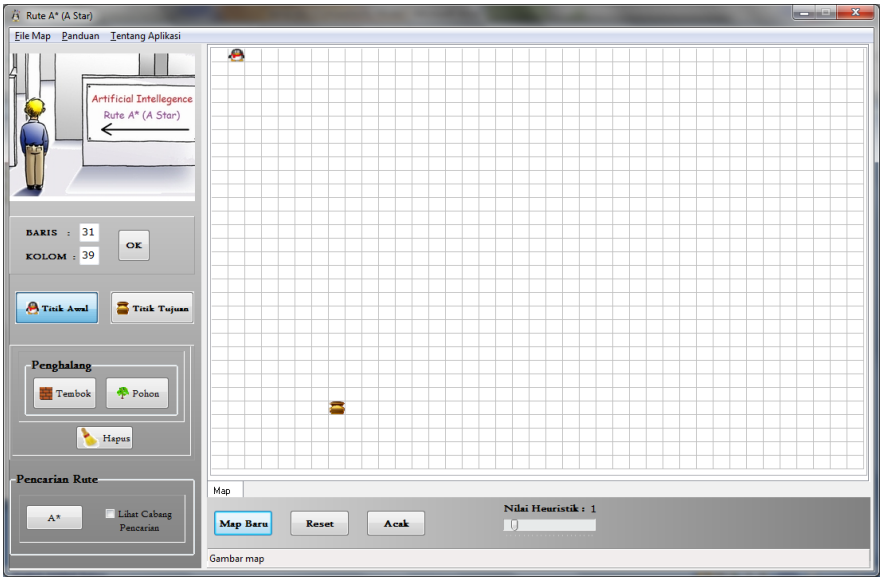
LAMPIRAN A

TAMPILAN ANTARMUKA

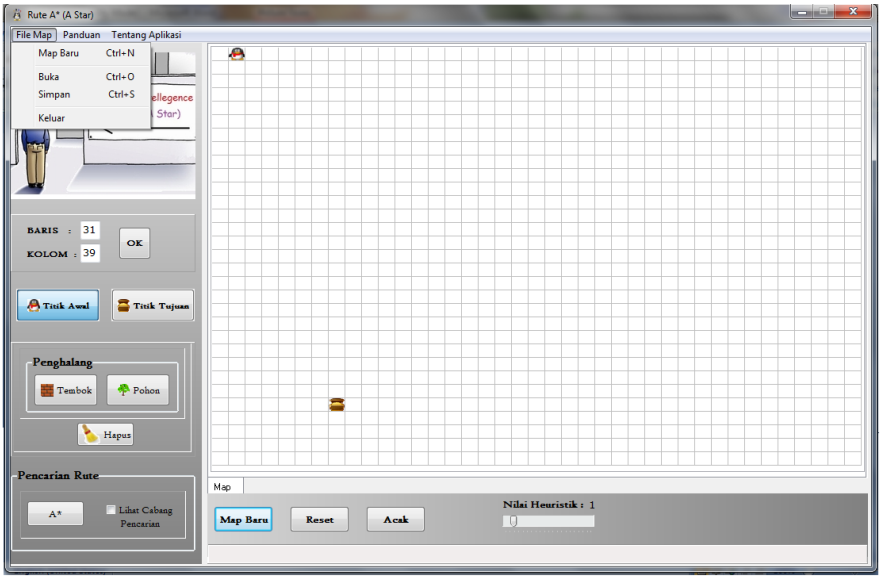
LAMPIRAN A

TAMPILAN ANTARMUKA

1. Tampilan Utama Aplikasi *Game Pathfinding*



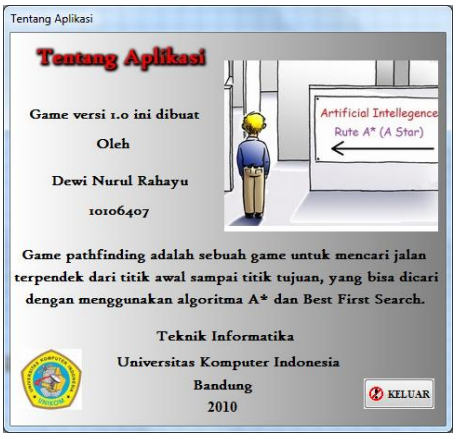
2. Tampilan Submenu *File Map*



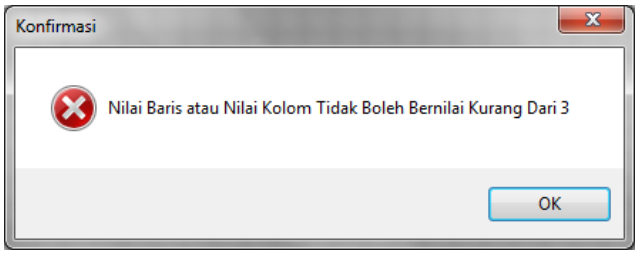
