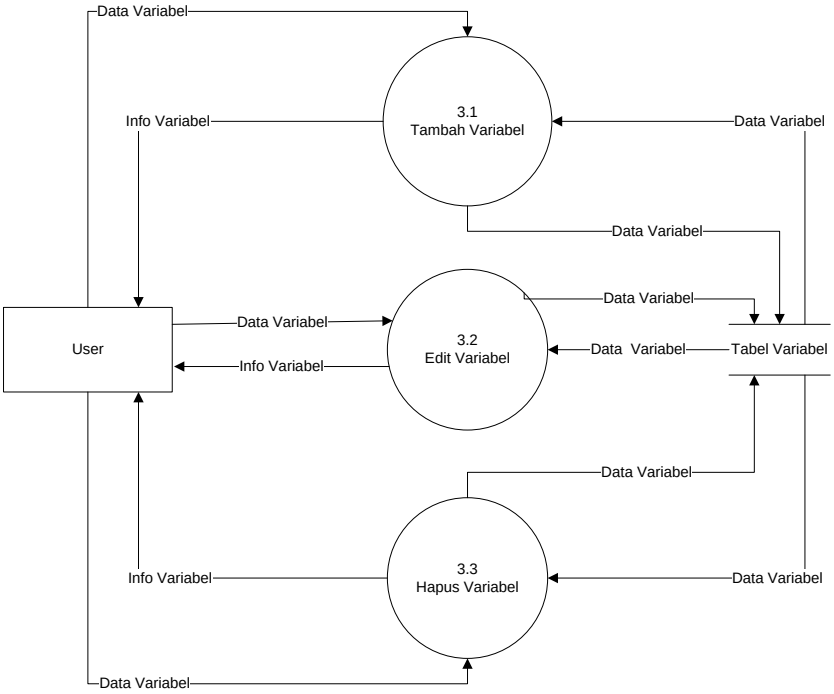
   DFD Level 2 Proses 2 adalah proses dimana pengguna mengelola *problem* proses yang dijalankan yaitu menambah, edit serta menghapus *problem* dari table database *Problem.*



Gambar 3.8. DFD Level 2 Proses 2 Kelola Problem

3. DFD Level 2 Proses 3 Kelola Variabel

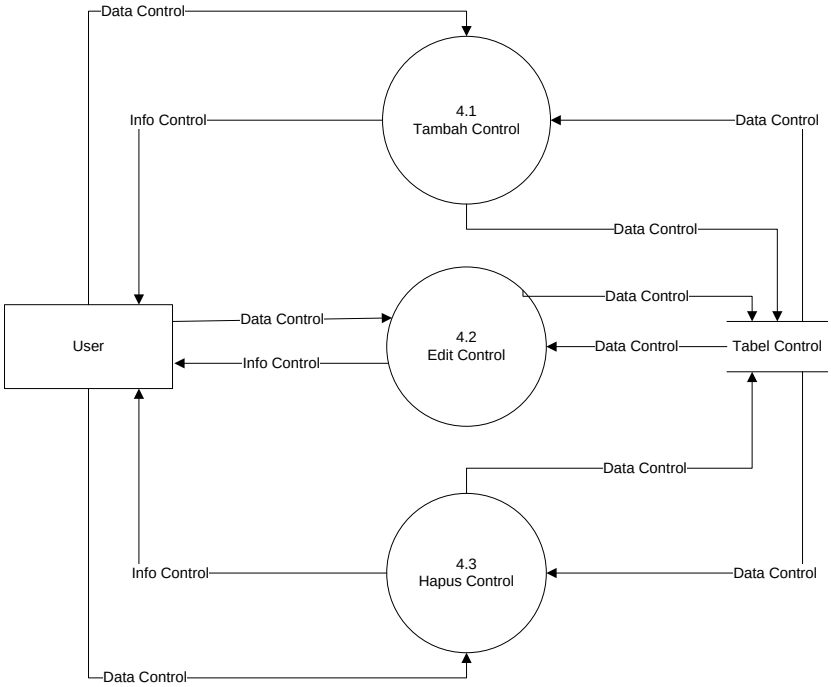
   DFD Level 2 Proses 3 adalah proses dimana pengguna mengelola *variabel*. Proses yang dijalankan yaitu menambah, edit serta menghapus *variabel* dari table database *Variabel.*



