

# UNIVERSITAS GUNADARMA



## PRAKTIKUM PENGANTAR KECERDASAN BUATAN

### MANUAL BOOK

#### *“GAME : SENGOKU LUDO”*

#### MENGGUNAKAN STRAWBERRY PROLOG

<b>Nama</b>	<b>: Syaiful Haq Al Furuqi</b>
<b>NPM</b>	<b>: 5A413515</b>
<b>Kelas</b>	<b>: 3IA26</b>
<b>Fakultas</b>	<b>: Teknik Industri</b>
<b>Jurusan</b>	<b>: Teknik Informatika</b>
<b>PJ</b>	<b>: Aziz Sudrajat</b>

**Ditulis Guna Melengkapi Sebagian Syarat**

**Praktikum Pengantar Kecerdasan Buatan Jenjang S1**

**Universitas Gunadarma**

**2016**

## Daftar Isi

Daftar Isi .....	I
1. Initial State .....	1
2. Konsep AI .....	2
3. Konsep Permainan.....	3
4. Flowchart.....	3
5. Algoritma .....	10
6. Listing Program.....	11
7. Daftar Pustaka .....	II

## 1. INITIAL STATE

Ludo adalah salah satu permainan komputer yang menggunakan konsep kecerdasan buatan. Ludo adalah permainan dimana pemain harus menjalankan semua bidak pada sarang awal menuju sarang akhir. Dimana bidak tersebut melangkah sesuai dengan dadu yang dikocok. Bidak tersebut dapat keluar dari sarang awal menunjukkan angka 6. Kecerdasan buatan yang dibuat disini adalah setiap pengocokan dadu akan bersifat acak dan pemain bermain dengan melawan komputer

Game ini dapat dimainkan oleh semua kalangan dari anak-anak hingga orang dewasa untuk melatih kecerdasan atau strategi pemain dalam menjalankan suatu bidak. Game ini biasanya dimainkan di suatu kertas atau karton, karena teknologi sudah semakin maju dan sudah jarang sekali masyarakat memainkan game ini maka kami membuatnya menggunakan media komputer ataupun laptop. Karena media tersebut sudah banyak digunakan oleh masyarakat dan jika menggunakan laptop maka pengguna dapat memainkan game ini dimanapun dan kapanpun mereka ingin bermain game ludo ini. Pembuatan aplikasi ini menggunakan Bahasa Pemrograman Strawberry Prolog.

### SPLASH SCREEN



Interface game ini diawali dengan splash screen loading.

## MENU UTAMA



Setelah loading bar selesai pada Splash Screen akan muncul window menu untuk memilih 'mulai' atau 'keluar' untuk keluar dari game tersebut.

## WINDOW UTAMA PERMAINAN



## 2. Konsep AI

Dalam permainan ludo, pemain dan lawan (komputer) memainkan masing - masing 4 bidak dalam permainan ini. Bidak berjalan mengikuti blok-blok yang tersusun memutar seluruh area permainan dari tempat awal menuju tempat akhir. Tempat awal dan tempat akhir itu berada pada area masing – masing pemain.

Penyelesaian :

1. Identifikasi ruang keadaan

Permasalahan ini dapat dilambangkan dengan (jumlah jalur 80 blok, 4 bidak pemain , 4 bidak lawan, 1 buah dadu).

2. Keadaan awal & tujuan

Keadaan Awal = 4 buah bidak berada di sarang masing-masing pemain, jalur ludo, dadu yang teracak.

Keadaan Tujuan = bidak – bidak sampai di tempat akhir

### 3. Aturan-aturan

- a. Dadu harus muncul pada angka 6 untuk mengeluarkan bidak.
- b. Bidak harus melewati jalur yaitu dengan memutari seluruh area permainan.
- c. Bidak tidak boleh berada pada blok yang sama dengan bidak lawan, karena bidak akan kembali ke sarang.
- d. Pemain mendapat kesempatan untuk melempar dadu lagi apabila muncul angka 6 pada lemparan pertama.