3. Tampilan Panduan Aplikasi



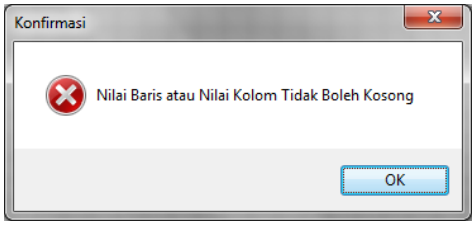
4. Tampilan Tentang Aplikasi

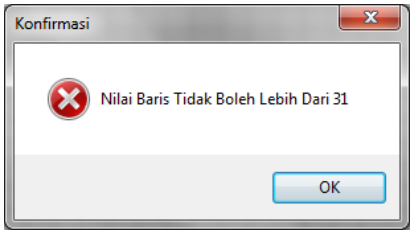


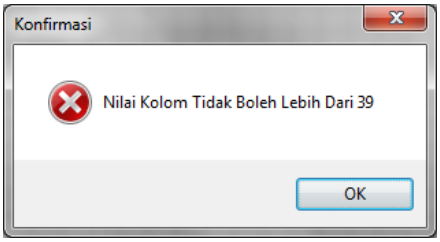
5. Tampilan Pesan

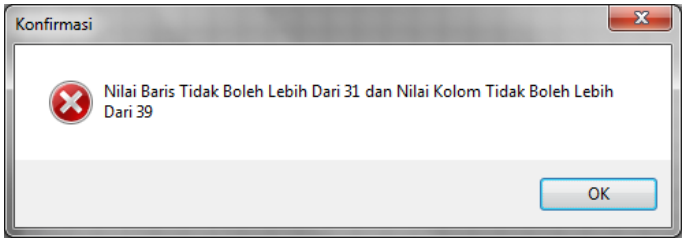


Tampilan Pesan (lanjutan)