Gambar 3.9. Level 2 Proses 3 Kelola Variabel

4. DFD Level 2 Proses 4 Kelola Control

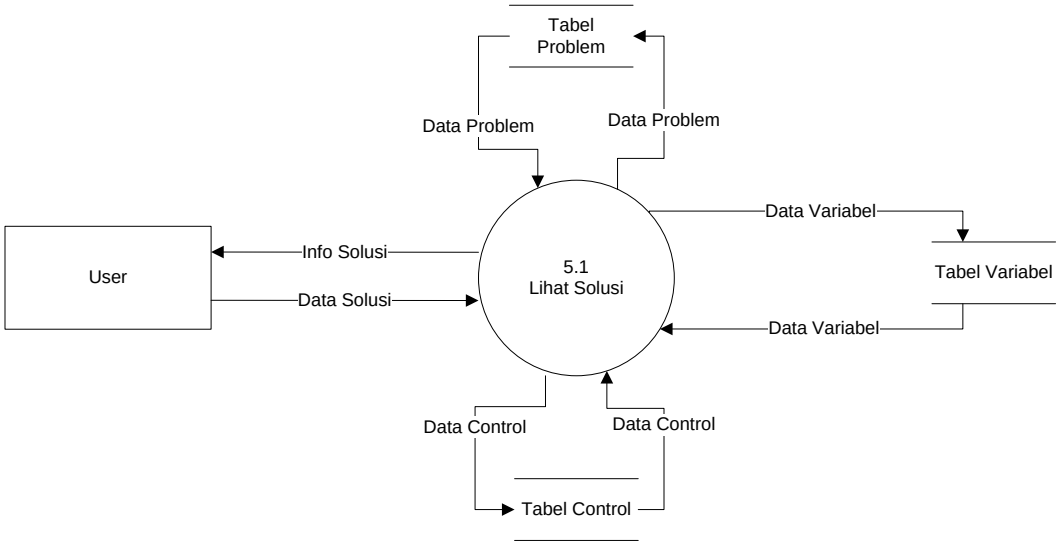
   DFD Level 2 Proses 4 adalah proses dimana pengguna mengelola *control*. Proses yang dijalankan yaitu menambah, edit serta menghapus *control* dari table database *control*.



Gambar 3.10. Level 2 Proses 4 Kelola Control

5. DFD Level 2 Proses 5 Lihat Solusi

   DFD Level 2 Proses 5 adalah proses dimana pengguna dapat melihat hasil dari solusi yang didapat melalui kombinasi yang diperoleh dari sistem yang diambil dari tabel *variabel*, tabel *Problem*, tabel *Control.*



Gambar 3.11. Level 2 Proses 5 lihat solusi

3.1.4.4 Spesifikasi Proses

Spesifikasi proses merupakan deskripsi dari setiap elemen proses yang terdapat dalam program, yang meliputi nama proses, input, output dan keterangan dari proses. Berikut ini spesifikasi proses yang berjalan :

Tabel 3.6. Spesifikasi Proses

No	Proses	Keterangan
1	No. Proses	1.0
	Nama Proses	Lihat Teori
	Source (sumber)	*User*
	Input	- Data Teori *Breadth-First Search* - Data Teori *State and Space* - Data Contoh Soal
	Output	- Info Teori *Breadth-First Search* - Info Teori *State and Space* - Info Contoh Soal
	Destination (tujuan)	*User*
	Logika Proses	1. *User* memilih menu utama *State and Space* 2. *User* memilih submenu Materi -> *State and Space, BFS.* 3. *User* memilih submenu Contoh Soal 4. Sistem melakukan proses menampilkan data Breadth First Search, State and Space, dan Contoh Soal, data diambil dari folder *images* yaitu berupa file *.jpg 5. *User* melihat data teori yang ditampilkan sistem
2	No. Proses	2.0

	Nama Proses	Kelola Problem
	Source (sumber)	*User*
	Input	- Data *Problem*
	Output	- Info *Problem*
	Tujuan	*User*
	Logika Proses	1. *User* memilih menu utama File 2. *User* memilih submenu New -> *Problem* 3. Sistem menampilkan form dan data *Problem* 4. *User* melihat data *problem*
3	No. Proses	3.0
	Nama Proses	Kelola Variabel
	Source (sumber)	*User*
	Input	- Data Variabel
	Output	- Info Variabel
	Destination (tujuan)	*User*
	Logika Proses	1. *User* memilih menu utama File 2. *User* memilih submenu New -> Variabel 3. Sistem menampilkan form dan data Variabel 4. *User* melihat data Variabel
4	No. Proses	4.0
	Nama Proses	Kelola Control
	Source (sumber)	*User*
	Input	- Data Control

	Output	- Info Control
	Destination (tujuan)	*User*
	Logika Proses	Begin 1. *User* memilih menu utama File 2. *User* memilih submenu New -> Control 3. Sistem menampilkan form dan data Control 4. *User* melihat data Control
5	No. Proses	1.1
	Nama Proses	Lihat Teori BFS
	Source (sumber)	*User*
	Input	- Pilih menu *State and Space* - Pilih menu Materi - Pilih menu BFS
	Output	- Tampilan Teori BFS
	Destination (tujuan)	*User*
	Logika Proses	1. *User* memilih menu BFS 2. Sistem melakukan proses pemanggilan data materi BFS dari folder *images* 3. Sistem menampilkan materi BFS 4. *User* melihat data materi BFS
6	No. Proses	1.2
	Nama Proses	Lihat Teori State and Space
	Source (sumber)	*User*
	Input	- Pilih menu *State and Space* - Pilih menu Materi - Pilih menu *State and Space*
	Output	- Tampilan teori State and Space

	Destination (tujuan)	*User*
	Logika Proses	1. *User* memilih menu *State and Space* 2. Sistem melakukan proses pemanggilan data materi *State and Space* dari folder *images* 3. Sistem menampilkan materi *State and Space* 4. *User* melihat data materi *State and Space*
7	No. Proses	1.3
	Nama Proses	Lihat Contoh Kasus
	Source (sumber)	*User*
	Input	- Pilih menu *State and Space* - Pilih menu Contoh Kasus
	Output	- Tampilan Form Contoh Kasus
	Destination (tujuan)	*User*
	Logika Proses	1. *User* memilih menu Contoh Kasus 2. Sistem melakukan proses pemanggilan data Contoh Kasus dari folder *images* 3. Sistem menampilkan data Contoh Kasus 4. *User* melihat data materi Contoh Kasus
8	No. Proses	2.1
	Nama Proses	Tambah Problem
	Source (sumber)	*User*
	Input	- Data Problem
	Output	- Info Problem
	Destination (tujuan)	*User*
	Logika Proses	1. *User* milih menu baru

