

POLA DASAR PEMBUATAN MEDIA PENGAJARAN FISIKA DENGAN KOMPUTER MENGGUNAKAN BAHASA PASCAL

Disajikan dalam Lokakarya Pembuatan Media Pengajaran
dengan Komputer
31 Januari s.d 3 Februari 2000

Oleh: **Drs. Akmam, M.Si** JAGA DAN PERGUNAKANLAH KOLEKSI
INI DENGAN BAIK

SUATU SAAT ANAK DAN CUCU ANDA
SANGAT MEMBUTUKANNYA

MILIK PERPUSTAKAAN UNIV. NEGERI PADANG	
DITERIMA TGL.	: 12-11-2001
SUMBER/MARGA	: <i>Hadijah</i>
KOLEKSI	: <i>K</i>
NO. INVENTARIS	: <i>630/K/2001-p1/2</i>
KLASIFIKASI	: <i>571.334 Akam-10</i>

Jurusan Fisika
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI PADANG
2000

Pola Dasar Pembuatan Media Pengajaran Fisika Dengan Komputer Menggunakan Bahasa Pascal¹⁾

Drs. Akmam, M.Si²⁾

Awal abad ke-21 ditandai dengan pesatnya perkembangan ilmu dan teknologi termasuk teknologi komputer. Sejak awal kali komputer ditemukan, komputer berfungsi sebagai alat bantu untuk mempercepat proses perhitungan. Dalam perkembangan selanjutnya, seiring dengan banyaknya data yang harus diolah permasalahan menjadi besar. Hal ini tentu menimbulkan persoalan baru yaitu bagaimana data yang sangat besar bisa ditangani dan disajikan dengan cara yang lebih efektif dan efisien.

Kemudian orang mengembangkan suatu alat bantu yang bisa dipakai untuk menyajikan data dalam jumlah besar ke dalam sebuah bagan, sehingga lebih mudah untuk dipahami dan bahkan akan lebih bermakna dari jutan deretan angka yang ditampilkan. Dengan komputer, pada saat tertentu dan untuk keperluan tertentu biasanya orang lebih suka menggunakan bagan dari pada melihat deretan angka yang membosankan tersebut. Termasuk dalam melaksanakan proses belajar mengajar di dalam kelas. Sejalan dengan hal ini, ahli komputer mengembangkan perangkat lunak yang membuat komputer menjadi semakin pandai yaitu dengan menambah kemampuan grafis pada komputer-komputer. Perkembangan ini melahirkan suatu bidang kajian pada ilmu komputer yaitu komputer grafik.

Bidang kajian komputer grafik ini mempelajari tentang cara-cara untuk meningkatkan dan memudahkan komunikasi antara manusia dengan mesin dengan membangkitkan, menyimpan, dan memanipulasi gambar atau model suatu objek dengan menggunakan komputer. Dengan komputer grafik memungkinkan kita berkomunikasi dengan lewat gambar-gambar, bagan-bagan ataupun dengan diagram-diagram.

Fisika merupakan suatu ilmu yang empiris dimana pernyataan fisika harus didukung oleh hasil eksperimen. Disamping itu fisika juga merupakan abstraksi dari berbagai sifat alam dalam bentuk konsep-konsep yang merupakan hampiran (aproksimasi) terhadap realita. Sifat khusus fisika lainnya adalah kuantitatif yaitu penggunaan konsep-konsep yang banyak dengan menggunakan peralatan matematika.

¹⁾ Disampaikan pada Lokakarya Pembuatan Media Pengajaran dengan Komputer 31-1-2000 sampai dengan 3 - 2 - 2000

²⁾ Dosen laboratorium Fisika Komputasi Jurusan Fisika FMIPA UNP Padang

Sifat abstraksi, empiris dan penggunaan konsep matematika membuat komputer banyak berperan dalam pendidikan fisika dan perkembangan pada berbagai tingkat pendidikan. Dengan menggunakan komputer konsep-konsep yang abstrak dapat dikongkritkan pada layar monitor melalui suatu visualisasi statis ataupun dinamis. Proses yang terlalu cepat, terlalu lambat, terlalu besar, berbahaya dan sebagainya dapat ditampilkan dilayar dan dipelajari perilakunya.

Dengan perkembangan komputer grafik mendorong berkembangnya pemanfaatan komputer sebagai alat bantu (media) untuk mengajar. Alat bantu tersebut dibuat dalam bentuk software. Walaupun demikian pembuatan software media pendidikan ini cukup lamban perkembangannya. Hal disebabkan oleh membuat program tersebut membutuhkan waktu yang lama dan mahal. Maka di Negara maju tidak banyak peminat mengembangkan media pengajaran. Fasilitas mendukung pendidikan, CSE, *computer-supported education*, dapat berupa (CMI, *computer managed instruction*, CAI, *computer-assisted instruction*, CBL, *computer-based learning*, CAL, *computer-assisted learning*). (Mitzel : 1974, Nobar 1996).

Sebagai tahap awal pengembangan/pembuatan media pengajaran melalui komputer sebagai alat bantu pendidikan, maka dalam makalah ini akan dijelaskan secara ringkas tentang cara mencipta media pengajaran sederhana dengan menggunakan trik-trik pemrograman tanpa mengabaikan kaidah-kaidah dalam fisika.

A. Superposisi Dua Gelombang.

Ambil bentuk dua persamaan gelombang sederhana:

$$(1) \quad y_1 = a \sin(\omega_1 t)$$

$$(2) \quad y_2 = a \cos(\omega_2 t)$$

dimana superposisi kedua gelombang di atas dapat ditulis menjadi

$$(3) \quad y = y_1 + y_2$$

Kerana kita telah menggunakan komputer, penjumlahan kedua gelombang di atas tidak perlu dilakukan secara analisi, cukup ditulis persamaan saja

Perlu diperhatikan disini bahwa dengan pada komputer sudut harus dalam bentuk radian.

Langkah-langkah membuat program di atas adalah sebagai berikut:

1. Tentukan jumlah gelombang (n , amplitudo gelombang)

2. Hitung posisi partikel untuk setiap saat dengan persamaan (1) dan (2)
3. Hitung superposisi kedua gelombang

Untuk tahap awal salinlah program berikut dengan tepat dan benar, kemudian buatlah agar mempunyai penampilan yang bagus dan dapat berfungsi sebagai media penyampaian informasi yang bermanfaat.