LAMPIRAN B

LISTING PROGRAM

LAMPIRAN B

LISTING PROGRAM

```
unit U_Utama;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ActnList, StdCtrls, ExtCtrls, Grids, ComCtrls, PathPlanner, ImgList,
  Buttons, Menus, AppEvnts, XPMan, jpeg;

type
  TTerrainTypes = (ttWall, ttPohon, ttVisited{for TPathPlanner});
  TTerrainCosts = Array[TTerrainTypes] of Integer;
  TTerrainStrings = Array[TTerrainTypes] of String;
  TTerrainColors = Array[TTerrainTypes] of TColor;

const
  TerrainCost: TTerrainCosts = (255, 255, -1);
  Terrains: TTerrainStrings = ('Wall', 'Pohon', 'Visited');
  TerrainColors: TTerrainColors = (clTeal, clTeal, clMaroon);

type
  TMainForm = class(TForm)
    tsSimple: TTabSheet;
    sgMap: TStringGrid;
    StatusBar1: TStatusBar;
    actStepPath: TAction;
    actFindAStarPath: TAction;
    MainMenu1: TMainMenu;
    OpenDialog1: TOpenDialog;
    SaveDialog1: TSaveDialog;
    actSave: TAction;
    actSaveAs: TAction;
    actOpen: TAction;
    ApplicationEvents1: TApplicationEvents;
    SimplePathPlanner1: TSimplePathPlanner;
    AStarPathPlanner1: TAStarPathPlanner;
    SearchableMap1: TSearchableMap;

    procedure ApplicationEvents1Hint(Sender: TObject);
    procedure SimplePathPlanner1MapPassability(Sender: TObject; X, Y: Integer; var result:
Single);
    procedure SimplePathPlanner1BlockLocation(Sender: TObject; X,
      Y: Integer);
    procedure SimplePathPlanner1PrepareMap(Sender: TObject);
    procedure SimplePathPlanner1ValidLocationCheck(Sender: TObject; X,
      Y: Integer; var result: Boolean);
```

```
implementation

{$R *.DFM}
uses
   MMSystem, U_Bantuan, U_TentangAplikasi;

function TAStarPathPlanner.Search(Start: TPoint;
 var ScoreLimit: TPathCost): Boolean;
var
  Node: TCustomSearchState;
  ChildNode: TCustomSearchState;
  GoalState: TCustomSearchState;
  StartState: TCustomSearchState;
  Stack: TObjectStack;
  x, y: Integer;
  i: Integer;
  NewLimit: TPathCost;
  Done: Boolean;
begin
    result := False;
    if StateFactory = nil then
      raise EPathException.Create( SMissingStateFactory );
    Stack := TObjectStack.Create;
    try
      GoalState := StateFactory.CreateState( Self, FGoal.X, FGoal.Y );
      if GoalState = nil then
        raise EPathException.Create( SUnableCreateState );
      StartState := StateFactory.CreateState( Self, Start.X, Start.Y );
      StartState.GoalCostEstimate := StateFactory.EstimateGoalCost( StartState, GoalState );
      StartState.Cost := 0;
      if StartState = nil then
        raise EPathException.Create( SUnableCreateState );
      Stack.Push( StartState );
      NewLimit := MaxInt - 1;
      Done := False;
      while (Stack.Count > 0) and (Not Done) do
      begin
          Node := TCustomSearchState(Stack.Pop);
          if Node.Equals( GoalState ) then
          begin
              while Node <> nil do
              begin
                  StateFactory.GetXY( Node, X, Y );
                  Path.Add( X, Y );
                  Node := Node.Parent;
              end;
              Path.Reverse;
              result := True;
              Exit;
          end;
          if (Node.Parent <> nil) and
            (Node.Cost > (Node.Parent.Cost + Node.TerrainCost)) then
```

```
          Node.Cost := Node.Parent.Cost + Node.TerrainCost;      // f(n)=g(n)+h(n)
        for i := 0 to Node.NeighborCount - 1 do
        begin
          ChildNode := Node.Neighbor[i];
          if ChildNode = nil then continue
          else if not (sfVisited in ChildNode.SearchFlags) then
          begin
            ChildNode.GoalCostEstimate := StateFactory.EstimateGoalCost( ChildNode,
GoalState );
            ChildNode.Cost := StateFactory.CalculateCost( Node, ChildNode );
            ChildNode.SearchFlags := ChildNode.SearchFlags + [sfVisited];
            if ChildNode.Score <= ScoreLimit then
