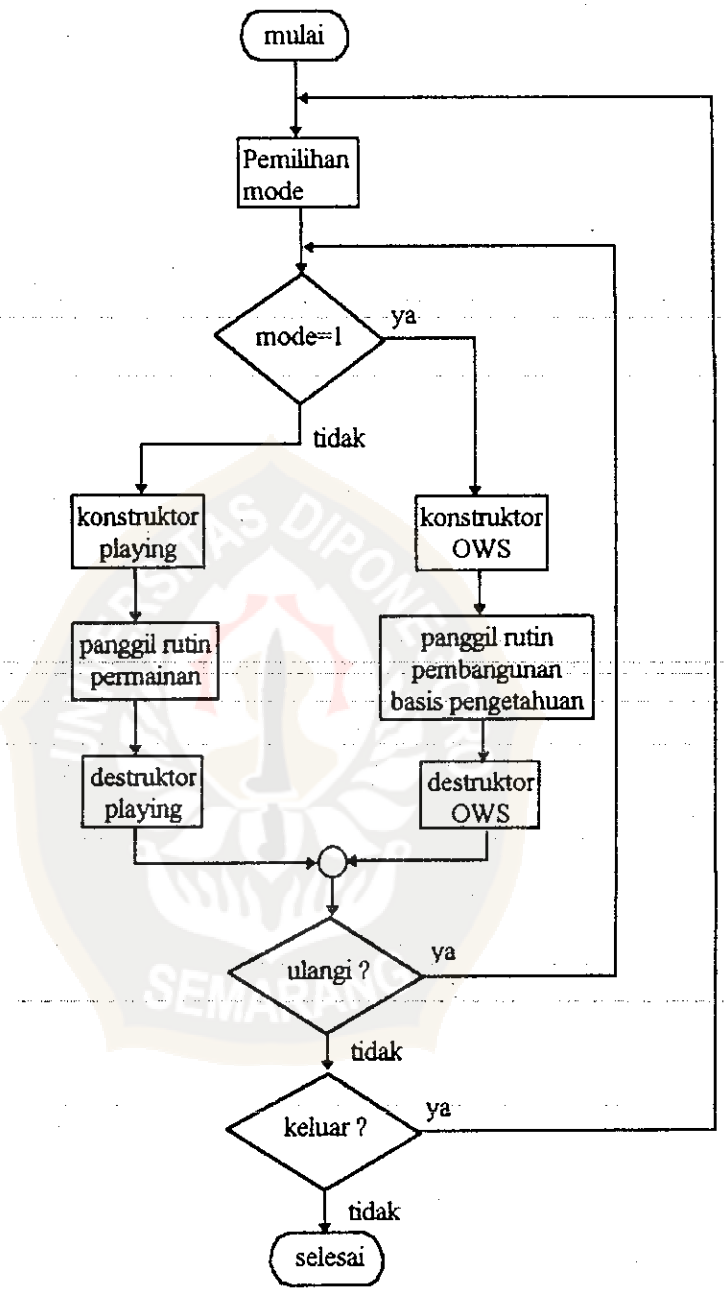
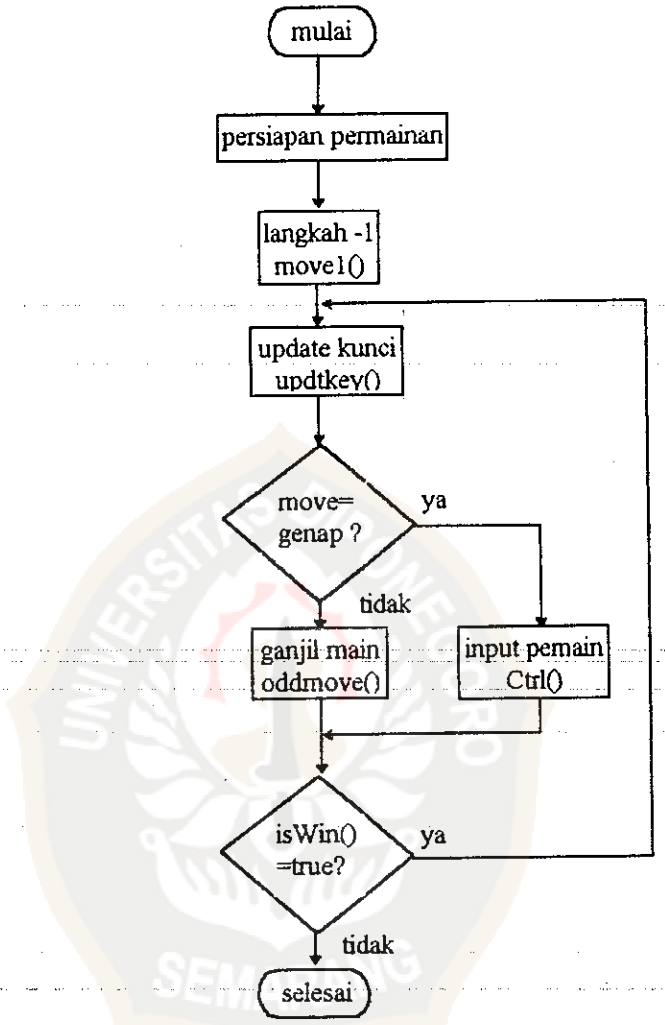
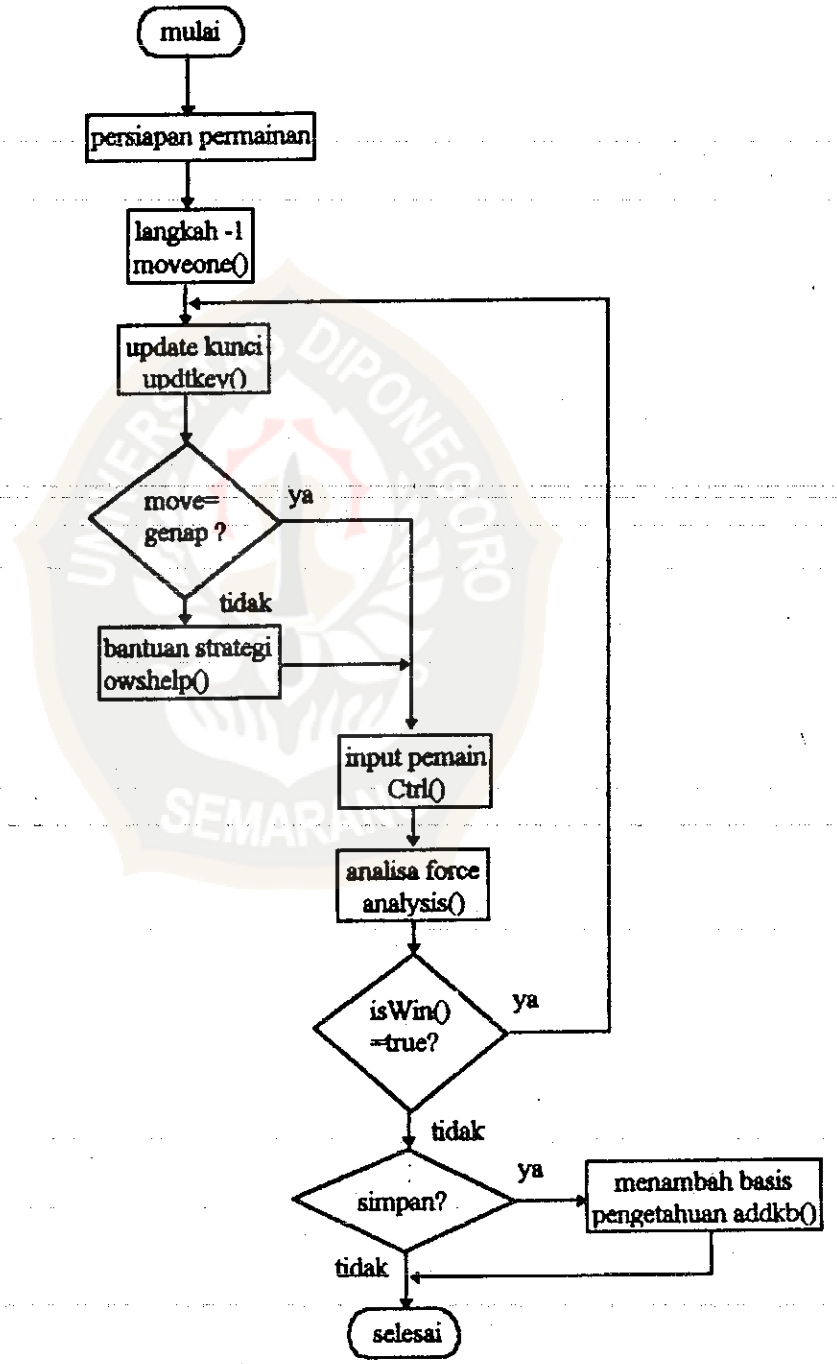
Skema program utama.



Flowchart rutin permainan.



Flowchart rutin pembangunan basis pengetahuan.