Program 1. Grafik Sinus

```
PROGRAM grafik_Sinus;
USES CRT,GRAPH;
Var
Gd,Gm,t,n,n2,i   : Integer;
y,y2,y3         : real;
Begin
  Gd := Vga;
  Gm := VgaHi;
  write('banyak gelombang sinus yang inginkan : ');readln(n);
  write('banyak gelombang cosinus yang inginkan : ');readln(n2);
  initgraph(gd,gm,'C:\TP\BGI ');
  rectangle(0,0,getmaxx,getmaxy);
  line(100,30,getmaxx,30);
  line(100,100,getmaxx,100);
  line(100,160,getmaxx,160);
  line(100,0,100,getmaxy);
  for t:=0 to 539 do
  begin
    y := -10 *sin(((t-100)*pi/180)*n*360/539);
    putpixel(100+t,round(y)+30,1);
    y2 :=-20*cos(((t)*pi/180)*(n2)*360/539);
    putpixel(100+t,round(y2)+100,1);
    y3 := y +y2;
    putpixel(100+t,round(y3) +160,1);
  end;
  settextstyle(smallfont,1,4);
  outtextxy(10,110,'superposisinya');
  outtextxy(40,1,'gelombang sinus ');
  outtextxy(80,45,'gelombang cosinus ');
  readln;
end.
```

B. Gerak Melingkar

Aplikasi lain dari trik pemrograman sinus dan cosinus di atas adalah membuat gerak melingkar dengan menguraikan komponen vektor posisi r

menjadi komponen x dan y dimana $r = (x^2 + y^2)^{1/2}$ dengan

$$x = r \cos \theta \text{ dan } y = r \sin \theta$$

Dalam program ini diaplikasikan tipe data pointer untuk membuat citra yang disimpan pada memori yang telah di bahas pada pemakalah terdahulu. Untuk lebih memahami pembuatan program gerak melingkar ini editlah dan sempurnakanlah agar diperoleh sebuah media pengajaran.

Program 2. Gerak Melingkar

```

program Gerak_Melinkar;
uses crt,graph;
var
gd,gm,i:integer;
pilih,jawab,ulang : char;
procedure gambar;
var
i:integer;
s : integer;
xmax,ymax:integer;
x,y :integer;
putar :pointer;
size :word;
begin
gd := vga;
gm := vgaHi;
pieslice(20,20,0,360,3);
size:=imagesize(16,16,23,22);
getmem(putar,size);
getimage(15,15,24,23,putar^);
cleardevice;
moveto(getmaxx,9*getmaxy div 10);
settextstyle(100,horizdir,10);
settextjustify(center,center);
pieslice(getmaxx div 2,getmaxy div 2,0,360,20);
setcolor(red);
outtextxy(getmaxx div 2,getmaxy div 2 -8,'inti');
setcolor(5);
outtextxy(getmaxx div 2,getmaxy div 2,'+');
setbkcolor(2);
setcolor(red);
repeat
s := s +1;
y:=round(40*cos(s*10*pi/300));
x:=round(150*sin(s*10*pi/300));
putimage(getmaxx div 2+x,getmaxy div 2+y,putar^,copyput);
until keypressed;
end;
{PROGRAM UTAMA.....}
BEGIN
clrscr;
gd:=detect;
ulang :='Y';

```

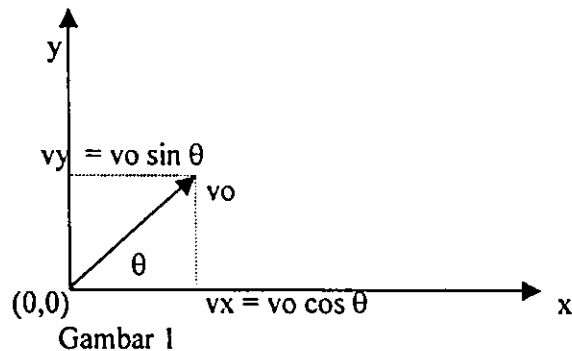
```

while(ulang='Y') or (ulang='y') do
begin
Initgraph(gd,gm,'c:\Tp\bgi');
if graphresult < 0 then halt (1);
gambar;
closegraph;
NOSOUND;
writeln( 'Anda mau ulang Y/T : ');readln(ulang);
end;
end.

```

C. Lintasan Gerak Pada Peluru

Umpamakan sebuah peluru/bola pejal dilemparkan dengan membantuk sudut θ terhadap bidang horizontal dengan kecepatan awal v_0 . Untuk memudahkan perhitungan biasanya vektor kecepatan diuraikan menjadi komponen kecepatan arah sumbu-x (v_x) dan sumbu-y (v_y). Apabila gesekan peluru dengan udara sangat kecil dibandingkan dengan gaya tarik bumi terhadap peluru sehingga dapat diabaikan, maka peluru akan membentuk lintasan parabolik. Perhatikan gambar 3 berikut:



Berdasarkan gambar 1 di atas, persamaan gerak peluru tersebut ditentukan dengan menggunakan persamaan parameter sebagai berikut:

$$x = v_0 \cos \theta \cdot t$$

$$y = v_0 \sin \theta \cdot t - 0.5 g t^2$$

Untuk membentuk simulasi lintas peluru di atas pada layar monitor dapat dilakukan langkah-langkah sebagai berikut:

1. Tentukan besar sudut elevasi (θ) dan kecepatan lemparan awal (v_0).
2. Buat kondisi awal pada saat $t=0$, $v = v_0$, $x = x_0$, $y = y_0$,
3. Hitung komponen kecepatan awal dalam arah sumbu-x dan sumbu-y
 $v_x = v_0 \cos \theta$ dan $v_y = v_0 \sin \theta$
4. Hitung posisi partikel setiap saat dengan persamaan:

$$x = v_0 \cos \theta \cdot t$$

$$y = v_0 \sin \theta \cdot t - 0.5 g t^2$$

5. Lakukan pengulangan perhitungan dalam selang waktu dt . (ambil $dt = 0.001$)

Untuk memudahkan pengertian peserta editlah program gerak peluru berikut dengan tepat, kemudian modifikasilah agar menjadi sebuah media pengajaran yang baik