            begin
              ChildNode.Parent := Node;
              Stack.Push( ChildNode );
            end
            else if ChildNode.Score < NewLimit then
              NewLimit := ChildNode.Score;
          end
          else if ChildNode.Cost > StateFactory.CalculateCost( Node, ChildNode ) then
          begin
            ChildNode.Cost := StateFactory.CalculateCost( Node, ChildNode );
            if ChildNode.Score <= ScoreLimit then
            begin
              ChildNode.Parent := Node;
              Stack.Push( ChildNode );
            end
            else if ChildNode.Score < NewLimit then
              NewLimit := ChildNode.Score;
          end
          else if (ChildNode.Score > ScoreLimit) and (ChildNode.Score < NewLimit) then
            NewLimit := ChildNode.Score;
          if (Node.Score > ScoreLimit) and (Node.Score < NewLimit) then
            NewLimit := Node.Score;
        end;
        Node.SearchFlags := Node.SearchFlags + [sfVisited];
      end;
      ScoreLimit := NewLimit;
    finally
      Stack.Free;
    end;
end;

function TAStarPathPlanner.Search: Integer;
var
  Open: TStrings;
  GoalState: TCustomSearchState;
  Node: TCustomSearchState;
  ChildNode: TCustomSearchState;
  i: Integer;
  Index: Integer;
  NewCost: TPathCost;
```

```
  OldScore: TPathCost;
  X, Y: Integer;
begin
    if StateFactory = nil then
      raise EPathException.Create( SMissingStateFactory );
    Open := TStringList.Create;
    TStringList(Open).Sorted := True;
    try
      Node := StateFactory.CreateState( Self, Start.X, Start.Y );
      GoalState := StateFactory.CreateState( Self, Goal.X, Goal.Y );
      Node.GoalCostEstimate := StateFactory.EstimateGoalCost( Node, GoalState );
      Node.Cost := 0;
      Node.Parent := nil;
      Open.AddObject( Format( '%s%p', [FormatFloat('000000.00',Node.Score),
                                       Pointer(Node)] ), Node );
      result := -1;
      while Open.Count > 0 do
      begin
         Node := TCustomSearchState(Open.Objects[0]);
         Open.Delete( 0 );
         Node.SearchFlags := Node.SearchFlags - [sfInOpenQueue];
         Inc(result);
         if Node = GoalState then
         begin
            while Node <> nil do
            begin
               StateFactory.GetXY( Node, X, Y );
               Path[Path.Add( X, Y )].Tag := round(Node.Cost);
               Node := Node.Parent;
            end;
            Path.Reverse;
            Exit;
         end;
         if Assigned(FOnSearchState) then FOnSearchState( Self, Node );
         for i := 0 to Node.NeighborCount - 1 do
         begin
            ChildNode := Node.Neighbor[i];
            if ChildNode = nil then Continue;
            NewCost := Node.Cost + StateFactory.CalculateCost( Node, ChildNode );
            if ((sfInOpenQueue in ChildNode.SearchFlags) or
               (sfInClosedQueue in ChildNode.SearchFlags)) and
              (ChildNode.Cost <= NewCost) then Continue;
            ChildNode.Parent := Node;
            OldScore := ChildNode.Score;
            ChildNode.Cost := NewCost;
            ChildNode.GoalCostEstimate := StateFactory.EstimateGoalCost( ChildNode, GoalState );
            if sfInClosedQueue in ChildNode.SearchFlags then
            begin
               ChildNode.SearchFlags := ChildNode.SearchFlags - [sfInClosedQueue];
            end;
            if sfInOpenQueue in ChildNode.SearchFlags then
            begin
```

```
            Index := Open.Indexof(Format( '%s%p', [FormatFloat('000000.00',OldScore),
                                Pointer(ChildNode)] ) );
            if Index = -1 then
              raise Exception.Create('Should not happen');
            Open.Delete(Index);
          end;
          Open.AddObject( Format( '%s%p', [FormatFloat('000000.00',ChildNode.Score),