		2. *User* memasukkan data problem 3. Sistem menyimpan data problem 4. Sistem menampilkan data problem 5. *User* melihat data problem
9	No. Proses	2.2
	Nama Proses	Edit Problem
	Source (sumber)	*User*
	Input	- Data Problem
	Output	- Info Problem
	Destination (tujuan)	*User*
	Logika Proses	1. *User* memilih menu edit 2. Sistem menyimpan data yang telah di edit 3. Sistem menampilkan data problem 4. *User* melihat data problem
10	No. Proses	2.3
	Nama Proses	Hapus Problem
	Source (sumber)	*User*
	Input	- Data Problem yang akan dihapus
	Output	- Peringatan menghapus problem - Info Problem
	Destination (tujuan)	*User*
	Logika Proses	1. *User* memilih menu Hapus 2. Sistem menampilkan data problem
11	No. Proses	3.1

	Nama Proses	Tambah Variabel
	Source (sumber)	*User*
	Input	- Data Variabel
	Output	- Info Variabel
	Destination (tujuan)	*User*
	Logika Proses	1. *User* milih menu baru 2. *User* memasukkan data variabel 3. Sistem menyimpan data variabel 4. Sistem menampilkan data variabel 5. *User* melihat data variabel
12	No. Proses	3.2
	Nama Proses	Edit variabel
	Source (sumber)	*User*
	Input	- Data Variabel
	Output	- Info Variabel
	Destination (tujuan)	*User*
	Logika Proses	1. *User* memilih menu edit 2. Sistem menyimpan data Variabel yang telah di edit 3. Sistem menampilkan data Variabel 4. *User* melihat data Variabel
13	No. Proses	3.3
	Nama Proses	Hapus variabel
	Source (sumber)	*User*
	Input	- Data variabel yang akan dihapus

	Output	- Peringatan menghapus variabel - Info variabel
	Destination (tujuan)	*User*
	Logika Proses	1. *User* memilih menu Hapus 2. Sistem menampilkan data Variabel
14	No. Proses	4.1
	Nama Proses	Tambah Control
	Source (sumber)	*User*
	Input	- Data Control
	Output	- Info Control
	Destination (tujuan)	*User*
	Logika Proses	1. *User* milih menu baru 2. *User* memasukkan data Control 3. Sistem menyimpan data Control 4. Sistem menampilkan data Control 5. *User* melihat data Control
15	No. Proses	4.2
	Nama Proses	Edit Control
	Source (sumber)	*User*
	Input	- Data Control
	Output	- Info Control
	Destination (tujuan)	*User*
	Logika Proses	1. *User* memilih menu Control 2. Sistem menyimpan data Control yang telah di edit 3. Sistem menampilkan data Control 4. *User* melihat data Control

16	No. Proses	4.3
	Nama Proses	Hapus Control
	Source (sumber)	*User*
	Input	- Data Control yang akan dihapus
	Output	- Peringatan menghapus Control - Info Control
	Destination(tujuan)	*User*
	Logika Proses	1. *User* memilih menu Hapus 2. Sistem menampilkan data Control

3.1.4.5 Struktur tabel

Dalam database, aplikasi memiliki 3 buah tabel, yaitu:

1. Tabel *Problem*.

Tabel ini berisi nama *problem* atau permasalahan yang pernah didaftarkan.

Struktur tabel adalah:

Tabel 3.7. Struktur Tabel Problem

No.	Field Name	Data Type	Field Size
1.	ProblemID	Text	10
2.	NamaProblem	Text	100

2. Tabel Variabel.

Tabel ini berisi variabel-variabel yang terdapat pada *problem*.

Struktur tabel adalah:

Tabel 3.8. Struktur Tabel Variabel

No.	Field Name	Data Type	Field Size
1.	ProblemID	Text	10
2.	VariabelID	Text	10
3.	NamaVariabel	Text	50
4.	Driver	Boolean	-

3. Tabel Control.

Tabel ini berisi kontrol yang harus dipatuhi pada *problem*.

Struktur tabel adalah:

Tabel 3.9. Struktur Tabel Control

No.	Field Name	Data Type	Field Size
1.	ProblemID	Text	10
2.	Variabel1	Text	50
3.	AttackID	Text	50
4.	Variabel2	Text	50
5.	Variabel3	Text	50
6.	Variabel4	Text	50

3.1.5 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan.

Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

3.1.5.1 Analisis Perangkat Keras

Perangkat keras yang dipergunakan untuk melakukan pembangunan aplikasi yaitu :

1. *Processor* Intel Core Duo T4300 2,1 GHz.
2. *Harddisk* Seagate ST3320620AS 160 GB SATA2.
3. *Memory*DDR2 1 GB.
4. VGA Intel GMA 4500M.
5. *Keyboard* dan *mouse.*

Sedangkan spesifikasi minimum untuk implementasi aplikasi perangkat lunak yaitu :

1. *Processor* dengan kecepatan 1,6 GHz.
2. Kapasitas *harddisk* minimal 20 GB.