Program 3 Program lintasan gerak peluru.

```

Program Gerak_Peluru;
uses crt,graph;
var
gd,gm :integer;
vo,teta,x,y,t,dt,voy,vox : real;
bola : pointer;
size : word;
begin
gd := vga;
gm := vgaHi;
initgraph(gd,gm,'c:\tp\bgi');
outtextxy(8,4,'Kecepatan Awal : ');write(' ');readln(vo);
outtextxy(8,22,'sudut elevasi : '); write (' ');readln(teta);
teta := teta*pi/180;
voy := vo*sin(teta);
vox := vo*cos(teta);
dt := 0.001; t :=0;
initgraph(gd,gm,'c:\tp\bgi');
pieslice(20,100,0,360,5);
size := imagesize(15,95,25,105);
getmem(bola,size);
getimage(14,94,25,106,bola^);
cleardevice;
line(10,getmaxy div 50,10,9*getmaxy div 10);
line(10,9*getmaxy div 10,9*getmaxx div 10, 9*getmaxy div 10 );
repeat
t := t + dt;
x := vox*t;
y := voy*t - 0.5*t*t;
putimage(10+ round(x),(9*getmaxy div 10)-round(y),bola^,copyput);
putimage(10+ round(x),(9*getmaxy div 10)-round(y),bola^,copyput);
putpixel(10+ round(x),(9*getmaxy div 10)-round(y)+5,green);
line(10,getmaxy div 50,10,9*getmaxy div 10);
line(10,9*getmaxy div 10,9*getmaxx div 10, 9*getmaxy div 10 );
until (y < 0) or (x > 9*getmaxx div 10);
outtextxy(7*getmaxx div 8,19*getmaxy div 20,'Tekan Esc');
repeat until readkey = #27;
closegraph;
end.

```

D. Program Medan Potensial Muatan Listrik

Setiap titik dalam ruangan dekat bumi dapat dipandang sebagai sebuah vektor dibawah pengaruh medan gravitasi bumi (g). Vektor ini merupakan percepatan gravitasi yang akan dialami oleh sebuah partikel yang ditempatkan pada titik tersebut. Bila m adalah massa partikel dan F adalah gaya gravitasi yang berkerja pada partikel tersebut, maka g dapat rumuskan dengan:

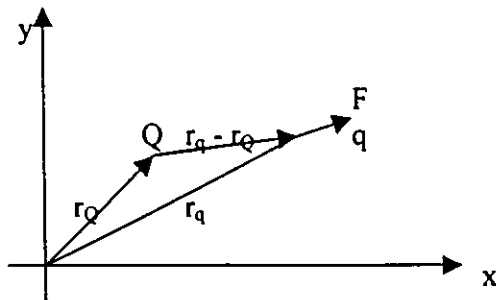
$$\vec{g} = \frac{\vec{F}}{m} \quad 1$$

Demikian pula halnya dengan medan listrik, dimana bila sebuah muatan uji (test charge) ditempatkan dalam ruang medan dibawah pengaruh sebuah benda bermuatan, maka gaya elektrostatis akan bekerja pada muatan tersebut. Maka dikatakan bahwa di dalam ruangan tersebut terdapat medan listrik. Hal yang sama juga berlaku untuk medan magnet.

Dalam makalah ini dipelajari cara memvisualisasikan medan listrik dan garis eksponensial dari potensial listrik dengan bantuan komputer. Untuk maksud di atas terlebih dahulu ditinjau pengertian Gaya listrik dan kuat medan listrik.

1. Gaya Listrik

Sekarang tinjau ruangan yang di dalamnya terdapat 2 partikel bermuatan Q dan q seperti gambar 1 berikut:



Gambar 2: Gaya Listrik pada Dua Muatan Listrik

Berdasarkan hukum Coulomb didefensikan bahwa gaya listrik oleh muatan Q pada muatan q adalah:

$$F = k \frac{Qq}{|r_q - r_Q|^3} (r_q - r_Q) \quad (2)$$

dengan $(r_q - r_Q)$ vektor jarak dari Q ke q , $k = 8,988 \times 10^9 \text{ Nm}^2\text{c}^{-2}$ dan muatan Q dan q dalam satuan Coulomb. Dari persamaan (2) terlihat bahwa gaya F akan

mempunyai arah menjauhi Q dari q jika tanda muatan q dan Q sama., sebaliknya gaya F akan mempunyai arah dari Q jika tanda q dan Q berbeda.

2. Kuat Medan Listrik, E

Sebelum munculnya teori Faraday, gaya yang bekerja di antara partikel-partikel bermuatan dipandang sebagai suatu interaksi sesaat yang berlangsung di antara kedua partikel tersebut. Padang aksi pada suatu jarak (action at a distance) tersebut ini juga berlaku untuk gaya magnetik dan gaya gravitasi. Setelah Faraday muncul pemikiran bahwa:

1. Muatan Q (seperti terlihat pada gambar (1)) menimbulkan sebuah medan listrik di dalam ruangan disekitarnya
2. Medan tsb. bekerja pada muatan q dalam bentuk gaya F yang dialami oleh q. Medan ini memainkan peranan perantara pada gaya-gaya diantara muatan, seperti dalam kasus:
 - a. menghitung medan yang ditimbulkan oleh distribusi muatan yang diberikan
 - b. menghitung gaya-gaya yang akan diberikan pada muatan-muatan oleh medan-medan yang ditempatkan di dalam medan-medan tersebut.

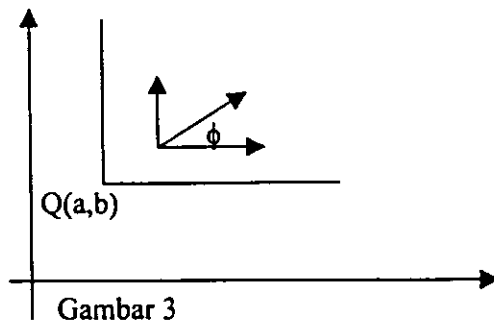
Bila medan listrik pada suatu titik didefenisikan sebagai perbandingan antara gaya Coulomb (F) dengan muatan (q), dimana q menuju nol atau

$$\vec{E} = \lim_{q \rightarrow 0} \frac{\vec{F}}{q} \quad 3$$

Jika Q adalah suatu muatan titik terisolasi, maka persamaan (4) dan (5) berikut dapat digunakan untuk menghitung medan listrik di dekat muatan Q pada suatu titik yang didefenisikan oleh vektor perpindahan:

$$\vec{E} = k \frac{Q}{|\vec{r}_q - \vec{r}_Q|^3} (\vec{r}_q - \vec{r}_Q)$$

Untuk kasus khusus 2 dimensi, apabila muatan Q terletak pada titik dengan koordinat (a,b), untuk pemograman diharapkan diperoleh komponen x dan y dan medan listrik pada titik x dan y terhadap muatan di posisi (x,y) dan medan listrik pada titik x dan y terhadap muatan Q sendiri. (seperti gambar 3).