## 3. Konsep Permainan

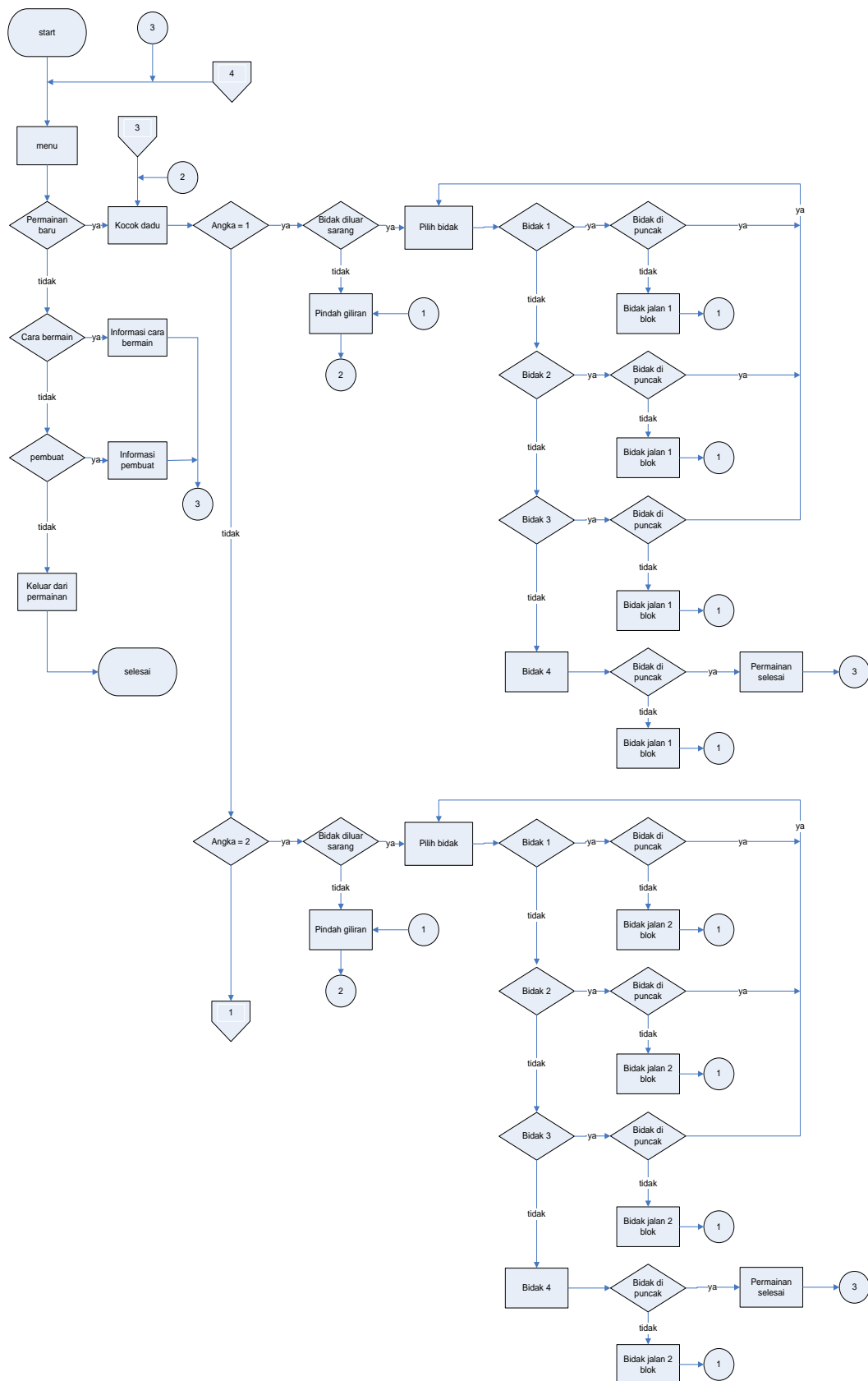
Konsep permainan yang diterapkan dalam permainan ludo ini adalah pemain dan lawan harus menjalankan masing – masing bidak mengikuti blok – blok yang tersusun memutari seluruh area permainan. Bidak pemain tidak boleh berada pada satu blok yang sama dengan bidak lawan, karena bidak yang tertiban akan kembali ke sarang bidak dan mengulang dari tempat awal.

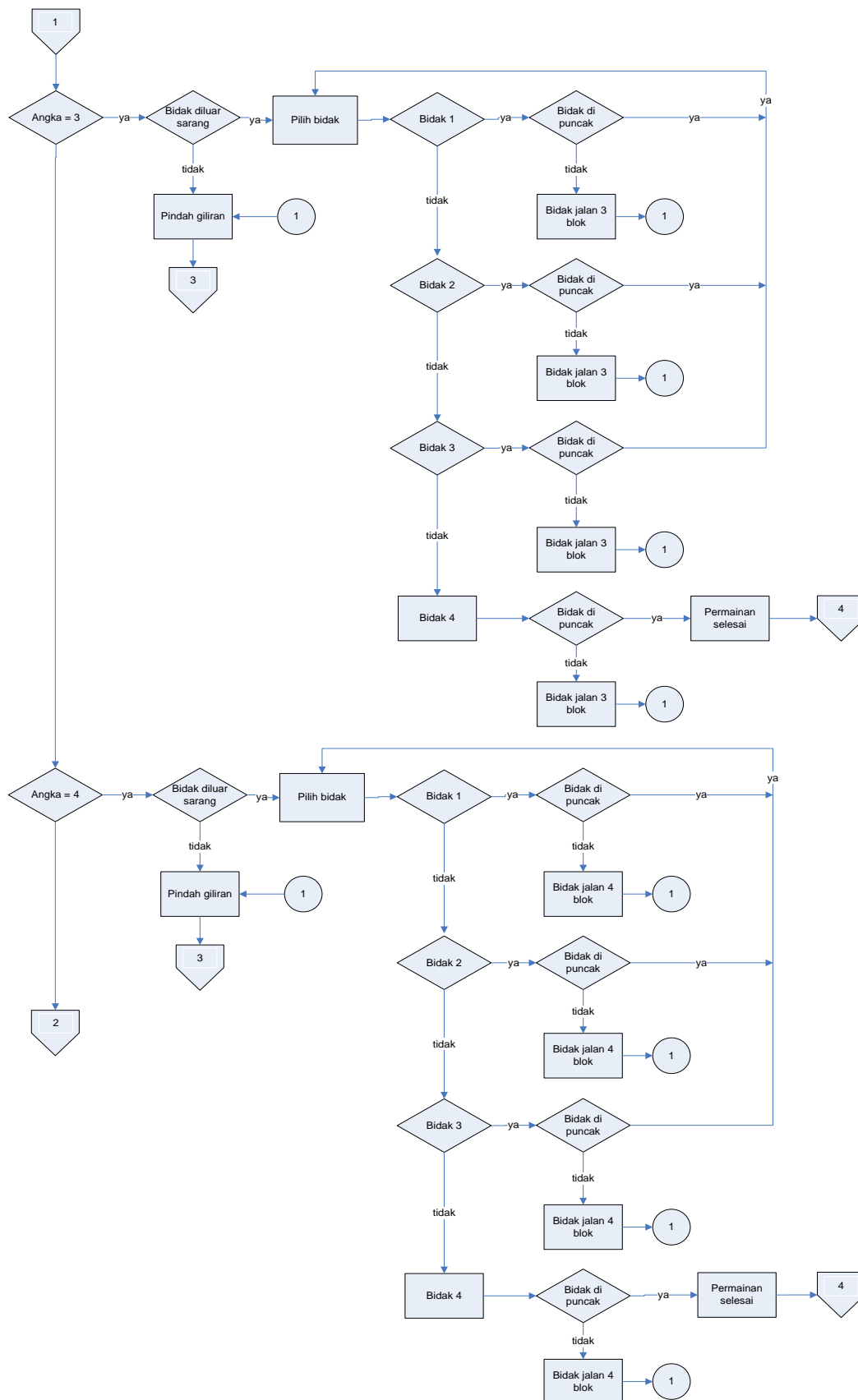
Terdapat jalur yang tersusun dari blok-blok sebagai tempat jalur bidak berjalan. Tempat awal bidak, ketika bidak keluar dari sarang yaitu blok pertama yang berada di sebelah kanan dari tempat akhir bidak. Tempat akhir bidak adalah tujuan akhir bidak setelah bidak memutari seluruh area permainan, dan terdapat satu blok yang menjadi puncak tempat akhir. Pada puncak ini bidak berhenti dan tidak dapat dijalankan lagi. Tempat akhir ini ditandai dengan blok yang berwarna sama dengan bidak. Tempat akhir ini hanya dilalui oleh masing – masing bidak pemain.