```
#include <conio.h> //clrscr()
#include <stdlib.h> //div()
#include <iostream.h>
#include <mpltn.h>
#include <knwldg.h>
#include <screen.h>
#include <hlp.h>

class cheking : public mpltn
{ public:
    bool chkfrc(char a, char b, char c);
    bool chkfrc2(char a, char b, char c);
    bool isWin();
    void chek(char a, char b, char c, char d, char e, char line);
    void InitC();
    void Chkfrc2();
    void showC();  //tester
  protected:
    bool frc2;
  private:
    unsigned char C[9];
    void WIN(char line, char nline);
    bool open1(char a, char b, char c, char *ord);
    bool Chkcell(char a, char b, char c);
 };

class process : public cheking, public knwldg, public hlp
{ public:
    bool force();
    void updtkey(char Move);
    void oddmove();
    void test();  //tester
  protected:
    long int fptr;
  };

class media : public screen
{ public:
    void Board(char Bx, char By);
 };

class playing : private process, private media
{ public:
    playing();
    ~playing();
    void play();
 };

class OWS : private process, private media
{ public:
    void buildkb();
  private:
    void init();
    void addkb();
    void owshelp();
    void analysis();

  };
```

```
void main()
{
  textattr(0x07);
  clrscr();
//  playing nt3;
//  nt3.play();
  OWS nttt;
  nttt.buildkb();
  getch();
 }

// Fungsi-fungsi yang di test :

void OWS::buildkb()
{
  init();
  do
  { if ((move%2)==1) owshelp();
    Ctrl();
    updtkey(move-2);
    analysis();
    } while (lisWin());
   addkb();
 }

void OWS::init()
{ move=1;
  knwldg::init();
  mpltn::init();
  frc2=false;
  Board(Bx,By);
  owshelp();
  moveone();
  updtkey(move-2);
  analysis();
 }

void OWS::addkb()
{ char ask;
  cout<<"Do you want save this patern game ? [y/n]";
  do { ask=getch(); } while ((ask!='y')&&(ask!='n'));
  if (ask=='n') return;
  if (chkkb(move)) cout<<" The patern is exist";
  else cout<<" Save the patern into knowledge base";
//  wrt();
 };

void showfrc(char x, char y, char line, char nline, bool frc)
{ gotoxy(x,y);
  switch(line)
  { case 1: cout<<"Horz. line ";break;
    case 2: cout<<"Vert. line ";break;
    case 3: cout<<"Diag. line ";break;
   }
  cout<<(int)nline<<" :";
  if (frc) { textattr(0x3E); cputs("is Force !"); }
  else { textattr(0x3C); cputs("available"); }
 }
```

```
void OWS::owshelp()
{ char hx=40,hy=4;
  switch(move)
  { case 1: hlp1(hx,hy); break;
    case 3: Chkfrc2();
            if (!frc2)
            { hlp3(hx,hy); }
            else
            { hlp3f(hx,hy); }
            break;
    case 5: if (!frc2)
            { hlp5(hx,hy); }
            else
            { hlp5f(hx,hy); }
  }
 }

void cheking::chek(char a, char b, char c, char d, char e, char line)
{ char val,ord;
  textattr(0x54);
  val=15-C[a]-C[b]-C[c];
  gotoxy(d,e);
  if ((line>=1)&&(line<=3)) cout<<"H_line "<<(int)line;
  else if ((line>=4)&&(line<=6)) cout<<"V_line "<<(int)line-3;
  else if ((line>=7)&&(line<=8)) cout<<"D_line "<<(int)line-6;
  if ((open1(a,b,c,&ord)) && !(Used(val,ord)) && (val<10) && (val>0))
  { textattr(0x5E); cputs(" FORCE to "); cout<<(int)val;
   }else cout<<" not force ";
 }


void OWS::analysis()
{ char x=32,y=19;
  InitC();
  chek(8,1,2,x,y,1); chek(1,0,5,x+24,y++,5);
  chek(7,0,3,x,y,2); chek(2,3,4,x+24,y++,6);
  chek(6,5,4,x,y,3); chek(8,0,4,x+24,y++,7);
  chek(8,7,6,x,y,4); chek(2,0,6,x+24,y++,8);
 }


/*************************************************************************
// kode fungsi-fungsi anggota kelas cheking:
bool cheking::open1(char a, char b, char c, char *ord)
{ char code=0;
  if (C[a]==0) { code++; *ord=a;}
  if (C[b]==0) { code++; *ord=b;}
  if (C[c]==0) { code++; *ord=c;}
  if (code==1) return true;
  else return false;
 }

bool cheking::chkfrc(char a, char b, char c)
{ char val,ord;
  val=15-C[a]-C[b]-C[c];
  if ((open1(a,b,c,&ord)) && ((val%2)==1)
       && !(Used(val,ord)) && (val<10) && (val>0))
  { play(ord,val); return true; }
  return false;
 }
```

```
bool cheking::chkfrc2(char a, char b, char c)