Kita harus melakukan perhitungan berikut untuk mendapatkan komponen x dan y dari medan listrik, E, pada titik (x,y):

Umpamakan

$$x_d = x - a \quad 6$$

$$y_d = y - b \quad 7$$

Sedangkan

$$r = (x_d^2 + y_d^2)^{1/2} \quad 8$$

maka diperoleh

$$E_x = E \cos \Phi = kQ \frac{x_d}{(x_d^2 + y_d^2)^{3/2}} \quad 9$$

dan

$$E_y = E \sin \Phi = kQ \frac{y_d}{(x_d^2 + y_d^2)^{3/2}} \quad 10$$

Persamaan 6 – 10 di atas akan digunakan pada program berikut ini untuk melukis garis medan di antara dua muatan listrik.

Algoritma : Melukis Garis Medan 2 muatan listrik

Langkah 1 : tentukan nilai Q,q,k dan posisi masing-masing muatan (x₀,y₀ dan x,y (untuk mudahnya ambil k=1)

langkah 2 : Hitung jarak kedua muatan dari titik acuan (ambil 0,0)

$$x_d = x - a \quad 11$$

$$y_d = y - b$$

$$r = (x_d^2 + y_d^2)^{1/2}$$

langkah 3 : Hitung E_x dan E_y dengan rumus

$$E_x = E \cos \Phi = kQ \frac{x_d}{(x_d^2 + y_d^2)^{3/2}} \quad 12$$

$$E_y = E \sin \Phi = kQ \frac{y_d}{(x_d^2 + y_d^2)^{1/2}} \quad 13$$

langkah 4 : Hitung vektor posisi

$$x := x + E_x/E * dx; \quad 14$$

$$y := y + E_y/E * dy;$$

Untuk lebih mudah memahami konsep di atas editlah program berikut, kemudian modifikasilah untuk mendapat media yang lebih baik.

Program 4. Garis Gaya muatan listrik.

Program Medan_Muatan_Listrik;

Uses Crt,Graph;

var

i,gd,gm : Integer;

q1,q2 : real;

Procedure dipole (Q1,Q2,x1,x2,yo,x,y : real);

var

dy,dx : Integer;

ds,r2,r1,ex,ey,e : Real;

keluar : Boolean;

begin

ds:= 0.5; dx:= 1; dy:= 1;

keluar := false;

circle(round(x1),round(yo),4);

settextstyle(2,0,4);

outtextxy(round(x1)-70,round(yo),'muatan 1');

circle(round(x2),round(yo),4);

settextstyle(2,0,4);

outtextxy(round(x1)+200,round(yo),'muatan 2');

repeat

r1 := sqrt(sqr(x-x1) + sqr(y-yo));

r2 := sqrt(sqr(x-x2) + sqr(y-yo));

ex :=q1*(x-x1)/(sqr(r1)*r1) + q2*(x-x2)/(sqr(r2)*r2);

ey :=q1*(y-yo)/(sqr(r1)*r1) + q2*(y-yo)/(sqr(r2)*r2);

E := Sqrt(sqr(Ex) + sqr(Ey));

x := x + Ex/E *dx;

y := y + Ey/E *dy;

putpixel(round(x),round(y),1);

putpixel(round(x),getmaxy-round(y),1);

if (x>getmaxx) or (x<1) or (y<1) or (y>getmaxy) then

keluar := True;

if (round(x)= round(x2)) and (round(y) = round(yo)) then

keluar := True;

until keluar;

end;

BEGIN

gd := detect;

```

write('Masukan harga Muatan Q1 (coulom : ');Readln(q1);
write('Masukan harga Muatan Q2 (coulom : ');Readln(q2);
initgraph (gd,gm,'C:\TP\BGI');
rectangle(0,0,getmaxx,getmaxy);
for i := 1 to 8 do
{dipole (Q1,Q2,x1,x2,yo,x,y : real);}
dipole(q1,q2,200,350,getmaxy div 2,196+8*i,((getmaxy div 2)-4));
for i := 8 downto 1 do
dipole(q1,q2,200,350,getmaxy div 2,354-8*i,((getmaxy div 2)-4));
settextstyle(3,0,2);
outtextxy(450,9*getmaxy div 10,' Terus Tekan Enter');
repeat until keypressed;
Closegraph;
end.

```

Berikutnya kerangka awal yang dapat dikembangkan untuk membuat media media pengajaran untuk gelombang berjalan. Editlah program berikut ini kemudian modifikasilah untuk dapat memperlihatkan superposisi dari 2 gelombang berjalan tersebut.

Program 5. Membuat Gelombang Begerak.

```

Program Gelombang_berjalan;
uses crt,graph;
var
gd,gm : integer;
x : integer;
y : array [0..720] of integer;
frekuensi,loop,frek : integer;
gelomb, gelombang : array [0..9] of pointer;
Procedure buat_gelombang(amplitudo,frek:integer);
begin
for x := 0 to 720 do
begin
y[x] := round(-amplitudo * sin(x*pi/180*frek));
end;
end;

Procedure gambar_gelombang(awal,akhir:integer);
begin
for x := awal to akhir do
begin
putpixel(x-awal,y[x]+64,white);
end;
end;

Procedure buat_gelomb(amplitudo,frek:integer);
begin
for x := 0 to 720 do

```

```

begin
  y[x] := round(-amplitudo * sin(x*pi/180*frek)*cos(x*pi/180*frek));
end;
end;

Procedure gambar_gelomb(awal,akhir:integer);
begin
  for x := awal to akhir do
  begin
    putpixel(x-awal,y[x]+124,red);
  end;
end;

begin
  gd := vga;
  gm := vgaHi;
  initgraph(gd,gm,'c:\tp\bgi');
  frekuensi := 2;
  frek := 6;
  outtextxy(20,50,'tunggu');
  buat_gelombang(15,frekuensi);
  buat_gelomb(15,frek);
  cleardevice;
  for loop := 0 to 9 do
  begin
    rectangle(0,45+60,360,85+60);
    gambar_gelomb(0+round(36/frek)*loop,360+round(36/frek)*loop);
    getmem(gelombang[loop],imagesize(0,45+60,360,85+60));
    getimage(0,45+60,360,85+60,gelombang[loop]^);
    cleardevice;
  end;
  for loop := 0 to 9 do
  begin
    rectangle(0,45+60,360,85+60);
    gambar_gelomb(0+round(36/frekuensi)*loop,360+round(36/frekuensi)*loop);
    getmem(gelomb[loop],imagesize(0,45,360,85));
    getimage(0,45,360,85,gelomb[loop]^);
    cleardevice;
  end;
  loop := 0;
  repeat
    putimage(10,60,gelombang[loop]^,normalput);
    delay(500);
    putimage(10,100,gelomb[loop]^,normalput);
    loop := loop + 1;
  until loop = 10 then loop := 0;
  until keypressed;
  repeat until keypressed;
  closegraph;
end.