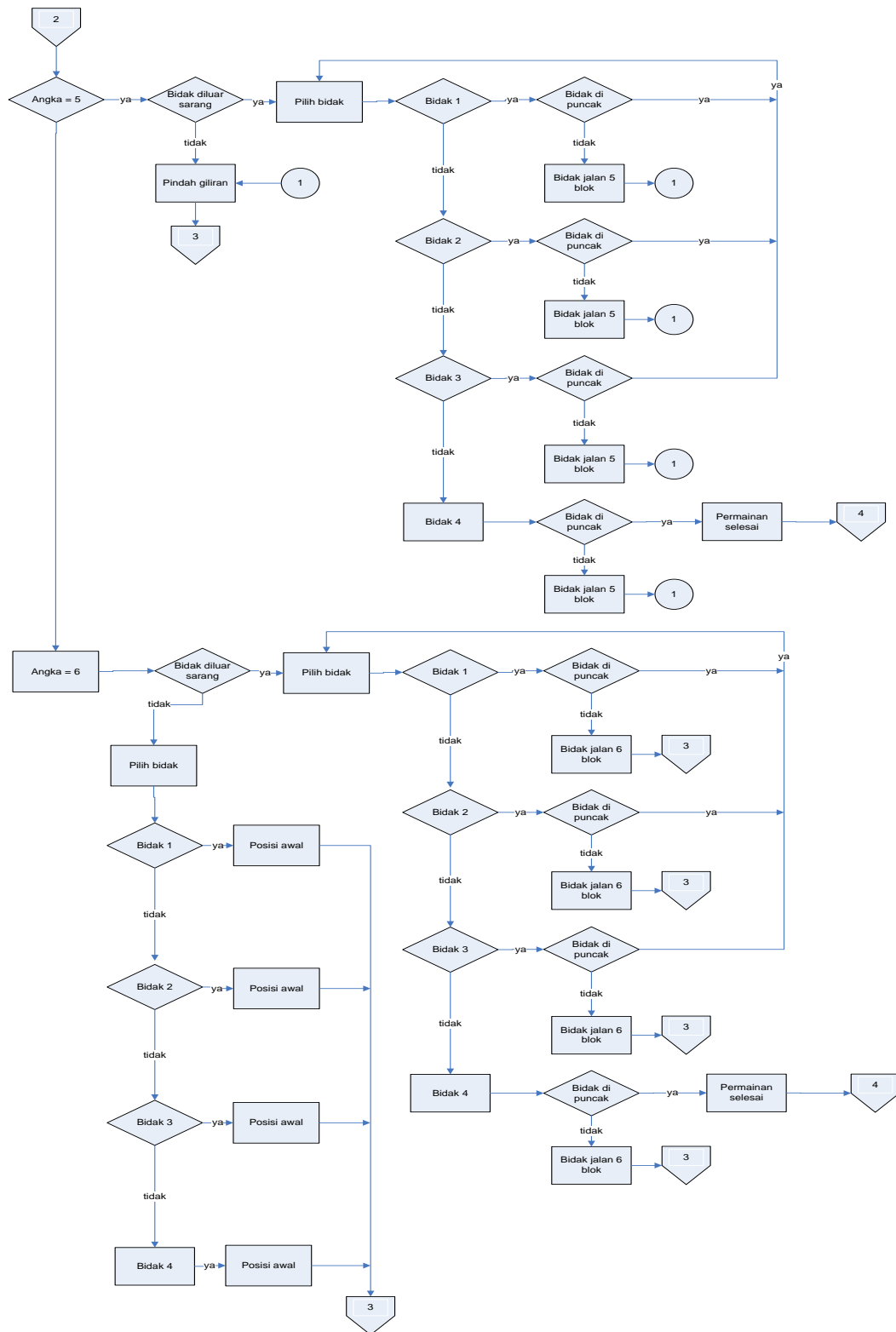
Untuk menjalankan bidak tersebut pemain harus mengeluarkannya terlebih dahulu dari “sarang” bidak yaitu dengan mengocok dadu hingga muncul angka 6. Jalannya bidak sesuai dengan angka yang muncul pada dadu dan melewati jumlah blok yang muncul pada angka dadu tersebut. Jika pada lemparan dadu muncul angka 6 maka pemain mendapatkan kesempatan lagi untuk mengocok dadu. Keempat bidak tersebut dapat dijalankan bersamaan atau satu persatu apabila telah keluar dari sarangnya. Apabila keempat bidak tersebut telah sampai di puncak tempat akhir maka pemain memenangkan permainan ini.