{ char val,ord;
  val=15-C[a]-C[b]-C[c];
  if ((open1(a,b,c,&ord)) && !(Used(val,ord)) && (val<10) && (val>0))
  { if ((val%2)==0) frc2=true; return true; }
  return false;
 }

void cheking::Chkfrc2()
{ InitC();
  if (chkfrc2(8,1,2)) return; // | 8 | 1 | 2 |
  if (chkfrc2(7,0,3)) return; // | 7 | 0 | 3 |
  if (chkfrc2(6,5,4)) return; // | 6 | 5 | 4 |
  if (chkfrc2(8,7,6)) return;
  if (chkfrc2(1,0,5)) return;
  if (chkfrc2(2,3,4)) return;
  if (chkfrc2(8,0,4)) return;
  if (chkfrc2(2,0,6)) return;
 }

void cheking::InitC()
{ char i;
  for (i=0;i<=8;i++) C[i]=0;
  for (i=0;i<=move-1;i++) C[Ord[i]]=Cell[i];
 }

void cheking::WIN(char line, char nline)
{ gotoxy(12,22);
  cout << " Winning reached ! ";
  switch(line)
   { case 0: cout<<"Horisontal";break;
     case 1: cout<<"Vertical";break;
     case 2: cout<<"Diagonal"; }
   cout<<" line "<<(int)nline<<" by ";
   if (((move-1)%2)==1) cout<<"Odd player"<<endl;
    else cout<<"Even player"<<endl;
 }

bool cheking::isWin()
{ InitC();
  if (Chkcell(8,1,2)) { WIN(0,1); return true; }
  if (Chkcell(7,0,3)) { WIN(0,2); return true; }
  if (Chkcell(6,5,4)) { WIN(0,3); return true; }
  if (Chkcell(8,7,6)) { WIN(1,1); return true; }
  if (Chkcell(1,0,5)) { WIN(1,2); return true; }
  if (Chkcell(2,3,4)) { WIN(1,3); return true; }
  if (Chkcell(8,0,4)) { WIN(2,1); return true; }
  if (Chkcell(2,0,6)) { WIN(2,2); return true; }
  if (move>9) { gotoxy(15,22); cout <<"The game is draw !"<<endl; return tr
  return false;
 }

bool cheking::Chkcell(char a, char b, char c)
{ if (((C[a]+C[b]+C[c])==15) &&
      (C[a]!=0) && (C[b]!=0) && (C[c]!=0))
       return true;
   return false;
 }
```

```
/*****************************************************************/
// Kode fungsi-fungsi anggota kelas process:
void process::updtkey(char Move)
{ char cell,ord;
  cell = convert(Cell[Move])*10;
  ord  = rotate(Ord[Move]);
  keys[Move]=cell+ord;
//  cout<<"keys["<<(int)Move<<"]="<<(int)keys[Move]<<endl;
 }

bool process::force()
{ InitC();
  if (chkfrc(8,1,2)) return true; // | 8 | 1 | 2 |
  if (chkfrc(7,0,3)) return true; // | 7 | 0 | 3 |
  if (chkfrc(6,5,4)) return true; // | 6 | 5 | 4 |
  if (chkfrc(8,7,6)) return true;
  if (chkfrc(1,0,5)) return true;
  if (chkfrc(2,3,4)) return true;
  if (chkfrc(8,0,4)) return true;
  if (chkfrc(2,0,6)) return true;
  return false;
 }

void process::oddmove()
{ div_t buff;
  char i,ord,j;
  if (move==1)
  {
    move1();
    fptr=0;
    for (i=0;i<=8;i++) keys[i]=0;
    filesize();
    load(&fptr);
//    cout<<"fsize="<<(int)fsize<<" fptr="<<(int)fptr;
    return;
  }
//  for (i=0;i<=8;i++) cout<<" "<<(int)keys[i];
  do
  {
    for (i=0;i<=49;i++)
    {
//      for (char j=0; j<=8; j++)
//      {  cout<<(int)kbase[i].ows[j]<<" ";}
//      cout<<endl;
        if (memcmp(kbase[i].ows,keys,move-1)==0)
        {
          buff=div(kbase[i].ows[move-1],10);
          ord= backrot(buff.rem);
          buff.quot=getnum(buff.quot);
          play(ord,buff.quot);
          return;
        }
//        if (i==23) getch();
      }
      if (fptr<fsize)
      {
        load(&fptr);
        continue;
      } else break;
  } while(1);
```

```
    textcolor(3);
    gotoxy(2,25);
    cout<<"Warning! not enough knowledge base,
    I'll play with random"<<endl;
    fptr=0;
    rndplay();
    return;
}
/*
void process::test()
{ clrscr();
  Cell[0]=1; Ord[0]=1;
  Cell[1]=8; Ord[1]=2;
  Cell[2]=3; Ord[2]=8;
  Cell[3]=2; Ord[3]=7;
  move=4;
//  Cell[4]=5; Ord[4]=5;
//  Cell[5]=6; Ord[5]=0; move=6;
  if (force()) { cout << " Is force !\n"; }
  oddmove();
  showC();
  isWin();
 }

void cheking::showC()
{ for(char i=0;i<=move-1;i++) C[Ord[i]]=Cell[i];
  cout<<(int)C[8]<<" "<<(int)C[1]<<" "<<(int)C[2]<<endl;
  cout<<(int)C[7]<<" "<<(int)C[0]<<" "<<(int)C[3]<<endl;
  cout<<(int)C[6]<<" "<<(int)C[5]<<" "<<(int)C[4]<<endl;
  };
  */
/*************************************************************/
// Kode fungsi-fungsi anggota kelas playing.