                                Pointer(ChildNode)] ), ChildNode );
          ChildNode.SearchFlags := ChildNode.SearchFlags + [sfInOpenQueue];
        end;
        Node.SearchFlags := Node.SearchFlags + [sfInClosedQueue];
      end;
      result := -1;
    finally
      Open.Free;
    end;
end;

procedure TMainForm.actFindPathExecute(Sender: TObject);
var
  i: Integer;
  Atime: Double;
  col: Integer;
  row: Integer;

begin
    ResetMap;
    SimplePathPlanner1.SetStart( StartCell );
    SimplePathPlanner1.SetEnd( EndCell );
    ATime := timeGetTime;
    col:= sgMap.ColCount;
    row:= sgMap.RowCount;

    StatusBar1.SimpleText := 'Jalan ditemukan dengan waktu pencarian:
'+IntToStr((round(timeGetTime-ATime))*1000)+' ms dan simpul yang diperiksa:
'+IntToStr(SimplePathPlanner1.FindPath);

    if SimplePathPlanner1.IsFound then
    begin
        for i := 1 to SimplePathPlanner1.Path.Count - 2 do
        begin
            sgMap.Cells[SimplePathPlanner1.Path[i].X, SimplePathPlanner1.Path[i].Y] := IntToStr(i);
        end;
    end
    else if  SimplePathPlanner1.Path.Count = (col*row)-2 then
        StatusBar1.SimpleText := 'Jalan ditemukan.';
end;

procedure TMainForm.actStepPathExecute(Sender: TObject);
var
  i: Integer;
begin
```

```
    if SimplePathPlanner1.Path.Count > 1 then
      with SimplePathPlanner1.Path[SimplePathPlanner1.Path.Count - 1] do
          sgMap.Cells[X, Y] := Terrains[ttVisited];
    StatusBar1.SimpleText := 'Simpul yang diperiksa = '+IntToStr(SimplePathPlanner1.FindPath(1));
    if SimplePathPlanner1.IsFound then
    begin
        for i := 1 to SimplePathPlanner1.Path.Count - 2 do
        begin
            sgMap.Cells[SimplePathPlanner1.Path[i].X, SimplePathPlanner1.Path[i].Y] := IntToStr(i);
        end;
    end
    else if SimplePathPlanner1.Path.Count > 1 then
        with SimplePathPlanner1.Path[SimplePathPlanner1.Path.Count - 1] do
        begin
            sgMap.Cells[X,Y] := '*';
        end
    else
        StatusBar1.SimpleText := 'Jalan tidak ditemukan.';
end;

procedure TMainForm.actFindAStarPathExecute(Sender: TObject);
var
  i,j: Integer;
  t1: Double;
  r: Integer;
  TotalTime: Double;
  AnEvent: TStateEvent;
begin
 AnEvent := AStarPathPlanner1.OnSearchState;
 SearchableMap1.Weight := tbHeuristic.Position/10;
 Screen.Cursor := crHourGlass;
 try
  // Fill the map
  ResetMap;
  Application.ProcessMessages;
  if PageControl1.ActivePage = tsSimple then
  begin
     SearchableMap1.Clear;
     for i := 0 to sgMap.ColCount - 1 do
       for j := 0 to sgMap.RowCount - 1 do
       begin
           SearchableMap1[i,j].Terrain := TerrainCost[TTerrainTypes(sgMap.Objects[i,j])];
       end;
     TotalTime := 0;
     for r := 1 to 1 {tbRepetitions.Position} do
     begin
       t1 := timeGetTime;
       SearchableMap1.Reset;
       j := AStarPathPlanner1.FindPath( StartCell, EndCell );
       TotalTime := TotalTime + (timeGetTime - t1);
       if j > -1 then
       begin
```

```
            if r = 1{tbRepetitions.Position} then
            begin
              for i := 1 to AStarPathPlanner1.Path.Count - 2 do
              begin
                 sgMap.Cells[AStarPathPlanner1.Path[i].X, AStarPathPlanner1.Path[i].Y] :=
IntToStr(i){(AStarPathPlanner1.Path[i].Tag)};
              end;
              StatusBar1.SimpleText := 'Jalan ditemukan dengan'{pengulangan: '+IntToStr(r)}+'
waktu pencarian: '+FloatToStr((TotalTime / r))+
                                  ' ms dan simpul yang diperiksa: '+IntToStr(j+1);
            end;
          end
          else
          begin
            StatusBar1.SimpleText := 'Jalan tidak ditemukan (waktu pencarian):
'+FloatToStr((TotalTime / r))+' ms';
            Exit;
          end;
        end;
    end;

  finally
    Screen.Cursor := crDefault;
    AStarPathPlanner1.OnSearchState := AnEvent;
  end;
end;
end.
```