## 4. Flowchart

Berikut ini adalah flowchart dari permainan ludo :





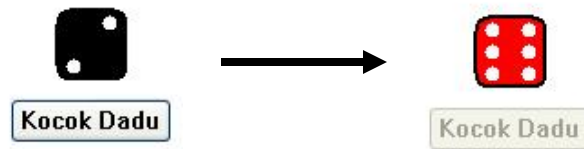


*Gambar 2.1. Flowchart pada permainan Ludo*

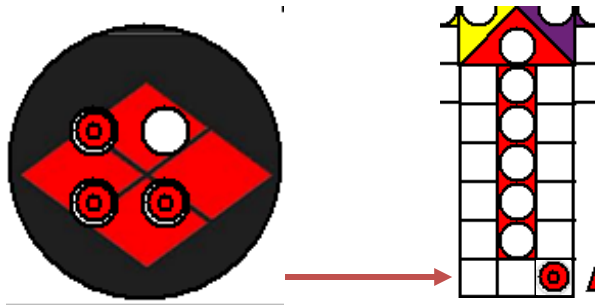
Diagram diatas adalah diagram untuk menentukan jalannya bidak –bidak yang terdapat pada permainan ludo ini. Penjelasan alur untuk jalannya bidak adalah sebagai berikut :



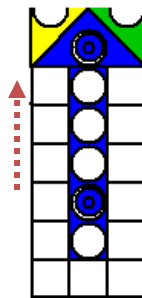
- Pada awal permainan, pemain diberikan kesempatan untuk mengocok dadu. Dadu akan mengacak angka untuk mencari angka enam yang terdapat pada dadu.



- Jika angka yang muncul pada dadu belum menunjukkan angka enam maka bidak tidak dapat keluar dari sarang, dan harus melakukan kocok dadu.



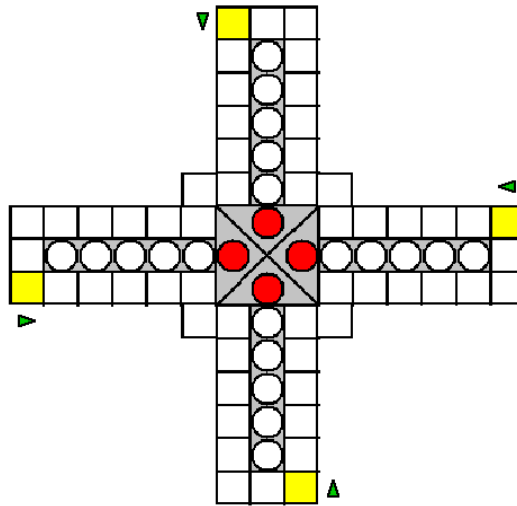
- Jika angka enam muncul pemain dapat memilih bidak untuk keluar dari sarangnya. Kemudian untuk angka enam pemain mendapatkan kesempatan untuk mengocok dadu lagi. Pada garis putus - putus adalah perpindahan pada bidak setelah dadu dikocok lagi dan bidak keluar dari sarang. Dadu akan berjalan pada jalur yang tersusun atas blok dan sesuai jumlah angka dadu yang muncul pada saat mengocok dadu.



- Bidak berada pada area tempat akhir setelah sekali memutar jalur yang tersusun atas blok. Setelah sampai di tempat akhir bidak harus masuk ke puncak tempat akhir. Jika sampai di puncak tempat akhir bidak tidak dapat dijalankan lagi.

## Rancangan Tampilan

Berikut ini merupakan rancangan tampilan untuk papan ludo yang digunakan :

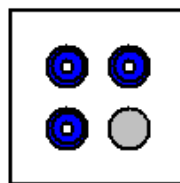


*Gambar 2.2. Rancangan Tampilan Papan Ludo*

Keterangan gambar :

- Papan terdiri dari 80 kotak yang pada penulisan ini disebut blok.
- Blok tersebut berfungsi sebagai jalur bidak.
- Blok yang berwarna kuning adalah tempat awal untuk posisi bidak yang keluar dari sarang pada masing – masing area.
- Blok berupa lingkaran yang terdapat pada bagian tengah pada tiap area adalah tempat akhir bagi bidak.
- Blok berupa lingkaran yang berwarna merah merupakan puncak dari tempat akhir
- Segitiga berwarna hijau merupakan arah jalan untuk jalur bidak.

Berikut ini merupakan rancangan tampilan untuk sarang bidak dan bidak yang digunakan :



*Gambar 2.3. Rancangan Sarang bidak dan bidak*

Keterangan gambar :

- Gambar kotak besar yang berwarna dasar putih merupakan sarang untuk menyimpan bidak.
- Gambar bulat yang berwarna biru merupakan bidak – bidak yang masih tersimpan disarang bidak.
- Lingkaran kecil berwarna abu – abu merupakan bidak yang keluar dari sarang.