void playing::playing(){ mpltn::move=1; }
void playing::~playing() { }
void playing::play()
{
  Bx=30,By=8; //Save location board.
  Board(30,8);
  do
  { (move%2) ? oddmove() : Ctrl();
     updtkey(move-1);
//  cout<<" keys ["<<(int)(move-2)<<"]="<<(int)keys[move-2];
    } while (!isWin());
 }


/*************************************************************/
// Kode fungsi-fungsi anggota kelas media.

void media::Board(char x, char y)
{
  box(x-2,y-1,x+21,y+8,0x3F);
  gotoxy(x-2,y-1);textattr(0x1F);
  cputs("    Tic-Tac-Toe Board   ");
  textattr(0x31);
  gotoxy(x,y++); cputs("   1     2     3     ");
  gotoxy(x,y++); cputs("╔═══╦═══╦═══╗");
  gotoxy(x,y++); cputs("╠═══╬═══╬═══╣ 1");
```

```
    gotoxy(x,y++); cputs("┌──────┬──────┬──────┐  ");
    gotoxy(x,y++); cputs("│      │      │      │ 2");
    gotoxy(x,y++); cputs("├──────┼──────┼──────┤  ");
    gotoxy(x,y++); cputs("│      │      │      │ 3");
    gotoxy(x,y);   cputs("└──────┴──────┴──────┘  ");
    textattr(0x17);
}
```

```
//SCREEN.H

class screen{
  struct map{ char font[16]; };
  char *scrn[5];
 public:
    void newascii();
    void Ctxt(char y, char str[80]);
    void fillscr(char x1, char y1, char x2, char y2, char chr);
    void cls(char x1, char y1, char x2, char y2);
    void box(char x1, char y1, char x2, char y2, char attr);
    void savescr(char x1, char y1, char x2, char y2, char buf);
    void restscr(char x1, char y1, char x2, char y2, char buf);
 private:
    void remap(char ascii,char font[16]);
    void vga();
    int offset(char x, char y);
    void shadow(char x1, char y1, char x2, char y2);
};


//KNWLDG.H

class knwldg   //kelas pengetahuan
{
  public:
    void init();
    void filesize();
    void wrt();
    void load( long int *fptr);
    char chkkb( char move);

  protected:
    struct OWS{ char ows[9]; } kbase [50];
    char keys[9];
    long int fsize,fptr;

  private:

 };



//HLP.H
#include <iostream.h>
#include <conio.h> //gotoxy(),cputs(),textattr()

class hlp
{ public:
    void hlpms(char x, char y);
    void hlp1(char x, char y);
    void hlp3(char x, char y);
    void hlp3f(char x, char y);
    void hlp5(char x, char y);
    void hlp5f(char x, char y);
 };
```

```
enum bool {false,true};

class mpltn     // kelas manipulasi
{ char table [9],deg;

  public:
   char Bx,By;
   mpltn() { Bx=5; By=4; init();};
   char convert(char val);
   char getnum(char val);
   char rotate(char pos);
   char backrot(char pos);
   char getdeg();
   void gettbl();
   void rndplay();
   void play(char ord, char val);
   bool Used(char key, char ord);
   void Ctrl();
   void init();
   void move1();
   void moveone();

  protected:
   char Cell[9],Ord[9],move;

  private:
   char degree(char pos);
   void gotoord(char ord);
 };
```

```
// SCREEN.CPP

#include <conio.h>     //gotoxy(),clrscr(),window(),textattr(),puttext()
#include <dos.h>       //pokeb
#include <iostream.h>
#include <string.h>    //strlen()
#include <stdlib.h>    //div()
#include "screen.h"

/*************************************************************************
// Kode fungsi-fungsi anggota dari kelas knowledge
// Public:

void screen::newascii()
{
  map fnt[9]={ 0xFF,0xFF,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,
               0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,
               0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,
               0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
               0xFF,0xFF,0x03,0x03,0x03,0x03,0x03,0x03,