```

E. Program Transformasi Objek

Jenis transformasi objek yang banyak digunakan di dalam komputer grafik adalah transformasi afin (affine transformation). Transformasi ini sangat sederhana. Dalam transformasi 2-D fungsi transformasi T akan memetakan $P = (P_x, P_y)$ menjadi $Q = (Q_x, Q_y)$, dimana Q_x dan Q_y dihubungkan dengan P_x dan P_y melalui persamaan:

$$\begin{aligned} Q_x &= aP_x + cP_y + TR_x \\ Q_y &= bP_x + dP_y + TR_y \end{aligned} \quad 15$$

dimana a,b,c,d TR_x , TR_y merupakan sembarang konstanta.

Dengan menggunakan notasi matriks persamaan di atas dapat ditulis menjadi:

$$(Q_x, Q_y) = (P_x, P_y) \begin{pmatrix} a & b \\ c & d \end{pmatrix} + (TR_x, TR_y) \quad 16$$

Persamaan di atas dapat ditulis dalam notasi yang sederhana yaitu:

$$Q = PM + TR \quad 17$$

dengan $TR = (TR_x, TR_y)$ yang disebut juga dengan offset vector dan M adalah matriks berukuran (2 x 2).

$$M \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad 18$$

Transformasi yang dilakukan dengan transformasi ini adalah pergeseran, penskalaan, pemutaran, pergeseran. Karena terbatas waktu maka dalam makalah ini hanya dibahas tentang transformasi putar dan geser.

1. Transformasi Shearing

Sembarang titik pada bidang xy digeser kesembarang tempat dengan menambahkan besaran pada sumbu-x dan sumbu-y. Dengan menggunakan persamaan $Q = PM + TR$, maka dihasilkan pergeseran dengan persamaan:

$$(Q_x, Q_y) = (P_x + TR_x, P_y + TR_y) \quad 20$$

dan matriks M adalah matriks identitas.

Shearing merupakan suatu proses untuk mentransformasikan objek dengan cara membebani objek tersebut pada arah tertentu, sehingga dihasilkan suatu objek yang terdistorsi. Proses shearing kearah sumbu-x dapat ditunjukkan dengan menggunakan persamaan:

$$\begin{aligned} Q_x &= P_x + hP_y \\ Q_y &= P_y \end{aligned} \quad 21$$

dengan h menunjukkan bagian ordinat yang ditambahkan ke absis.

Secara umum proses ke arah sumbu- x dan sumbu- y dinyatakan dengan :

$$\begin{aligned} Q_x &= P_x + hP_y \\ Q_y &= gP_x + P_y \end{aligned} \quad 22$$

3. Transformasi Pemutaran

Pemutaran suatu objek dapat dilakukan dengan memutar setiap titik diujung setiap garis. Dalam pemutaran objek dapat diputar searah dengan jarum jam atau berlawanan dengan jarum jam. Dengan menganggap besar sudut pemutaran θ , maka persamaan matrik untuk pemutaran adalah:

$$\begin{bmatrix} Q_x \\ Q_y \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} P_x \\ P_y \end{bmatrix} \quad 23$$

Dari semua transformasi di atas terlihat bahwa transformasi geser mempunyai bentuk yang agak beda dengan transformasi yang lain, dimana pada transformasi geser terdapat operasi penjumlahan sedang pada transformasi yang terdapat operasi perkalian matriks. Agar transformasi pergeseran dapat dimasukkan ke dalam transformasi yang lain, maka perlu buat sistem koordinat homogen.

Sistem koordinat homogen yang dimaksudkan disini adalah membuat sistem koordinat yang mempunyai dimensi satu lebih tinggi dari sistem koordinat objek yang akan ditransformasikan. Dengan menggunakan sistem koordinat homogen di atas maka diperolehlah matrik transformasi untuk kasus dua dimensi seperti di atas sebagai berikut:

$$\text{Matriks transformasi Pegeseran: } M_T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ TR_x & TR_y & 1 \end{bmatrix}$$

$$\text{Matriks transformasi Shearing: } M_{SH} = \begin{bmatrix} 1 & g & 0 \\ h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Matriks transformasi Pemutaran : } M_R = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Untuk lebih memahami teknik transformasi diatas kerjakan program berikut, kemudian sempurnakan agar diperoleh hasil yang lebih baik