Berikut ini merupakan rancangan tampilan untuk dadu yang digunakan :

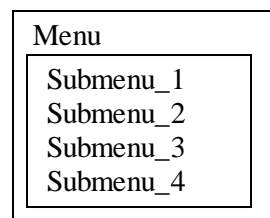


Gambar 2.4. Rancangan Dadu

Keterangan gambar :

- Gambar kotak hitam merupakan dadu yang digunakan.
- Terdapat 6 angka pada dadu.
- Tombol Kocok Dadu digunakan untuk mengocok dadu.

Berikut ini merupakan rancangan untuk menu yang digunakan pada permainan ludo, yaitu *pop-up menu* :



Gambar 2.5. Rancangan Pop-Up Menu

Keterangan gambar :

- Submenu\_1 untuk “Permainan Baru” yaitu memulai permainan dari awal.
- Submenu\_2 untuk “Cara Bermain” yaitu informasi cara permainan ini.
- Submenu\_3 “Pembuat” untuk informasi pembuat permainan.
- Submenu\_4 “Keluar” untuk keluar dari game.

## Tampilan Output

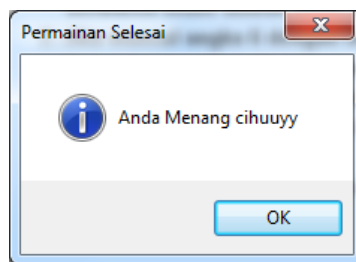


*Gambar 2.6. Tampilan Output*

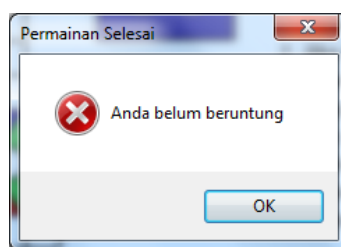
Pada permainan ini user yang memiliki area berwarna merah sedangkan lawan (komputer) berwarna biru. Permainan dimulai dengan menekan tombol Kocok Dadu. Dadu akan teracak dan akan berhenti dengan menampilkan angka. Jika angka yang muncul bukan angka enam maka kesempatan untuk mengocok akan berpindah ke lawan, kemudian lawan melakukan hal yang sama. Jika angka enam muncul, pemain dapat memilih salah satu bidak yang berada pada sarang untuk keluar.

Kemudian bidak akan berada pada posisi awal dan pemain mendapatkan kesempatan untuk mengocok dadu lagi karena mendapatkan angka enam. Setelah kocokan dadu muncul bidak yang berada posisi awal dapat dipilih untuk berjalan memutar seluruh area atau lintasan ludo.

Jika keempat bidak sudah memutar seluruh area dan berada pada puncak yang terdapat pada tempat akhir, maka akan dinyatakan sebagai pemenang.



*Gambar 2.13. Tampilan informasi jika pemain menang*



*Gambar 2.14. Tampilan informasi jika pemain kalah*

## **5. Algoritma**

Penulisan ini akan membahas implementasi algoritma Greedy dalam pemodelan AI dalam permainan Pacman.

Algoritma *greedy* merupakan metode yang paling populer untuk memecahkan masalah optimasi. Algoritma *greedy* membentuk solusi langkah per langkah. Pendekatan yang digunakan di dalam algoritma *greedy* adalah membuat pilihan yang tampak memberi perolehan

terbaik, yaitu dengan membuat pilihan optimum lokal pada setiap langkah dengan harapan akan mengarah ke solusi optimum global.

Prinsip algoritma *greedy* pada setiap langkah ialah mengambil pilihan terbaik yang dapat diperoleh saat itu tanpa memperhatikan konsekuensi ke depan, dan berharap bahwa dengan memilih optimum lokal pada setiap langkah akan menghasilkan optimum global pada akhir proses.

Persoalan optimasi algoritma *greedy* disusun oleh elemen-elemen berikut :

1. Himpunan kandidat, yang berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi, berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
3. Fungsi seleksi, dinyatakan dengan predikat SELEKSI memilih kandidat yang paling memungkinkan mencapai solusi optimal pada setiap langkah.
4. Fungsi kelayakan, dinyatakan dengan predikat LAYAK, memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak dengan tidak melanggar *constraints* yang ada.
5. Fungsi objektif, yang memaksimumkan atau meminimumkan nilai solusi.

Prinsip algoritma *greedy* pada setiap langkah ialah mengambil pilihan terbaik yang dapat diperoleh saat itu tanpa memperhatikan konsekuensi ke depan, dan berharap bahwa dengan memilih optimum lokal pada setiap langkah akan menghasilkan optimum global pada akhir proses.

