Bab 6

Solusi Persamaan Diferensial Biasa

Persamaan diferensial adalah gabungan dari fungsi yang tidak diketahui dengan turunan (diferensial)-nya.

6.1. Kelompok Persamaan Diferensial

Persamaan diferensial dapat dibagi menjadi dua kelompok besar, yaitu :

1. **Persamaan diferensial biasa** (PDB) - *Ordinary Differential Equations* (ODE)

   PDB adalah persamaan diferensial yang hanya mempunyai satu peubah bebas.

   Contoh 6.1

   (i) $\dfrac{dy}{dx} = x + y$

   (ii) $y' = x^2 + y^2$

2. **Persamaan Diferensial Parsial** (PDP) - *Partial Differential Equations* (PDE).

   PDP adalah persamaan diferensial yang mempunyai lebih dari satu peubah bebas.

   Contoh 6.2

   (i) $\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = 6xye^{x+y}$ (yang dalam hal ini, $u = g(x,y)$)

   (ii) $\dfrac{\partial u}{\partial t} = 3\sin(x + t) + \dfrac{\partial^2 u}{\partial x^2} + (1 + x^2)\dfrac{\partial^2 u}{\partial y^2}$ (yang dalam hal ini, $u = g(x,y,t)$)

6.2 PDB Orde Satu

Bentuk baku PDB orde satu dengan nilai awal ditulis sebagai

$$y' = f(x,y)$$

dengan nilai awal $y(x_0) = y_0$ (P.6.2)

Contoh 6.3

(i) $2y' + xy = 100$; $y(0) = 1$

Bentuk baku: $y' = (100 - xy)/2$; $y(0) = 1$

(ii) $- xy' + 2y/x = y' - y$; $y(1) = -1$

Bentuk baku: $y' = \dfrac{2y/x + y}{1 + x}$; $y(1) = -1$

Penyelesaian PDB secara numerik berarti menghitung nilai fungsi di $x_{r+1} = x_r + h$, dengan h adalah **ukuran langkah** *(step)* setiap iterasi. Pada metode analitik, nilai awal berfungsi untuk memperoleh solusi yang unik, sedangkan pada metode numerik nilai awal *(initial value)* pada persamaan (P.6.2) berfungsi untuk memulai iterasi.

6.3. Metode Euler

Diberikan PDB orde satu,

$$y' = dy/dx = f(x,y) \text{ dan } y(x_0) = y_0$$

Misalkan

$$y_r = y(x_r)$$

adalah hampiran nilai y di x_r yang dihitung dengan metode Euler. Dalam hal ini

$$x_r = x_0 + rh, \qquad r = 0,1,2,...n,$$

Metoda Euler diturunkan dengan menguraikan $y(x_{r+1})$ di sekitar x_r ke dalam deret Taylor :

$$y(x_{r+1}) = y(x_r) + \frac{(x_{r+1} - x_r)}{1!} y'(x_r) + \frac{(x_{r+1} - x_r)^2}{2!} y''(x_r) + \ldots \qquad \text{(P.6.3)}$$

Bila persamaan (P.6.3) dipotong sampai suku orde tiga, diperoleh

$$y(x_{r+1}) \approx y(x_r) + \frac{(x_{r+1} - x_r)}{1!} y'(x_r) + \frac{(x_{1r+1} - x_r)^2}{2!} y''(t) \quad , x_r < t < x_{r+1} \qquad \text{(P.6.4)}$$

Berdasarkan persamaan (P.6.2),

$$y'(x_r) = f(x_r, y_r)$$

dan

$$x_{r+1} - x_r = h$$

maka persamaan (P.6.4) dapat ditulis menjadi

$$y(x_{r+1}) \approx y(x_r) + hf(x_r, y_r) + \frac{h^2}{2} y''(t) \qquad \text{(P.6.5)}$$

Dua suku pertama persamaan (P.6.5), yaitu

$$\boxed{y(x_{r+1}) \approx y(x_r) + hf(x_r, y_r)} \quad ; \quad r = 0,1,2,\ldots,n \qquad \text{(P.6.6)}$$

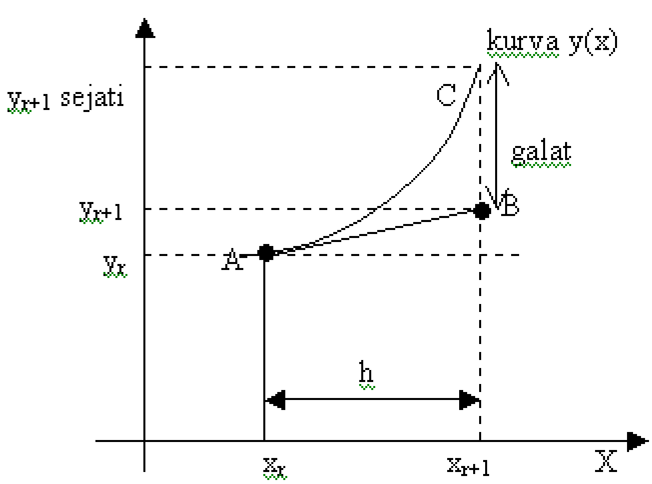
menyatakan persamaan **metode Euler** atau **metode Euler-Cauchy**. Metode Euler disebut juga **metode orde-pertama**, karena pada persamaan (P.6.5) kita hanya mengambil sampai suku orde pertama saja.