Program 6. Transformasi Putar Pada Titik Tertentu.

```
Program transformasi;
uses
  crt, graph ;
var
  ymax, xmax, ymin, xmin :real;
  drive, mode, i : integer;
  x1, x2, y1, y2, yy, yx, yyy, yyx :real;
  B :Array [0..4] of real;
  teta :real;
  ukur :integer;
  kotak:pointer;
  procedure kali_matrik(x1, x2, sudut:real;var y1, y2 :real);
  var
    C :Array[0..4,0..4] of real;
    A,B:Array[0..4] of real;
    i, j, k :integer;
  begin
    a[0]:=X1;
    a[1]:=X2;
    sudut:=sudut*pi/180;
    C[0,0] :=cos(sudut);
    C[0,1] :=sin(sudut);
    C[1,0] :=-sin(sudut);
    C[1,1] :=cos(sudut);
    for i:= 0 to 1 do
      begin
        b[i] :=0;
        for j:= 0 to 1 do
          begin
            b[i]:=b[i]+c[i,j]*a[j];
          end;
        end;
      end;
    y1:=b[0]; y2:=b[1];
  end;
  procedure bidang_gambar(var x_min,y_min,x_max,y_max:real);
  begin
    X_min :=-800;
    Y_min :=-800;
    X_max :=800;
    y_max :=800;
  end;
  procedure trans(X_min,Y_min,x_max,Y_max,x_G,Y_g:real;var
X_1,Y_1:real);
  begin
    setbkcolor(blue);
    X_1:= getmaxx/(x_max-x_min)*(-x_min+x_g);
    Y_1:= getmaxy/(Y_max-y_min)*(y_max-y_g);
  end;
  begin
```



```

clrscr;
write('Sudut Putar ');readln(teta);
teta := -teta+40;
bidang_gambar(Xmin,Ymin,Xmax,Ymax);
drive:=Vga;
mode:=Vgahi;
initgraph(drive,mode,'c:\tp\bgi');
setcolor(yellow);
rectangle(2,2,40,40);
ukur:=imagesize(2,2,40,40);
getmem(kotak,ukur);
getimage(1,1,51,51,kotak^);
cleardevice;
trans(Xmin,Ymin,Xmax,Ymax,xmin,0,x1,y1);
trans(Xmin,Ymin,Xmax,Ymax,xmax,0,x2,y2);
line(round(x1),round(y1),round(x2),round(y2));
trans(Xmin,Ymin,Xmax,Ymax,0,ymin,x1,y1);
trans(Xmin,Ymin,Xmax,Ymax,0,ymax,x2,y2);
line(round(x1),round(y1),round(x2),round(y2));
kali_matrik(400,400,teta,yy,yx);
trans(Xmin,Ymin,Xmax,Ymax,yy,yx,x1,y1);
putimage(round(x1),round(y1),kotak^,normalput);
delay(20);
trans(Xmin,Ymin,Xmax,Ymax,yy,yx,x1,y1);
putimage(round(x1),round(y1),kotak^,normalput);
outtextxy(3*getmaxx div 2, getmaxy div 2,'pusat');
readln;
closegraph;
end.

```

Program 7: transformasi_dengan_koordinat_Bebas;

```

uses crt,graph;
type poli=array[1..6] of pointtype;
    mat =array[1..3,1..3] of real;
    vek =array[1..3] of real;
const sl:poli=((x:300;y:150),(x:350;y:100),
              (x:400;y:150),(x:400;y:300),
              (x:300;y:300),(x:300;y:150));
{ xp=0; yp=0; {sudut 0}
{ xp=0; yp=600; {sudut 90}
{ xp=600; yp=600;} {sudut 180}
{ xp=600; yp=0; {sudut 270}
{ xp=-125; yp=300; {sudut 45}
var gd,gm,i,xp,yp : integer;
    sl1,sl2,sl3 : poli;
    vekh : vek;
    sdt : real;
    pil,ul : char;
{prosedur untuk membentuk matrik transformasi
untuk penggeseran objek (mt)

```

630/k/2001-p1/2

371.334

AKM

PC

```
    trx : penggeseran ke arah x
    try : penggeseran ke arah y
procedure gs(var mt:mat; trx,try:integer);
begin
mt[1,1]:=1; mt[1,2]:=0; mt[1,3]:=0;
mt[2,1]:=0; mt[2,2]:=1; mt[2,3]:=0;
mt[3,1]:=trx; mt[3,2]:=try; mt[3,3]:=1;
end;
{prosedur untuk membentuk matrik transformasi
untuk penskalaan objek (ms)
  sx : faktor skala ke arah x
  sy : faktor skala ke arah y}
procedure sk(var ms:mat; sx,sy:real);
begin
ms[1,1]:=sx; ms[1,2]:=0; ms[1,3]:=0;
ms[2,1]:=0; ms[2,2]:=sy; ms[2,3]:=0;
ms[3,1]:=0; ms[3,2]:=0; ms[3,3]:=1;
end;
{prosedur untuk membentuk matrik transformasi
untuk pemutaran objek (mr)
  tt : sudut putar}
procedure rts(var mr:mat; tt:real);
begin
tt:=tt*pi/180;
mr[1,1]:=cos(tt); mr[1,2]:=sin(tt); mr[1,3]:=0;
mr[2,3]:=0; mr[2,3]:=0;
mr[2,1]:=-sin(tt); mr[2,2]:=cos(tt);
mr[3,1]:=0; mr[3,2]:=0; mr[3,3]:=1;
end;
{prosedur untuk membentuk matrik transformasi
untuk men-shear objek (msh)
  sh : shear factor ke arah x
  sg : shear factor ke arah y}
procedure geser(var msh:mat; sh,sg:real);
begin
msh[1,1]:=1; msh[1,2]:=sg; msh[1,3]:=0;
msh[2,1]:=sh; msh[2,2]:=1; msh[2,3]:=0;
msh[3,1]:=0; msh[3,2]:=0; msh[3,3]:=1;
end;
{prosedur untuk mengalikan matrik dg matrik}
procedure pm(var mh:mat; ma,mb:mat);
var i,j,k : byte;
begin
for i:=1 to 3 do
for j:=1 to 3 do
begin
mh[i,j]:=0;
for k:=1 to 3 do
mh[i,j]:=mh[i,j]+ma[i,k]*mb[k,j];
end;
end;
```

```

end;
{prosedur untuk mengalikan vektor dg matrik}
procedure pv(var vh:vek; va:vek; ma:mat);
var i,j : byte;
begin
for i:=1 to 3 do
begin
vh[i]:=0;
for j:=1 to 3 do
vh[i]:=vh[i]+va[j]*ma[j,i];
end;
end;
{prosedur untuk mentransformasikan sembarang gambar}
procedure pg(s:real);
var i,j : byte;
vekh,vekt : vek;
matt, mata,matb : mat;
begin
gs(mata,-xp,-yp); {menggeser sumbu}
rts(matb,s); {memutar objek}
pm(matt,mata,matb);
for i:=1 to 6 do
begin
{membentuk koordinat homogen}
vekt[1]:=sl[i].x;
vekt[2]:=sl[i].y;
vekt[3]:=1;
pv(vekh,vekt,matt);
{kembali ke koordinat biasa}
sl2[i].x:=round(vekh[1]);
sl2[i].y:=round(vekh[2]);
{membentuk koordinat homogen}
vekt[1]:=sl1[i].x;
vekt[2]:=sl1[i].y;
vekt[3]:=1;
pv(vekh,vekt,matt);
{kembali ke koordinat biasa}
sl3[i].x:=round(vekh[1]);
sl3[i].y:=round(vekh[2]);
end;
end;
begin
clrscr;
gd:=detect; gm:=vgahi;
ul:='y';
while(ul='y')or(ul='Y')do
begin
gotoxy(5,4);writeln('Besarnya sudut putar untuk transformasi terhadap titik
bebas');
gotoxy(5,5);writeln('selain sudut 90, 180, 270 dan 360');