## 6. Listing Program

```
%*****
%*****
%*****%
%          nama      :
Syaiful Haq Al Furuqi
%
%          npm       :
5A413515
%
%          kelas    :
3IA26
%
%          GAME
SENGOKU LUDO
%
%*****
%*****
%*****%
?-
```

```
set(pos([0,0,0,0],[0,0,0,0],
2)),
    G_Flag=0,
    G_Blue is
brush(rgb(0,0,255)),
    G_Red is
brush(rgb(255,0,0)),

    G_BlueL is
brush(rgb(170,170,255)),
    G_RedL is
brush(rgb(255,150,150)),

    G_Yellow is
brush(rgb(255,255,0)),
    G_Purple is
brush(rgb(108,34,121)),
    G_White is
brush(rgb(255,255,255)),
```

```

    G_Black is
brush(rgb(0,0,0)),
    G_Pen is
pen(2,rgb(0,0,0)),
    G_PenW is
pen(2,rgb(255,255,255)),
    G_X := 138,

% Splash Screen %

window(G_splash,_,window_spl
ash(_),"SENGOKU LUDO",
350,100,435,440).
window_splash(init):-
    G_batas :=0,
    G_waktu2 is
set_timer(_,0,timer_splash).

window_splash(paint):-
    draw_bitmap(0,0,bitmap_
image("images//back0.bmp",_)
,_,_).

% TIMER SPLASH SCREEN (untuk
mengatur loading bar dan
timer splash screen) %

timer_splash(end):-
    text_out(138,200,print(
"LOADING : " + G_batas + "
%")),
    pen(5,rgb(128,128,255))
,
    brush(rgb(255,255,255))
,
    rect(133,225,240,260),
    pen(0,rgb(0,0,0)),
    brush(rgb(0,0,0)),
    rect(138,230,G_X,256),
    G_batas:=G_batas+1,
    G_X := G_X + 1,
    (G_batas=100-
>close_window(G_splash),
    window(G_menu,_,window_
menu(_),"SENGOKU LUDO",
350,100,435,440)).

window_menu(init):-
    button(_,_,tombol_1(_),
"MULAI",138,200,120,50),

```

```

    button(_,_,tombol_2(_),
"KELUAR",138,280,120,50).

window_menu(paint):-
    draw_bitmap(0,0,bitmap_
image("images//back0.bmp",_)
,_,_).

% BUTTON %
% BUTTON WINDOW AWAL %
tombol_1(press):-
    close_window(G_menu),
    window(G_pertama,_,win_
funk(_),"SENGOKU LUDO",
350,100,710,465),
    update_window(_).

% BUTTON WINDOW INSTRUKSI %
tombol_2(press):-
    close_window(G_menu).

% UTAMA %

win_funk(init):-

button(G_Button,_,but_funk(
_), "Kocok
Dadu",430,345,90,25),

static(_,_,fail(_),"Perhatia
n!! Cara Main
:",420,10,250,20),
    static(_,_,fail(_),"1.
Kocok dadu hingga muncul
angka 6",420,30,250,20),
    static(_,_,fail(_),"dadu
yang berwarna merah
adalah",425,50,250,20),
    static(_,_,fail(_),"dadu
pemain sedangkan dadu
yang",425,70,250,20),

static(_,_,fail(_),"berwarna
hitam adalah dadu
lawan.",425,90,250,20),
    static(_,_,fail(_),"2.
Jika muncul angka 6 dengan
dadu",410,110,250,20),

static(_,_,fail(_),"berwarna

```

```

merah maka jalankan
bidak",430,130,250,20),
    static(_,_,fail(_),"dengan
cara klik bidak yang
ingin",418,150,250,20),
    static(_,_,fail(_),"kita
jalankan,kemudian kocok
lagi",420,170,250,20),

```

```

static(_,_,fail(_),"dadunya.
",410,190,110,20),
    static(_,_,fail(_),"3.
Jalankan bidak hingga semua
bidak",417,210,250,20),
    static(_,_,fail(_),"sampai
di sarang
akhir.",387,230,250,20),
    menu( pop_up, _, _ ,
menu_file(_), "&File"),
    execute("sound//bb.exe"),
    put_buttons,
    put_mines.

```

```

win_funk(paint):-
    pen(G_Pen),
    brush(G_White),
    rect(135,135,260,260),
    rect(160,10,235,385),
    rect(10,160,385,235),
    fail.

```

```

win_funk(paint):-
    for(I,0,4),
    X is 25*I+35,
    line(160,X,235,X),
    fail.

```

```

win_funk(paint):-
    for(I,0,4),
    X is 25*I+260,
    line(160,X,235,X),
    fail.

```

```

win_funk(paint):-
    for(I,0,4),
    X is 25*I+35,
    line(X,160,X,235),
    fail.

```

```

win_funk(paint):-
    for(I,0,4),
    X is 25*I+260,
    line(X,160,X,235),
    fail.

```

```

win_funk(paint):-
    brush("images\\oda.bmp"),
    ellipse(0,0,135,135),
    brush(G_Yellow),
    rect(34,184,161,211),

fill_polygon(160,160,197,197
,160,235),

```

```

fill_polygon(15,243,30,248,1
5,253),

```

```

brush("images\\takeda.bmp"),
    ellipse(0,265,135,400),
    brush(G_Red),
    rect(184,234,211,361),

```

```

fill_polygon(160,235,197,197
,235,235),

```

```

fill_polygon(243,380,248,365
,253,380),

```

```

    brush("images\\date.bmp"),
    ellipse(270,0,405,135),
    brush(G_Blue),
    rect(184,34,211,161),

```

```

fill_polygon(160,160,197,197
,235,160),

```

```

fill_polygon(143,15,148,30,1
53,15),

```

```

    brush("images\\azai.bmp"),
    ellipse(270,265,405,400),
    brush(G_Purple),
    rect(234,184,361,211),

```

```

fill_polygon(235,160,197,197
,235,235),

```

```

fill_polygon(380,143,365,148
,380,153),

```

```

pos(_,_,Z),
    (G_Flag=0, Z<6 ->
        brush(G_Black)
    else
        brush(G_Red)