3. *Memory* 256 MB.
4. Kartu grafis 64 MB.
5. Monitor beresolusi 1024x600 piksel.
6. *Keyboard* dan *mouse.*

3.1.5.2 Analisis Pengguna

Sistem yang akan dibangun digunakan oleh 1 jenis pengguna saja, karena memang aplikasi ini sengaja dibangun untuk *open user*. Pengguna dapat mengakses aplikasi tanpa perlu melakukan *Log in* terlebih dahulu, kemudian pengguna dapat mengakses semua menu yang tersedia yaitu : melihat materi tetang *state and space*, contoh soal, melakukan simulasi baik secara manual maupun secara otomatis terhadap beberapa kasus *State and Space* yang sudah disediakan oleh pembangun aplikasi, dan yang terpenting yaitu pengguna dapat melakukan *input problem-problem* baru dan solusi serta mengujinya secara otomatis maupun manual.

Tabel 3.10. Karakteristik pengguna

Pengguna	Hak akses	Tingkat keterampilan
User	Akses Semua Menu	Menguasai komputer dan paham tentang logika kecerdasan buatan.

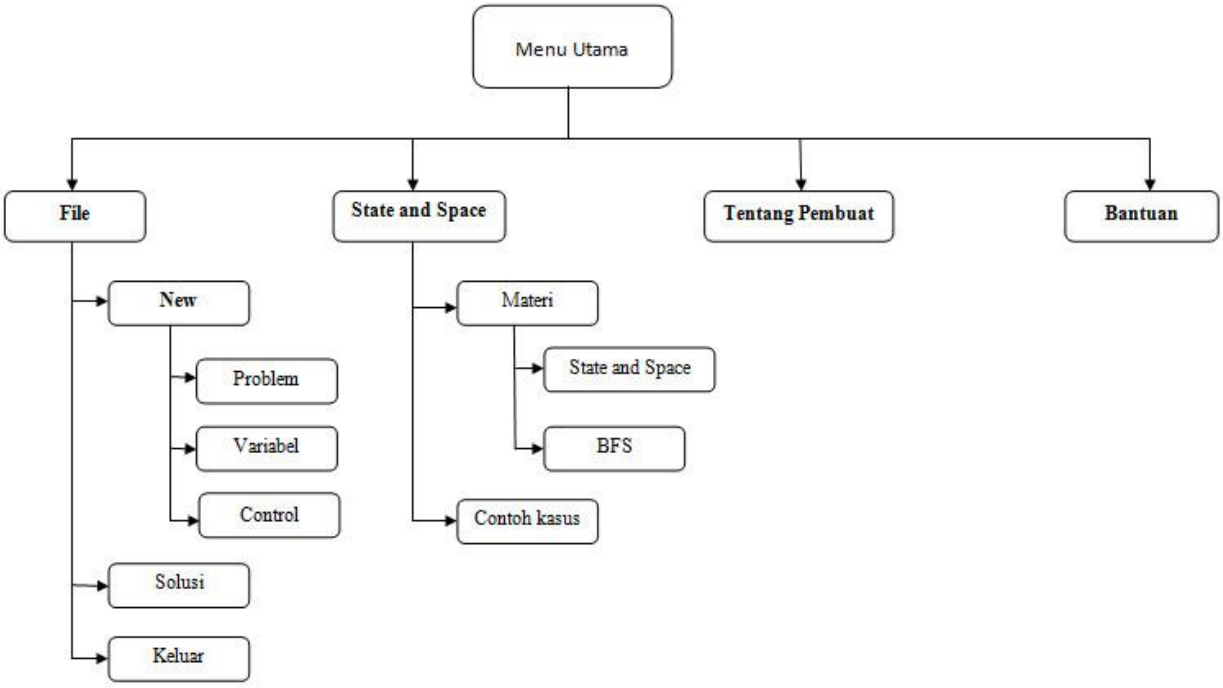
3.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan perancangan yang dibuat sebelum program aplikasi dibuat. Dengan perancangan akan mempermudah dalam proses

pembangunan sistem aplikasi. Perancangan perangkat lunak ini mengikuti konsep yang telah dilakukan pada tahap analisis sistem.

3.2.1 Perancangan Struktur Menu

Dalam perancangan sebuah aplikasi dibutuhkan struktur menu yang berisikan menu dan submenu yang berfungsi untuk memudahkan admin dalam menggunakan aplikasi tersebut. Berikut ini digambarkan mengenai stuktur menu dalam aplikasi ini, yaitu :