```

```

gotoxy(5,10);write('Sudut putar : '); readln(sdt);
initgraph(gd,gm,'c:\tp\bgi');
  if (sdt>=0) and (sdt<90) then begin xp:= 0; yp:=0; end;
  if (sdt>=90) and (sdt<180) then begin xp:= 0; yp:=600; end;
  if (sdt>=180) and (sdt<270) then begin xp:= 600; yp:=600; end;
  if (sdt>=270) and (sdt<360) then begin xp:= 600; yp:=0; end;
  if (sdt=360) then begin xp:= 0; yp:=0; end;
settextstyle(3,0,2); settextjustify(1,1);
outtextxy(getmaxx div 2,50,'TRANSFORMASI DENGAN TITIK BEBAS');
for i:=1 to 6 do
  begin
    {mengcopy bentuk rumah}
    sl1[i].x:=sl[i].x+100;
    sl1[i].y:=sl[i].y; end;
setcolor(14);
settextstyle(defaultfont,horizdir,1);
settextjustify(lefttext,centertext);
{menggambar bentuk objek asli}
outtextxy(510,300,'GAMBAR ASLI');
line(550,100,600,100); line(550,100,550,150);
outtextxy(610,100,'X+'); outtextxy(550,160,'Y+');
{hasil shearing}
drawpoly(6,sl); drawpoly(6,sl1);
setcolor(12); pg(sdt); {pg(45);}
drawpoly(6,sl2); drawpoly(6,sl3);
outtextxy(390,350,'HASIL PEMUTARAN');
readln; closegraph;
  gotoxy(5,14);write('Mau mengulang ?');
  gotoxy(5,20);write('Tekan y kalau ya dan t kalau tidak... '); readln(ul);
end; end.

```

Berikut ini adalah sebuah pola pokok program yang dapat dikembangkan menjadi program yang dapat digunakan untuk menjelas reaksi fusi.

Program 8 Reaksi Fusi.

```

Program Reaksi_Proton;
uses crt,graph;
var a_1, b, c, d, x, x1, x2, x3, y1, y2, y3, del : integer;
    pilih, ulang : char;
    y : real;
    s : string[3];
    mode, drive, i, j, k, a, n, dx, dy, l, m : integer;
    ukur, ukur1, ukur2, ukur3 : integer;
    bola, bola1, bola2, bola3 : pointer;
begin
  drive:=vga; mode:=vgahi;
  initgraph(drive,mode,'c:\tp\7bgi');
  rectangle(0,0,getmaxx,getmaxy); setbkcolor(3);
  line(15,170,25,170); line(20,165,20,175);
  circle(20,170,10);{+}

```



```

ukur:=imagesize(9,159,31,181);
getmem(bola,ukur);
getimage(9,159,31,181,bola^);
setcolor(red);
circle(220,173,10); setfillstyle(9,yellow); floodfill(220,173,red);
ukur1:=imagesize(209,162,231,184);
getmem(bola1,ukur1);
getimage(209,162,231,184,bola1^);
putimage(209,317,bola1^,ukur1);
delay(20);
dx:=1; dy:=1;
repeat
i:=i+dx; j:=j+dy;
putimage(i,160,bola^,copyput); putimage(i,320,bola^,copyput); {+kiri}
putimage(200,j,bola^,copyput); putimage(200,(getmaxy-4)-
j,bola^,copyput);{+atas}
delay(10);
if i=187 then begin dx:=0; end;
if j=142 then begin dy:=0; end;
until (i=187) and (j=142);
line(250,338,265,338); line(250,342,265,342);{=}
line(250,160,265,160); line(250,156,265,156);{=}
line(200,250,230,250); line(215,235,215,265);{+}
outtextxy(300,120,'He'); outtextxy(292,115,'3'); outtextxy(292,125,'2');
ukur2:=imagesize(188,139,241,199);
getmem(bola2,ukur2);
getimage(188,139,241,199,bola2^);
putimage(288,139,bola2^,copyput);
ukur3:=imagesize(188,319,241,355);
getmem(bola3,ukur3);
getimage(188,319,241,355,bola3^);
putimage(288,319,bola3^,copyput); delay(10);
dy:=1;
repeat
k:=k+dy;
putimage(288,319-k,bola3^,normalput);
delay(10); until k=158;
dy:=1;
repeat
l:=l+dy;
line(350,170,350+l,170); {tengah}
putimage(414,139,bola^,copyput);putimage(414,172,bola^,copyput);
putimage(402,157,bola1^,ukur1); putimage(423,157,bola1^,ukur1);
outtextxy(470,160,'He'); outtextxy(460,155,'4'); outtextxy(460,166,'2');
line(350,170,350+l,170-(2*l)); putimage(399,60,bola^,copyput); {atas}
outtextxy(470,60,'H'); outtextxy(460,55,'1'); outtextxy(460,65,'1');
line(350,170,350+l,170+(2*l)); putimage(399,260,bola^,copyput);{bawah}
outtextxy(470,265,'H'); outtextxy(460,260,'1'); outtextxy(460,270,'1');
until l=40; delay(80);

```

```

settextstyle(7,0,2); setcolor(15);
outtextxy(50,30,'TRANSFORMASI NUKLIR');
settextstyle(3,0,1); setcolor(14);
outtextxy(60,60,'Siklus Proton-Proton');
outtextxy(30,375,'Ini adalah salah satu dari deretan Reaksi Nuklir yang
terjadi dalam');
outtextxy(30,400,'bentang yang melibatkan penggabungan empat inti
Hidrogen mem');
outtextxy(30,425,'bentuk inti Helium disertai dengan Pelepasan Energi..');
readln; closegraph;
end.