```

```

),
round_rect(455,295,495,335,1
5,15),
brush(G_White),
fail.
win_funk(paint):-
for(A,0,1),
for(B,0,1),
X is 185+25*A,
Y is 10 + 350*B,
Y1 is Y+25,
line(X,Y,X,Y1),
line(Y,X,Y1,X),
for(I,0,1),
for(J,0,1),

ellipse(275*A+35*I+30,275*B+
35*J+30,275*A+35*I+55,275*B+
35*J+55),
fail.
win_funk(paint):-
for(I,0,5),

ellipse(185,35+25*I,210,60+2
5*I),

ellipse(185,210+25*I,210,235
+25*I),

ellipse(35+25*I,185,60+25*I,
210),

ellipse(210+25*I,185,235+25*
I,210),
fail.
win_funk(paint):-

pos([B0,B1,B2,B3],[R0,R1,R2,
R3],Z),
pen(G_PenW),
draw_points(Z),

put_pionka(0,0,B0,[B0,B1,B2,
B3]),

put_pionka(1,0,B1,[B0,B1,B2,
B3]),

put_pionka(2,0,B2,[B0,B1,B2,
B3]),

put_pionka(3,0,B3,[B0,B1,B2,
B3]),

put_pionka(0,1,R0,[R0,R1,R2,
R3]),

put_pionka(1,1,R1,[R0,R1,R2,
R3]),

put_pionka(2,1,R2,[R0,R1,R2,
R3]),

put_pionka(3,1,R3,[R0,R1,R2,
R3]),
fail.

win_funk(mouse_click(X,Y)):-
G_Flag=1,
cursor(w),
X1 is 10+25* ((X-
10)//25),
Y1 is 10+25* ((Y-
10)//25),
try_all(P,I),
koordinati(P,1,I,X2,Y2),
abs(X1-X2+6)<12,
abs(Y1-Y2+6)<12,
pos(B,R,Z),
member(R,I,P),
(I=0, Z=6->
S is 1
else (I>0, I+Z=<62->
S is I+Z
else
fail
)),
replace(R2,R,S,P),
play(1,S,I,P,B,R2,Z),
(R2=[62,62,62,62]->
G_Flag:=2,
set(pos(B,R2,Z)),
message("Permainan
Selesai","Anda Menang
cihuuyy",i),
set_text("SENGOKU LUDO
SELESAI" , G_Main),
fail
),
(S>0, S<57->
S2 is 1+ (S+27) mod 56,

```



```

        kick(B2,B,S2,0)
    else
        B2=B
    ),
    (Z<6->
        answer(B2,R2)
    else
        set(pos(B2,R2,Z))
    ),
    G_Flag:=0,
    enable_window(G_Button,
1),
    set_text("Kocok Dadu" ,
G_Main),
    !, fail.
win_funk(mouse_click(X,Y)):-
    beep.
but_funk(press):-
    G_Flag=0,
    cursor(w),
    enable_window(G_Button,
0),
    pos(B,R,_),
    Z is 1+random(6),
    pen(G_Pen),
    brush(G_Red),

round_rect(455,295,495,335,1
5,15),
    wait(0.5),
    brush(G_White),
    pen(G_PenW),
    draw_points(Z),
    (member(R,S,_), (S>0,
S+Z=<62;S=0, Z=6)->
        set(pos(B,R,Z)),
        G_Flag:=1,
        set_text("SENGOKU LUDO"
, G_Main)
    else
        answer(B,R),
        enable_window(G_Button,
1)
    ),
    !, fail.
but_funk(press):-
    beep.
answer(B,R):-
    wait(1),
    Z is 1+random(6),
    pen(G_Pen),

```

```

    brush(G_Black),

round_rect(455,295,495,335,1
5,15),
    wait(0.6),
    brush(G_White),
    pen(G_PenW),
    draw_points(Z),
    wait(1),

choose_answer(B,R,Z,I,P,S),
    replace(B2,B,S,P),
    play(0,S,I,P,R,B2,Z),
    (B2=[62,62,62,62]->
        G_Flag:=2,
        set(pos(B2,R,Z)),
        message("Permainan
Selesai","Anda belum
beruntung",s),
        set_text("SENGOKU LUDO
SELESAI" , G_Main),
        fail
    ),
    (S>0, S<57->
        S2 is 1+ (S+27) mod 56,
        kick(R2,R,S2,1)
    else
        R2=R
    ),
    (Z=6->
        answer(B2,R2)
    else
        set(pos(B2,R2,Z))
    ).
answer(B,R):-
    G_Flag<2,
    G_Flag:=0,
    enable_window(G_Button,
1),
    set_text("Kocok Dadu" ,
G_Main).
choose_answer(B,R,Z,I,P,S):-
    member(B,I,P),
    I>0,
    I+Z=<56,
    S is I+Z,
    S2 is 1+ (S+27) mod 56,
    member(R,S2,_).
choose_answer(B,R,6,0,P,1):-
    member(B,0,P).
choose_answer(B,R,Z,I,P,S):-