LAMPIRAN C

HASIL KUESIONER

No. Responden : 01

1. Apakah penelusuran dengan algoritma A* mudah dipahami ?

a. Sangat Setuju	d. Tidak Setuju
X. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

2. Apakah jalur yang dihasilkan merupakan jalan terpendek ?

a. Sangat Setuju	d. Tidak Setuju
X. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

3. Apakah aplikasi ini sudah sesuai dengan tujuan yang ingin dicapai ?

a. Sangat Setuju	d. Tidak Setuju
X. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

4. Apakah diperlukan fungsi manual untuk pengguna mencari jalan sendiri ?

a. Sangat Setuju	d. Tidak Setuju
X. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

5. Apakah aplikasi *game pathfinding* mudah digunakan ?

a. Sangat Setuju	d. Tidak Setuju
X. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

Saran :

Agar lebih menarik bisa ditambahkan warna untuk pertama kali game dijalankan

Bandung, 23 Desember 2010

Pengisi Kuesioner

Rudi Herdiman

No. Responden : 02

1. Apakah penelusuran dengan algoritma A* mudah dipahami ?

a. Sangat Setuju	d. Tidak Setuju
X Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

2. Apakah jalur yang dihasilkan merupakan jalan terpendek ?

X Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

3. Apakah aplikasi ini sudah sesuai dengan tujuan yang ingin dicapai ?

X Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

4. Apakah diperlukan fungsi manual untuk pengguna mencari jalan sendiri ?

a. Sangat Setuju	d. Tidak Setuju
X Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

5. Apakah aplikasi *game pathfinding* mudah digunakan ?

a. Sangat Setuju	d. Tidak Setuju
X Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

Saran :

Gambar titik awal dan titik tujuan terlalu kecil, sehingga tampak tidak jelas. Tampilannya harus lebih menarik dari sekarang.

Bandung, 23 Desember 2010

Pengisi Kuesioner

Dimas Rachmat Pratama

No. Responden : 03

1. Apakah penelusuran dengan algoritma A* mudah dipahami ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

2. Apakah jalur yang dihasilkan merupakan jalan terpendek ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

3. Apakah aplikasi ini sudah sesuai dengan tujuan yang ingin dicapai ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

4. Apakah diperlukan fungsi manual untuk pengguna mencari jalan sendiri ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

5. Apakah aplikasi *game pathfinding* mudah digunakan ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

Saran :

tambahkan lebih menarik lagi path/jalan tujuan lebih diperjelas seperti arah jalan dari pencari sampai ke pada yang di cari penambahan animasi agar supaya lebih menarik

Bandung, 25 Desember 2010

Pengisi Kuesioner

Radi Karismatoga

No. Responden : 09

1. Apakah penelusuran dengan algoritma A* mudah dipahami ?

[x] a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

2. Apakah jalur yang dihasilkan merupakan jalan terpendek ?

[x] a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

3. Apakah aplikasi ini sudah sesuai dengan tujuan yang ingin dicapai ?

[x] a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

4. Apakah diperlukan fungsi manual untuk pengguna mencari jalan sendiri ?

[x] a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

5. Apakah aplikasi *game pathfinding* mudah digunakan ?

[x] a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

Saran :

Tampilan dibuat lebih menarik, untuk gambar supaya lebih diperjelas

Bandung, 25 Desember 2010

Pengisi Kuesioner

.....