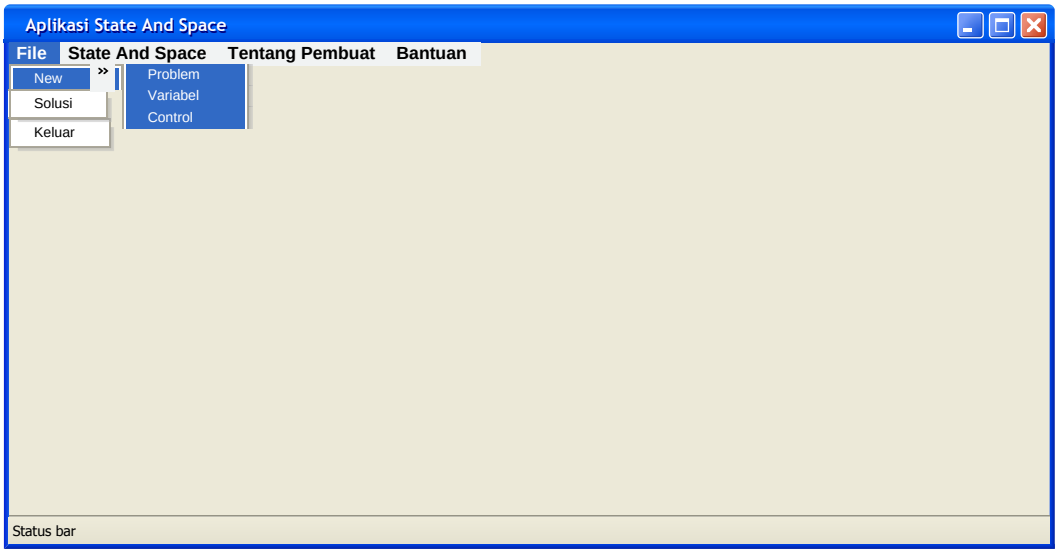
Gambar 3.5. Struktur Menu

3.2.2 Perancangan Antar Muka

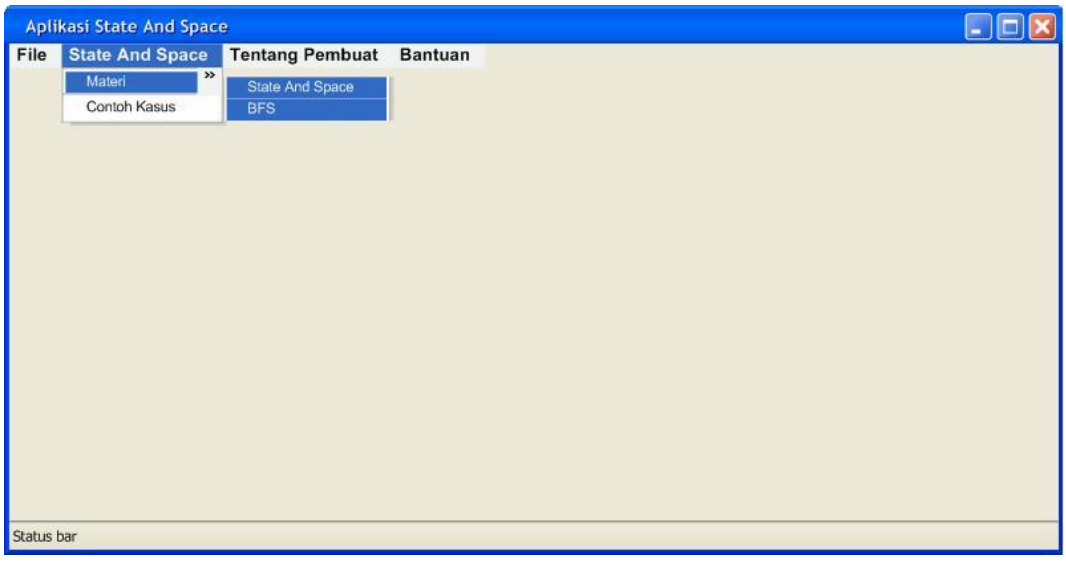
Perangkat lunak ini memiliki beberapa *form*, antara lain:

1. *Form* Utama.

Form utama merupakan tampilan awal dari program aplikasi. Menu berbentuk *toolbar*.



Gambar 3.6. Form Utama



Gambar 3.7. Form Utama

2. *Form* Pembelajaran, terdiri dari :
   - *Form* materi State and Space.

     Form ini merupakan *form* yang menampilkan materi pembelajaran dan pemahaman tentang *State and Space. Form* ini dapat diakses dari *form* utama, pilih menu *State and Space* →materi→*State and Space.*



Gambar 3.8. Form Materi State and Space

   - *Form* materi Pohon penelusuran dengan metode Breadth-First Search.

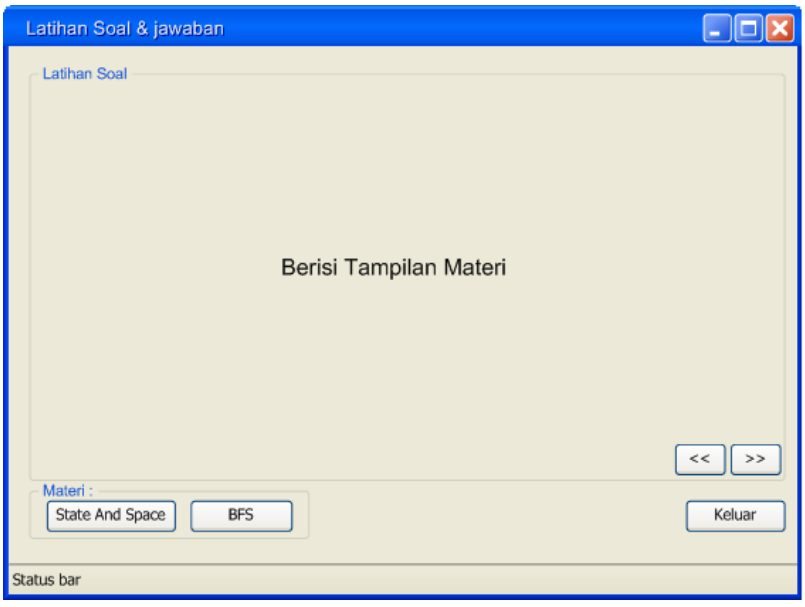
     Form ini merupakan *form* yang menampilkan materi pembelajaran dan pemahaman tentang Pohon Penelusuran dengan metode *Breadth-Firsth Search.*. *Form* ini dapat diakses dari *form* utama, pilih menu *State and Space* →materi→*BFS.*



Gambar 3.9. Form Materi BFS

- *Form* Latihan Soal dan Jawaban.