```

Berikut ini adalah program yang dapat dikembangkan untuk menjelaskan proses tumbukan . Program ini merupakan hasil kerja mahasiswa.

```

PROGRAM Tumbukan;
uses crt,graph; var dx1,dx2,x1,x2,x:integer;
ukuran:word;
bola1:pointer;
pilih:char;
gd,gm:integer;

PROCEDURE BOLA;
begin
  {buat bola}
  setcolor(3);
  setfillstyle(1,white);
  circle(70,70,10);
  floodfill(70,70,3);
  {simpan bola}
  ukuran:=imagesize(60,60,80,80);
  getmem(bola1,ukuran);
  getimage(59,59,81,81,bola1^);
  cleardevice;
end;

PROCEDURE BUKA;
begin
  gd:=vga;gm:=vgahi;
  initgraph(gd,gm,'c:\tp7\bgi');
end;

PROCEDURE TUTUP;
begin
  readln;
  closegraph;
end;

PROCEDURE ENTER;
begin
  repeat

```

```

settextjustify(0,0);
outtextxy(200,400,'Press ENTER to CONTINUE...!! ');
Delay(500);
cleardevice;
delay(500);
until keypressed;
end;
PROCEDURE TEXT3;
begin
settextjustify(0,0);
rectangle(150,10,500,120);
settextstyle(0,0,0);setcolor(yellow);
outtextxy(200,40,'Tumbukan dua buah benda A dan B');
outtextxy(200,50,'A diberikan kecepatan dan B dalam');
outtextxy(200,60,'keadaan diam, setelah bertumbukan');
outtextxy(200,70,'B ikut bergerak. ');
outtextxy(200,400,'Press ENTER to CONTINUE...!! ');
end;
PROCEDURE TEXT2;
begin
settextjustify(0,0);
delay(500);cleardevice;delay(500);
rectangle(150,10,550,120);
settextstyle(0,0,0);setcolor(yellow);
outtextxy(200,40,'Tumbukan dua buah benda A dan B');
outtextxy(200,50,'A dan B sama-sama diberikan kecepatan');
outtextxy(200,60,'dan terjadi tumbukan lenting sempurna. ');
outtextxy(200,400,'Press ENTER to CONTINUE...!! ');
end;
PROCEDURE TEXT1;
begin
settextjustify(0,0);
rectangle(150,10,550,120);
settextstyle(0,0,0);setcolor(yellow+blink);
outtextxy(200,40,'Tumbukan dua buah benda A dan B');
outtextxy(200,50,'A diberikan kecepatan sedangkan B');
outtextxy(200,60,'dalam keadaan diam, setelah bertumbukan');
outtextxy(200,70,'B bergerak karena mendapatkan gaya dari A. ');
outtextxy(200,400,'Press ENTER to CONTINUE...!! ');
end;
PROCEDURE MOMENTUM_1;
begin
BOLA;
dx1:=1;x1:=0;
pieslice(215,212,0,360,10);
TEXT1;
setbkcolor(8);
repeat
x1:=x1+dx1;sound(300);delay(1);nosound;
putimage(x1,200,bola1^,copyput);

```



```

    delay(1);
until x1>=195;
    x2:=200;
repeat
    x2:=x2+dx1;
    putimage(15+x2,200,bola1^,copyput);sound(random(6000));nosound;
    delay(1);
until x2>getmaxx;
    ENTER;
    freemem(bola1,ukuran);
end;
PROCEDURE MOMENTUM_2;
begin
BOLA;
    {proses}
    x1:=0; x2:=0;
    dx1:=1;dx2:=1;
TEXT2;
    setbkcolor(9);
repeat
    x1:=x1+dx1;
    x2:=x2+dx2;
    PutImage(x1, getmaxy div 2,
bola1^,copyput);sound(random(6000));delay(2);nosound;
    PutImage(getmaxx-x2, getmaxy div 2,
bola1^,copyput);sound(random(5870));delay(2);nosound;
    delay(5);
    if (x1+x2+20) > getmaxx then dx1 := -dx1;
    if x1 < 0 then dx1 := -dx1;
    if (x2+x1+20) > getmaxx then dx2 := -dx2;
    if x2 < 0 then dx2 := -dx2;
until (x1<0);
    ENTER;freemem(bola1,ukuran);
end;
PROCEDURE MOMENTUM_3;
var a,b:integer;
begin
    BOLA;
    {proses}
    dx1:=1;x1:=0;
    pieslice(315,210,0,360,10);
TEXT3;
    setbkcolor(7);
repeat
    x1:=x1+dx1;
    putimage(x1,200,bola1^,copyput);sound(random(6000));delay(2);nosound;
    delay(3);
    putimage(x1,200,bola1^,xorput);
    delay(3);
until x1>=295;

```

```

x2:=200;
repeat
  x2:=x2+dx1;
  putimage(115+x2,200,bola1^,copyput);
  delay(3);
  putimage(95+x2,200,bola1^,xorput);
  delay(2);
  putimage(95+x2,200,bola1^,copyput);
  delay(1);
  until x2>getmaxx-150;
  ENTER;freemem(bola1,ukuran);
end;
PROCEDURE DEKOR;
begin
  x:=0;
  repeat
    Rectangle(10+x,10+x,getmaxx-10-x,getmaxy-10-x);
    setcolor(5+x);
    x:=x+5; until x=25;
end;
PROCEDURE COVER;
Var
  readchard,jawaban,salah:char;
  X1,Y1,Y2 :integer;
begin
  clearviewport;
  DEKOR;
  X1:=getmaxx div 2;
  Y1:=getmaxy div 5 ;
  Y2:=getmaxy div 4 ;
  Setbkcolor(lightgray);
  setfillstyle(10,11);
  floodfill(50,50,red);
  setcolor(blue); settextstyle(5,0,3);
  settextjustify(center,center);
  DEKOR;
  rectangle(x1-110,y2+30,x1+160,y2+260);
  setcolor(5); settextstyle(3,0,2);
  outtextxy(X1+20,Y2-60,'MENU SIMULASI');
  {bk pressed enter dan choose the number}
  outtextxy(X1-100,Y2+90,['1] Momentum 1');
  outtextxy(X1-100,Y2+120,['2] Momentum 2');
  outtextxy(X1-100,Y2+150,['3] Momentum 3');
  outtextxy(X1-100,Y2+240,['4] Selesai disini...!!');
  setcolor(magenta);settextstyle(0,0,1);
  outtextxy(X1-100,Y2+308,'Press ENTER to CONTINUE...!!');
  outtextxy(X1-100,Y2-10,'Nomor Pilihan anda...!!');readln(pilih);
End;
{program
utama_____}

```