No. Responden : 05

1. Apakah penelusuran dengan algoritma A* mudah dipahami ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

2. Apakah jalur yang dihasilkan merupakan jalan terpendek ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

3. Apakah aplikasi ini sudah sesuai dengan tujuan yang ingin dicapai ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

4. Apakah diperlukan fungsi manual untuk pengguna mencari jalan sendiri ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

5. Apakah aplikasi *game pathfinding* mudah digunakan ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

Saran :

petaknya diperbesar agar pinguin terlihat jelas.
Pinguin menggunakan Flash agar terlihat seperti berjalan.

Bandung, 25 Desember 2010

Pengisi Kuesioner

.....

No. Responden : 06

1. Apakah penelusuran dengan algoritma A* mudah dipahami ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

2. Apakah jalur yang dihasilkan merupakan jalan terpendek ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

3. Apakah aplikasi ini sudah sesuai dengan tujuan yang ingin dicapai ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

4. Apakah diperlukan fungsi manual untuk pengguna mencari jalan sendiri ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

5. Apakah aplikasi *game pathfinding* mudah digunakan ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

Saran :

Tampilan dibuat lebih menarik

Bandung, 25 Desember 2010

Pengisi Kuesioner

Ardi Hugraha T

No. Responden : 07

1. Apakah penelusuran dengan algoritma A* mudah dipahami ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

2. Apakah jalur yang dihasilkan merupakan jalan terpendek ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

3. Apakah aplikasi ini sudah sesuai dengan tujuan yang ingin dicapai ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

4. Apakah diperlukan fungsi manual untuk pengguna mencari jalan sendiri ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

5. Apakah aplikasi *game pathfinding* mudah digunakan ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

Saran :

- Tampilan lebih menarik dan algoritmanya ditambah.

Bandung, 25 Desember 2010

Pengisi Kuesioner

.....

No. Responden : 08

1. Apakah penelusuran dengan algoritma A* mudah dipahami ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

2. Apakah jalur yang dihasilkan merupakan jalan terpendek ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

3. Apakah aplikasi ini sudah sesuai dengan tujuan yang ingin dicapai ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

4. Apakah diperlukan fungsi manual untuk pengguna mencari jalan sendiri ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

5. Apakah aplikasi *game pathfinding* mudah digunakan ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

Saran :

Map yang digunakan lebih besar dan warna tak harus mencolok dari penghalang

Bandung, 25 Desember 2010

Pengisi Kuesioner

.....

No. Responden : 09

1. Apakah penelusuran dengan algoritma A* mudah dipahami ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

2. Apakah jalur yang dihasilkan merupakan jalan terpendek ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

3. Apakah aplikasi ini sudah sesuai dengan tujuan yang ingin dicapai ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

4. Apakah diperlukan fungsi manual untuk pengguna mencari jalan sendiri ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

5. Apakah aplikasi *game pathfinding* mudah digunakan ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

Saran :

Aplikasinya udah ok, baik tampilan ataupun fungsionalitasnya.

Bandung, 25 Desember 2010

Pengisi Kuesioner

.....

No. Responden : 10

1. Apakah penelusuran dengan algoritma A* mudah dipahami ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

2. Apakah jalur yang dihasilkan merupakan jalan terpendek ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

3. Apakah aplikasi ini sudah sesuai dengan tujuan yang ingin dicapai ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

4. Apakah diperlukan fungsi manual untuk pengguna mencari jalan sendiri ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

5. Apakah aplikasi *game pathfinding* mudah digunakan ?

a. Sangat Setuju	d. Tidak Setuju
b. Setuju	e. Sangat tidak Setuju
c. Ragu-ragu	

Saran :

Sudah bagus dan mudah dimengerti.

Bandung, 25 Desember 2010

Pengisi Kuesioner

.....