```
    bi[i] := bi[i] *hrfact2;  //sesuai paper
   end;

 ti[p] := ti[p]*hrfact2;
 ti[q] := ti[q]*hrfact;
 ti[r] := ti[r];
 ti[s] := ti[s]*hrfact;
 ti[t] := ti[t]*hrfact2;

 RK4;
end;
```

8. Generate the angular velocity of motion aouud the limit cycle in 3D state space. Suppose that T(t) represents the time series generated by RR-process

$$\omega(t) = \frac{2\pi}{T(t)}.$$

```
//beberapa fungsi perlu di built sebelum masuk ke 3D state space//
//----fungsi untuk angular velocity----//
function TForm1.angfreq(in1 : real):real;
var
 i: integer;
begin
 i := 1+ floor(in1/dt);
 angfreq := 2.0*pi/norm(rr[i]);
end;

function TForm1.norm(in1 : real):real;
begin
 if (in1 = 0 ) then norm := 0.00000000000000000001
```

```
 else norm := in1;
end;
//----fungsi modulus untuk dimasukkan ke rumus z dot-----//
function TForm1.modulus(in1 :real;in2 : real):real;
begin
 while(in1 >= in2) do in1 := in1 -in2;
 modulus := in1;
end;
```

9. Generate the dynamical of x, y, and z with the equation :

$$\dot{x} = \alpha x - \omega y$$

$$\dot{y} = \alpha y + \omega x$$

$$\dot{z} = -\sum_{i \in \{P,Q,R,S,T^-,T^+\}} a_i \Delta\theta_i \exp(-\Delta\theta_i^2/2b_i^2) - (z - z_0)$$

$$\text{where } \alpha = 1 - \sqrt{x^2 + y^2},\ \Delta\theta_i = (\theta - \theta_i) \bmod 2\pi,\ \theta = \text{atan2}(y, x)$$

```
//----fungsi untuk rumus ODE 3d state----//
function TForm1.ddt(t0 : real;x0 : real;y0 : real;z0 : real;trig : integer) : real;
var
 a0,te,det,det2,temp,zbase : real;
 i : integer;
begin
 a0 := 1.0 - sqrt((x0*x0) + (y0*y0)); //rumus di paper untuk alpha
 if(trig = 1) then //trig=1 ==> rumus untuk x dot
   begin
    ddt := a0*x0 - angfreq(t0)*y0;
   end

 else if (trig = 2) then //trig=2 ==> rumus untuk y dot
   begin
```

```
    ddt := a0*y0 + angfreq(t0)*x0;
  end

 else    //trig=3 ==> rumus untuk z dot
  begin
    temp := 0;
    zbase  := 0.005*sin(2*pi*t0); //baseline wander
    te := arctan2(y0,x0);  //te = teta
    for i:= 1 to 5 do
      begin
        det := modulus(te-ti[i],2*pi); //det = delta teta
        det2 := det*det;
        temp := temp + -ai[i]*det*exp(-0.5*det2/(bi[i]*bi[i]));
      end;
    temp := temp -1.0*(z0 - zbase);
    ddt := temp;
  end;
end;
```

10. Integrating Step (9) using 4th order Runge Kutta:

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + \frac{h}{2}, \quad y_i + \frac{1}{2}k_1 h)$$

$$k_3 = f(x_i + \frac{h}{2}, \quad y_{\mathrm{i}} + \frac{1}{2}k_2 h)$$

$$k_4 = f(x_i + h, \quad y_{\mathrm{i}} + k_3 h)$$

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

```
procedure TForm1.RK4;
var
  timev : real;
  i : integer;
  k1x,k1y,k1z : real;
  k2x,k2y,k2z : real;
  k3x,k3y,k3z : real;
  k4x,k4y,k4z : real;
begin
  {Initial Condition}
  x[1] := 0.1; //x
  x[2] := 0.0; //y
  x[3] := 0.04; //z

  timev := 0.0;
  for i:= 1 to round(Nrr) do
  begin
    xt[i] := x[1];
    yt[i] := x[2];
    zt[i] := x[3];

    {konstanta 1}
    k1x := ddt(timev,x[1],x[2],x[3],1);
    k1y := ddt(timev,x[1],x[2],x[3],2);
    k1z := ddt(timev,x[1],x[2],x[3],3);

    {konstanta 2}
    k2x := ddt(timev + dt*0.5,x[1] + dt*0.5*k1x,x[2] + dt*0.5*k1y,x[3] + dt*0.5*k1z,1);
    k2y := ddt(timev + dt*0.5,x[1] + dt*0.5*k1x,x[2] + dt*0.5*k1y,x[3] + dt*0.5*k1z,2);
```

```
    k2z := ddt(timev + dt*0.5,x[1] + dt*0.5*k1x,x[2] + dt*0.5*k1y,x[3] + dt*0.5*k1z,3);

    {kosntanta 3}
    k3x := ddt(timev + dt*0.5,x[1] + dt*0.5*k2x,x[2] + dt*0.5*k2y,x[3] + dt*0.5*k1z,1);
    k3y := ddt(timev + dt*0.5,x[1] + dt*0.5*k2x,x[2] + dt*0.5*k2y,x[3] + dt*0.5*k1z,2);
    k3z := ddt(timev + dt*0.5,x[1] + dt*0.5*k2x,x[2] + dt*0.5*k2y,x[3] + dt*0.5*k1z,3);

    {konstanta 4}
    k4x := ddt(timev + dt,x[1] + k3x*dt,x[2] + k3y*dt,x[3] + k3z*dt,1);
    k4y := ddt(timev + dt,x[1] + k3x*dt,x[2] + k3y*dt,x[3] + k3z*dt,2);
    k4z := ddt(timev + dt,x[1] + k3x*dt,x[2] + k3y*dt,x[3] + k3z*dt,3);

    {result} //Rumus hasil runge kutta orde 4
    x[1] := x[1] + (dt/6)*(k1x + 2*k2x + 2*k3x + k4x);
    x[2] := x[2] + (dt/6)*(k1y + 2*k2y + 2*k3y + k4y);
    x[3] := x[3] + (dt/6)*(k1z + 2*k2z + 2*k3z + k4z);

    //if (timev < 3) then RRFreqSeries.AddXY(timev,x[3]);;
    //ecgseries.AddXY(timev,x[3]);
    timev := timev + dt;
    Series7.AddXY(i/fecg,x[3]);
    //IsoSurfaceSeries1.AddXYZ(i/fecg,x[2],x[3]);
    //Series9.AddXYZ(i,i,x[3]);
  end;

end;
```