               0x03,0x03,0x03,0x03,0x03,0x03,0x03,0x03,
               0x03,0x03,0x03,0x03,0x03,0x03,0x03,0x03,
               0x03,0x03,0x03,0x03,0x03,0x03,0x03,0x03,
               0x03,0x03,0x03,0x03,0x03,0x03,0x03,0x03,
               0x03,0x03,0x03,0x03,0x03,0x03,0xFF,0xFF,
               0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
               0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,
               0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,
               0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xFF,0xFF,
               0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,
               0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,
               0x81,0x42,0x24,0x18,0x18,0x24,0x42,0x81,
               0x81,0x42,0x24,0x18,0x18,0x24,0x42,0x81 };
   vga();
   remap(218,fnt[0].font);
   remap(196,fnt[1].font);
   remap(183,fnt[2].font);
   remap(221,fnt[3].font);
   remap(217,fnt[4].font);
   remap(220,fnt[5].font);
   remap(212,fnt[6].font);
   remap(179,fnt[7].font);
   remap(177,fnt[8].font);
  }

 void screen::fillscr(char x1, char y1, char x2, char y2, char chr)
 { char i,j;
   for (i=y1;i<=y2;i++)
   for (j=x1;j<=x2;j++)
   { pokeb(0xB800,(i-1)*160+(j-1)*2,chr);
     pokeb(0xB800,(i-1)*160+(j-1)*2+1,0x17);
    }
  }

 void screen::Ctxt(char y, char str[80])
 { div_t buff;
   char x;
   buff=div((82-strlen(str)),2);
   x=buff.quot;
   gotoxy(x,y); cout << str;
  }
```

```
void screen::cls(char x1, char y1, char x2, char y2)
{ window(x1,y1,x2,y2);
  clrscr();
  window(1,1,80,25);
 }

void screen::box(char x1, char y1, char x2, char y2, char attr)
{ char i;
  window(x1,y1,x2,y2);
  textattr(attr);
  clrscr();
  window(1,1,80,25);
  for (i=x1+1;i<=x2-1;i++)
  { gotoxy(i,y1); cout << "─"; //196
    gotoxy(i,y2); cout << "▄"; //220
   }
  for (i=y1+1;i<=y2-1;i++)
  { gotoxy(x1,i); cout << "│"; //179
    gotoxy(x2,i); cout << "▌"; //221
   }
   gotoxy(x1,y1); cout << "┌"; //218
   gotoxy(x1,y2); cout << "└"; //212
   gotoxy(x2,y1); cout << "┐"; //183
   gotoxy(x2,y2); cout << "┘"; //217
   shadow(x1+1,y1,x2,y2);
 }

void screen::savescr(char x1, char y1, char x2, char y2, char buf)
{ if (buf>5) { cout << "I only have 5 buffer, ok"; exit(1); }
  if (scrn[buf]!=0)
  { cout << "Sorry that buffer is not empty!"; exit(1); }
  scrn[buf]= new char[(x2-x1+1)*(y2-y1+1)*2];
  gettext(x1,y1,x2,y2,scrn[buf]);
 }

void screen::restscr(char x1, char y1, char x2, char y2, char buf)
{ if (scrn[buf]==0) { cout << "Buffer is empty!"; exit(1); }
  puttext(x1,y1,x2,y2,scrn[buf]);
  delete scrn[buf];
 }

//private:
void screen::remap(char ascii, char font[16])
{ struct REGPACK reg;
  reg.r_ax=0x1100;
  reg.r_bx=0x1600;
  reg.r_cx=0x0001;
  reg.r_dx=ascii;
  reg.r_es=FP_SEG(font);
  reg.r_bp=FP_OFF(font);
  intr(0x10,&reg);
 }

void screen::vga()
{ _AX=0x1A00;
  asm int 10h;
  if (_AL!=0x01A)
  { cout <<"This isn't vga\svga card monitor, terminate program";
    exit(1); }
 }

int screen::offset(char x, char y)
{ return((x-1)*2+(y-1)*160);
 }
```

```
void screen::shadow(char x1, char y1, char x2, char y2)
{ char i;
  for (i=x1;i<=x2;i++) pokeb(0xB800,offset(i+1,y2+1)+1,0x08);
  for (i=y1;i<=y2;i++) pokeb(0xB800,(offset(x2+1,i+1))+1,0x08);

  for (i=x1+1;i<=x2+1;i++) pokeb(0xB800,(offset(i+1,y2+1))+1,0x08);
  for (i=y1;i<=y2;i++) pokeb(0xB800,(offset(x2+2,i+1))+1,0x08);
 }
```

```
// KNWLDG.CPP

#include <conio.h>  //gotoxy(),clrscr(),window(),textattr(),puttext()
#include <fstream.h>