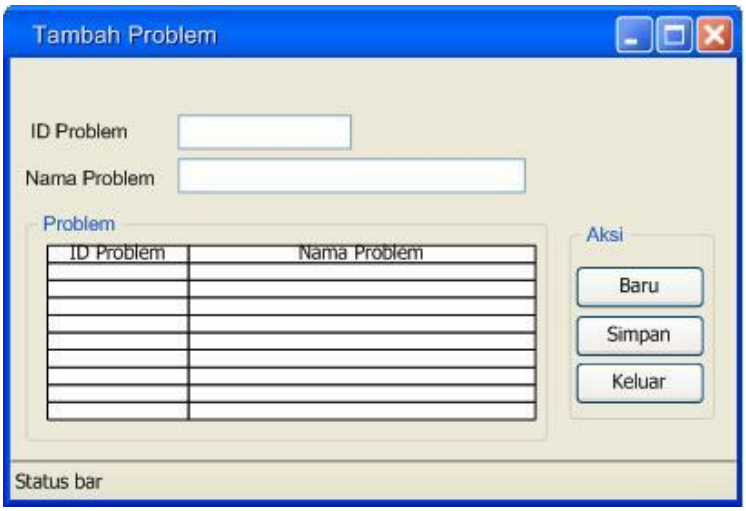
  Form ini merupakan *form* yang menampilkan contoh soal dan langkah penyelesaian *State and Space* secara teoritis. *Form* ini dapat diakses dari *form* utama, pilih menu *State and Space* → Contoh Kasus.



Gambar 3.10. Form Latihan Soal dan Jawaban

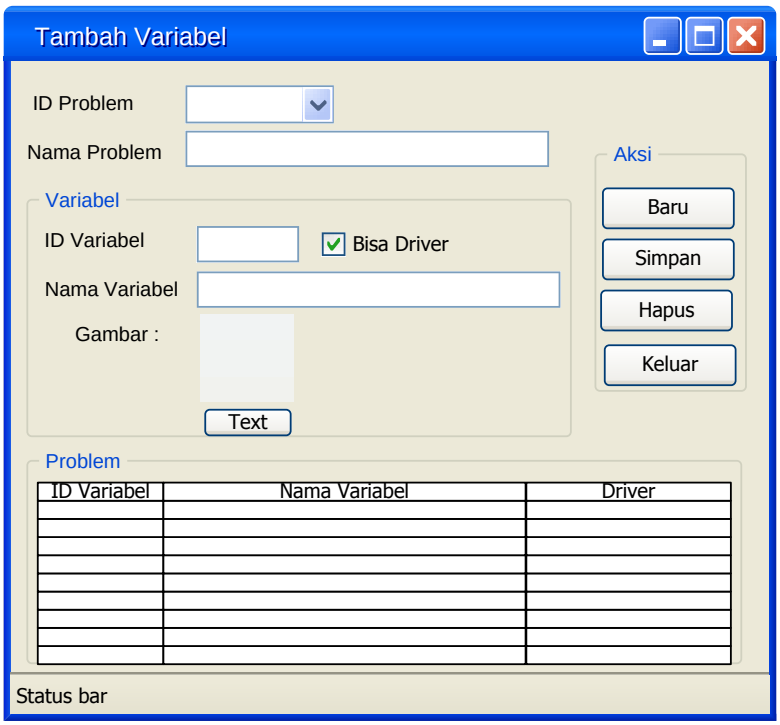
3. *Form Input Problem*.

   Form input problem merupakan form yang di pergunakan untuk memasukkan dan menambahkan permasalahan baru tentang *State and Space*. *Form* ini dapat diakses melalui menu utama, pilih menu File→New→*Problem.*



Gambar 3.11. Form input problem

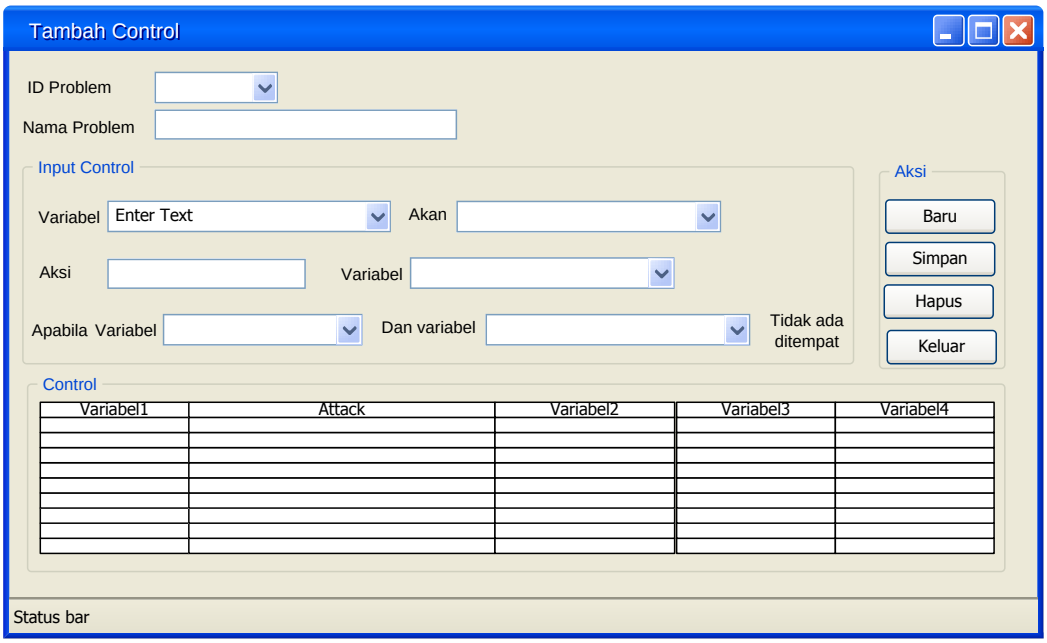
4. *Form input* variabel merupakan *form* yang di pergunakan untuk menambahkan, maupun mengedit variabel dari permasalahan yang sudah ada. *Form* ini dapat diakses melalui menu utama, pilih menu File→New→Variabel.