```

```

member(B,I,P),
I>0, I<57,

count_successes(member(B,I,_))>1,
S is I+Z.
choose_answer(B,R,Z,I,P,S):-
maximum(I2,X,P2,call(member(B,X,P2), X>0, X<57)),
I2>0,
I=I2,
S is I+Z,
P=G_P_sys.
choose_answer(B,R,Z,I,P,S):-
member(B,I,P),
I>0,
I+Z=<62,
S is I+Z.
choose_answer(B,R,Z,I,P,S):-
set(pos(B,R,Z)),
fail.
maximum(R,X,P,Pr):-
G_Max_sys:=0, fail.
maximum(R,X,P,Pr):- Pr,
X>G_Max_sys, G_Max_sys:=X,
G_P_sys:=P, fail.
maximum(R,X,P,Pr):-
R:=G_Max_sys.

kick(L3,L,S,C):-
member(L,S,P),
replace(L2,L,0,P),
koordinati(P,C,0,X,Y),
pionka(X,Y,C,L,0),
kick(L3,L2,S,C).
kick(L,L,S,C).

play(C,S,I,P,B,R,Z):-
for(S1,I,S-1),
koordinati(P,C,S1,X,Y),
pionka(X,Y,C,R,S1),
wait(0.2),
del_pionka(R,B,S1,X,Y,C),
fail.
play(C,S,I,P,B,R,Z):-
koordinati(P,C,S,X,Y),
pionka(X,Y,C,R,S).

try_all(P,0):-for(P,0,3).
try_all(_,I):-for(I,1,61).

put_pionka(P,C,I,CL):-
koordinati(P,C,I,X,Y),
pionka(X,Y,C,CL,I).

pionka(X,Y,Br,CL,I):-
(I>0,
count_successes(member(CL,I,_))>1->
Br2 is Br+2,
my_brush(Br2)
else
my_brush(Br)
),
pen(G_Pen),
ellipse(X+3,
Y+2,X+23,Y+22),
ellipse(X+8
,Y+8,X+17,Y+17).

del_pionka(B,R,S,X,Y,C):-
S>0, member(B,S,_), !,
pionka(X,Y,C,B,S).
del_pionka(B,R,S,X,Y,C):-
S>0, S<57, S2 is 1+(S+27)
mod 56,
member(R,S2,_), !,
C2 is 1-C,
pionka(X,Y,C2,R,S2).
del_pionka(B,R,S,X,Y,C):-
brush(G_White),
pen(G_PenW),
ellipse(X+3,
Y+2,X+23,Y+22).

my_brush(0):-brush(G_Blue).
my_brush(1):-brush(G_Red).
my_brush(2):-brush(G_BlueL).
my_brush(3):-brush(G_RedL).

koordinati(P,C,0,X,Y):- X is
275*(1-C)+35*(P mod 2)+30,
Y is 275*C+35*(P // 2)+30.
koordinati(P,C,N,X,Y):-
N>0, N<57,
(C=0->
koordinati2(N,X,Y)
else
N2 is 1+(N+27) mod 56,
koordinati2(N2,X,Y)
).

```

```

koordinati(_,0,N,X,Y):-
N>56, X=185, Y is 10 + (N-
56)*25.
koordinati(_,1,N,X,Y):-
N>56, X=185, Y is 360 - (N-
56)*25.

koordinati2(N,X,Y):- N>0,
N<7, X=160, Y is N*25- 14.
koordinati2(7,136,136).
koordinati2(N,X,Y):- N>7,
N<14, Y=161, X is 335 -
N*25.
koordinati2(14,10,186).
koordinati2(N,X,Y):- N>14,
N<21, Y=210, X is -365 +
N*25.
koordinati2(21,136,235).
koordinati2(N,X,Y):- N>21,
N<28, X=160, Y is N*25- 315.
koordinati2(28,185,360).
koordinati2(N,X,Y):- N>28,
N<35, X=209, Y is 1085-N*25.
koordinati2(35,234,234).
koordinati2(N,X,Y):- N>35,
N<42, Y=210, X is -666 +
N*25.
koordinati2(42,359,186).
koordinati2(N,X,Y):- N>42,
N<49, Y=161, X is 1434 -
N*25.
koordinati2(49,234,137).
koordinati2(N,X,Y):- N>49,
N<56, X=209, Y is 1386-N*25.
koordinati2(56,185,11).

draw_points(6):-
    ellipse(461,311,469,319),
    ellipse(481,311,489,319),
    ellipse(461,323,469,331),
    ellipse(481,323,489,331),
    ellipse(461,299,469,307),
    ellipse(481,299,489,307).
draw_points(5):-
    ellipse(461,323,469,331),
    ellipse(481,323,489,331),
    ellipse(461,299,469,307),
    ellipse(481,299,489,307),
    ellipse(471,311,479,319).
draw_points(4):-
    ellipse(461,323,469,331),
    ellipse(481,323,489,331),
    ellipse(461,299,469,307),
    ellipse(481,299,489,307).
draw_points(3):-
    ellipse(461,323,469,331),
    ellipse(481,299,489,307),
    ellipse(471,311,479,319).
draw_points(2):-
    ellipse(461,323,469,331),
    ellipse(481,299,489,307).
draw_points(1):-
    ellipse(471,311,479,319).

member([Element, _, _, _],
Element, 0).
member([_, Element, _, _],
Element, 1).
member([_, _, Element, _],
Element, 2).
member([_, _, _, Element],
Element, 3).

replace([Element, A, B, C],
[_, A, B, C], Element, 0).
replace([A, Element, B, C],
[A, _, B, C], Element, 1).
replace([A, B, Element, C],
[A, B, _, C], Element, 2).
replace([A, B, C, Element],
[A, B, C, _], Element, 3).

menu_file(init):-
    menu( normal, _, _,
menu_baru(_), "&Permainan
Baru"),
    menu( normal, _, _,
menu_bermain(_), "&Cara
Bermain"),
    menu( normal, _, _,
menu_pembuat(_),
"&Pembuat"),
    menu( normal, _, _,
menu_keluar(_), "&Keluar").

menu_baru(press):-
set(pos([0,0,0,0],[0,0,0,0],
2)),
update_window(_).

```

## Daftar Pustaka

1. [www.dobrev.com](http://www.dobrev.com)
2. <http://symphonyofika.blogspot.co.id/2014/02/game-ludo.html>