#include <iostream.h>
#include <stdlib.h>  //randomize(),rand()
#include "knwldg.h"

//*************************************************************************
// Kode fungsi-fungsi anggota dari kelas knowledge
// Public:

void knwldg::init()
{ for (char i=0;i<=8;i++) keys[i]=0;
  fptr=0;
  load(&fptr);
  filesize();
  }

void knwldg::filesize()
{ fstream fkbase("OWS.DAT", ios::in | ios::binary);
  if (!fkbase) { cout << "Error! can't open OWS.DAT"; exit(1); }
  fkbase.seekg(0,ios::end);
  fsize=fkbase.tellg();
  fkbase.close();
  }

void knwldg::wrt()
 { fstream fkbase("OWS.DAT", ios::app | ios::binary);
   if (!fkbase) { cout << "Error! can't open OWS.DAT"; exit(1); }
   fkbase.write(keys,sizeof(keys));
   fsize=fkbase.tellp();
   fkbase.close();
  }

void knwldg::load(long int *fptr)
{ fstream fkbase("OWS.DAT", ios::in | ios::binary);
  if (!fkbase) { cout << "Error! can't open OWS.DAT"; exit(1); }
  fkbase.seekg(*fptr,ios::beg);
  fkbase.read((char*)&kbase,sizeof(kbase));
  *fptr=fkbase.tellg();
  fkbase.close();
 }

char knwldg::chkkb(char move)
{
  fptr=0; load(&fptr);
  while (1)
  {
   for (char i=0;i<=49;i++)
   if (memcmp(kbase[i].ows,keys,move-2)==0) return 1;
   if (fptr<fsize)
   { load(&fptr);
     continue;
    }
    else break;
   }
   return 0;
}
```

```
/* Hlp.CPP */
#include <hlp.h>
void hlp::hlpms(char x, char y)
{ gotoxy(x-2,y++); cputs("      Magic Square      ");
  textattr(0xD0);
  gotoxy(x,y++); cputs("                        ");
  gotoxy(x,y++); cputs("  8   1   6             ");
  gotoxy(x,y++); cputs("                        ");
  gotoxy(x,y++); cputs("  3   5   7             ");
  gotoxy(x,y++); cputs("                        ");
  gotoxy(x,y++); cputs("  4   9   2             ");
  gotoxy(x,y++); cputs("                        ");
  textattr(0xB1);
  gotoxy(x-2,y); cputs("  for help in analysis ");
 }

void hlp::hlp1(char x, char y)
{ gotoxy(x,y);cout <<"Play 1,3,7, or 9 in any side cell";}

void hlp::hlp3(char x, char y)
{ gotoxy(x,y++);cout <<"Move two isn't force to even's win:";
  gotoxy(x,y++);cout <<"a> Force move four to the centre or to";
  gotoxy(x,y++);cout <<"   the corner.";
  gotoxy(x,y++);cout <<"b> Attack all open lines containing";
  gotoxy(x,y);  cout <<"   move two.";
 }

void hlp::hlp3f(char x, char y)
{ gotoxy(x,y++);cout <<"Move two is force to even's win:";
  gotoxy(x,y++);cout <<"l> Block with move two's remaining ood";
  gotoxy(x,y);  cout <<"   neighbor.";
 }

void hlp::hlp5(char x, char y)
{ gotoxy(x,y++);cout <<"a> Attack all open lines containing";
  gotoxy(x,y++);cout <<"   move four with move two's remaining";
  gotoxy(x,y);  cout <<"   ood neighbor.";
 }

void hlp::hlp5f(char x, char y)
{ gotoxy(x,y++);cout <<"a> Attack the open line containing";
  gotoxy(x,y++);cout <<"   move four with force.";
  gotoxy(x,y++);cout <<"b> force move six to play in both";
  gotoxy(x,y++);cout <<"   an open line containing move two and";
  gotoxy(x,y++);cout <<"   the open line containing move four.";
  gotoxy(x,y++);cout <<"c> Attack the open line containing";
  gotoxy(x,y++);cout <<"   move four by playing in the open ";
  gotoxy(x,y);  cout <<"   line containing move two.";
 }
```

```
// MPLTN.CPP

#include <conio.h>     //gotoxy(),clrscr(),window(),textattr(),puttext()
#include <bios.h>      //bioskey()
#include <iostream.h>
#include <stdlib.h>  //randomize(),rand()
#include "mpltn.h"

//*********************************************************
// Kode fungsi-fungsi anggota dari kelas manipulasi