Gambar 3.12. Form input variabel

5. *Form Input Control*.

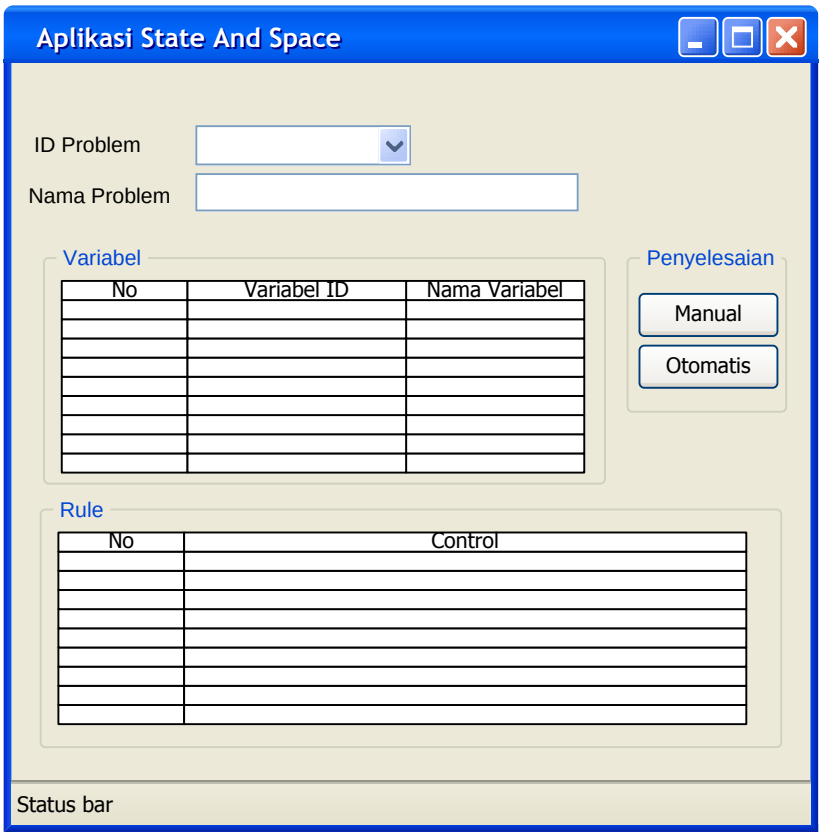
   Form input control merupakan *form* yang dipergunakan untuk menambahkan, maupun mengedit *control* dari permasalahan yang sudah ada. *Form* ini dapat diakses melalui menu utama, pilih menu File→New→*Control*.



Gambar 3.13. Form input control

6. *Form* Aplikasi *State and Space*

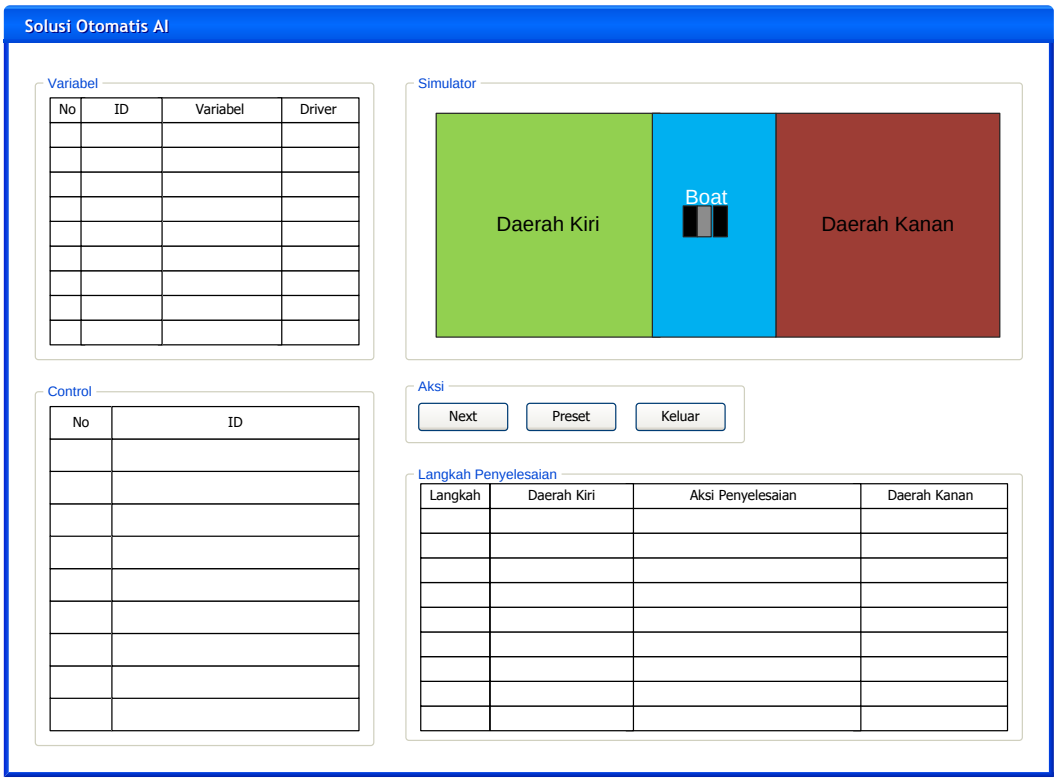
Form ini merupakan *form* yang dipergunakan untuk melakukan pencarian dan memilih serta melihat *problem state and space* yang sudah di *input*kan oleh *user* maupun *problem* yang telah ada dalam database, serta menampilkan,*variabel* dan *control* dari *problem*. *Form* ini memberikan menu langkah penyelesaian secara otomatis maupun secara manual. *Form* ini dapat diakses melalui *Form* utama, pilih menu File→Aplikasi.



Gambar 3.14. Form Aplikasi state and space

7. *Form* Penyelesaian Otomatis (Solusi *AI*).

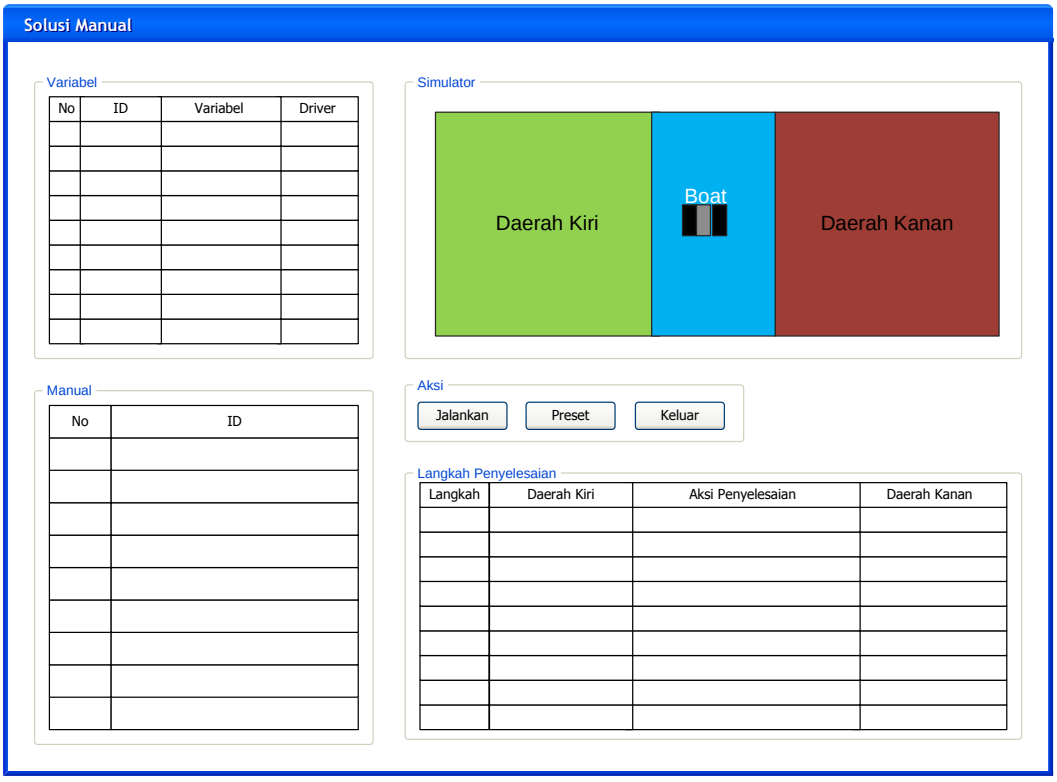
   Form penyelesaian otomatis merupakan *form* yang menampilkan langkah penyelesaian *problem* secara otomatis oleh komputer sehingga disebut dengan langkah penyelesaian dengan solusi *AI*. Form ini dapat diakses melalui *Form* Aplikasi dengan memilih *button* penyelesaian Otomatis.



Gambar 3.15. Form Penyelesaian Otomatis

8. *Form* Penyelesaian Manual.

   Form penyelesaian manual merupakan *form* yang menampilkan langkah penyelesaian *problem* secara manual, dimana *user* dapat menjalankan langkah penyelesaian dari *problem* yang telah dipilih secara manual dengan menjalankan variabel secara manual. Form ini dapat diakses melalui *Form* Aplikasi dengan memilih *button* penyelesaian Manual.



Gambar 3.16. Form Penyelesaian Manual

9. *Form* Tentang Pembuat.

   Form Tentang Pembuat menampilkan tentang pembuat aplikasi.



Gambar 3.17. Form Tentang Pembuat

10. *Form* Bantuan.

    Form bantuan menampilkan tentang cara dan petunjuk pemakaian aplikasi.



Gambar 3.18. Form Bantuan