// Public:

void mpltn::init()
{ for (char i=0;i<=8;i++)
  { Cell[i]=0;Ord[i]=9; }

 };

void mpltn::move1()
{ char val,ord;
  randomize();
  do val=(rand()%5)*2+1; while (val==5);
  ord=(rand()%4)*2+1;
  play(ord,val);
  gettbl();
  deg=getdeg();
 }

void mpltn::moveone()
{ while(1)
  { Ctrl();
    gotoxy(12,2);
    if ((Ord[0]%2)==0)
    { cout << " Don't play in corner or centre! ";
      gotoord(Ord[0]); cout << " ";
      move=1; Cell[0]=0; Ord[0]=9;
      continue;
     }
    if (Cell[0]==5)
    { cout << " Don't play 5!                    ";
      gotoord(Ord[0]); cout << " ";
      move=1; Cell[0]=0; Ord[0]=9;
      continue;
      }
    break;
    }
   gettbl();
   deg=getdeg();
 }


char mpltn::convert(char val)
{ char i,*p;
  p=&table[0];
  for (i=0;i<=8;i++,p++)
  if (*p==val) return i;
  if (val!=0) cout<<"Warning! can't convert value";
  return 0;
 }
```

```
char mpltn::getnum(char val)
{
  return (table[val]);
 }

char mpltn::rotate(char pos)
{ if (pos==0 || deg==0) return pos;
  if (pos<=deg) return(pos+8-deg);
  return (pos-deg);
 }

char mpltn::backrot(char pos)
{ if (pos==0 || deg==0) return pos;
  if (pos>deg)
  { pos+=deg;
    if (pos>8) pos-=8;}
  return pos;
 }

void mpltn::gettbl()
{ char tbl[9][4]={ 1,3,9,7,5,2,4,8,6,
                   3,1,7,9,5,2,6,8,4,
                   7,1,3,9,5,4,8,6,2,
                   9,3,1,7,5,6,8,4,2, };
  char *p,ptr;
  ptr=Cell[0];
  p=&tbl[0][0];
  for (char i=0;i<=3;i++,p+=9)
  if (*p==ptr) { memmove(table,p,9); return; }
  cout << "Error: Move-1 is 5 or nothing";
 }

void mpltn::rndplay()
{ char val,ord;
  randomize();
  do
   { val=(rand()%5)*2+1;
     ord=rand()%9;
     } while (Used(val,ord));
  play(ord,val);
 }

void mpltn::play(char ord, char val)
{ Ord[move-1]=ord;
  Cell[move-1]=val;
  gotoord(ord);
  cout << (int)val;
  move++;
 }

bool mpltn::Used(char key, char ord)
/* if that number in use, function return true */
{ char *p,*q;
  for(p=&Cell[0],q=&Ord[0];p<=&Cell[0]+8;p++,q++)
   if ((*p==key) || (*q==ord))
    return true;
   }
  return false;
 }
```

```
void mpltn::Ctrl()
{ register char i,j;
  char tblord[3][3]={ 8,1,2,
                      7,0,3,
                      6,5,4 };
  char Cx,Cy,*p,ord;
  int keyb;
  char key;
  Cx=Cy=0;
  gotoxy(Cx*6+Bx+3,Cy*2+By+2);
  do
  { do              /*-  User input inspection for key  -*/
    { keyb=bioskey(0);
      switch((char)(keyb>>8))
      { case 72: if (Cy!=0) Cy--;  // Up
                 break;
        case 75: if (Cx!=0) Cx--;  // Left
                 break;
        case 77: if (Cx!=2) Cx++;  // Right
                 break;
        case 80: if (Cy!=2) Cy++;  // Down
       }
      gotoxy(Cx*6+Bx+3,Cy*2+By+2);
     } while ((keyb & 0xFF)==0x00);
    if ((key=(char)keyb)==27) exit(1);
    key=((char)keyb)-48;
    ord=tblord[Cy][Cx];
    if ((Cell[move-1]==0) && !(Used(key,ord)) && (key>=1) && (key<=9))
    if ((move%2)==0)
      { if (((key%2)==0)) break; }
    else if (((key%2)==1)) break;
  } while(1);
// transfering value
  play(ord,key);
}


//private:
char mpltn::getdeg()
{ return(Ord[0]-1); }

void mpltn::gotoord(char ord)
{ switch (ord)
  { case 0: gotoxy(Bx+9,By+4); break;
    case 1: gotoxy(Bx+9,By+2); break;
    case 2: gotoxy(Bx+15,By+2); break;
    case 3: gotoxy(Bx+15,By+4); break;
    case 4: gotoxy(Bx+15,By+6); break;
    case 5: gotoxy(Bx+9,By+6); break;
    case 6: gotoxy(Bx+3,By+6); break;
    case 7: gotoxy(Bx+3,By+4); break;
    case 8: gotoxy(Bx+3,By+2); break;
   }
 }
```