Untuk menyederhanakan penulisan, persamaan (P.6.6) dapat juga ditulis lebih singkat sebagai

$$y_{r+1} \approx y_r + hf_r$$

6.3.1. Tafsiran Geometri Metode PDB

Tinjau Gambar 6.2 di bawah ini:



Gambar 6.2: Tafsiran geometri untuk penurunan metode Euler

Dari Gambar 6.2, gradien (m) garis singgung di x_r adalah

$$m = y'(x_r) = f(x_r, y_r) = \frac{\Delta y}{\Delta x} = \frac{BC}{AB} = \frac{y_{r+1} - y_r}{h}$$

$$\Leftrightarrow y_{r+1} = y_r + hf(x_r, y_r)$$

yang tidak lain adalah persamaan metode Euler.

6.3.2. Analisis Galat Metode Euler

Meskipun metode Euler sederhana, tetapi ia mengandung dua macam galat, *yaitu* **galat pemotongan** (*truncation error*) dan **galat longgokan** (*cumulative error*). Galat pemotongan dapat langsung ditentukan dari persamaan (P.6.5), yaitu

$$E_p \approx \frac{1}{2} h^2\, y''(t) = O(h^2) \qquad \text{(P.6.7)}$$

Galat pemotongan ini sebanding dengan kuadrat ukuran langkah h sehingga disebut juga **galat per langkah** *(error per step)* atau **galat lokal**. Semakin kecil h (yang berarti semakin banyak langkah perhitungan), semakin kecil pula galat hasil perhitungannya. Perhatikan bahwa nilai pada setiap langkah (y_r) dipakai lagi pada langkah berikutnya (y_{r+1}). Galat solusi pada langkah ke-r adalah tumpukan galat dari langkah-langkah sebelumnya. Galat yang terkumpul pada akhir langkah ke-r ini disebut **galat longgokan** *(cumulative error)*. Jika langkah dimulai dari $x_0 = a$ dan berakhir di $x_n = b$ maka total galat yang terkumpul pada solusi akhir (y_n) adalah

$$E_{tot} = \sum_{r=1}^{n} (1/2)h^2 y''(t) = n\frac{h^2}{2} y''(t) = \frac{(b-a)}{2h} h^2 y''(t) = \frac{(b-a)}{2} y''(t)\, h \qquad \text{(P.6.8)}$$

$$= O(h)$$

Galat longgokan total ini sebenarnya adalah

$$E_{tot} = y(b)_{eksak} - y(x_n)_{Euler} \qquad \text{(P.6.8a)}$$

Program 6.1: **Metode Euler**

```
function y_Euler (x0, y0, b, h:real) :real;
{menghitung y(b) pada PDB
       y' = f(x,y) ;    y(x0)=y0
  dengan metode Euler
}
var
    r, n: integer;
    x, y: real;
begin
    n:=(b-x0)/h;    {jumlah langkah}
    y:=y0;          {nilai awal}
    x:=x0;
    for r:=1 to n do begin
       y:=y + h*f(x,y);          { hitung solusi y[xr] }
       x:=x + h;                 { hitung titik berikutnya }
    end; {for}
    y_Euler:=y;   {y(b)}
end;
```

6.4. Metode Heun (Perbaikan Metoda Euler)

Metode Euler mempunyai ketelitian yang rendah karena galatnya besar (sebanding dengan h). Buruknya galat ini dapat dikurangi dengan menggunakan metode Heun, yang merupakan perbaikan metode Euler (*modified Euler's method*). Pada metode Heun, solusi dari metode Euler dijadikan sebagai solusi perkiraan awal (*predictor*). Selanjutnya, solusi perkiraan awal ini diperbaiki dengan metode Heun (*corrector*).

Persamaan

$$y_{r+1} = y_r + {}^{h}/_{2}\,[f(x_r,y_r) + f(x_{r+1}, y_{r+1})] \qquad \text{(P.6.11)}$$

yang merupakan **metode Heun**, atau **metode Euler-Cauchy yang diperbaiki**. Dalam persamaan (P.6.11), suku ruas kanan mengandung y_{r+1}. Nilai y_{r+1} ini adalah solusi perkiraan awal (*predictor*) yang dihitung dengan metode Euler. Karena itu, persamaan (P.6.11) dapat ditulis sebagai

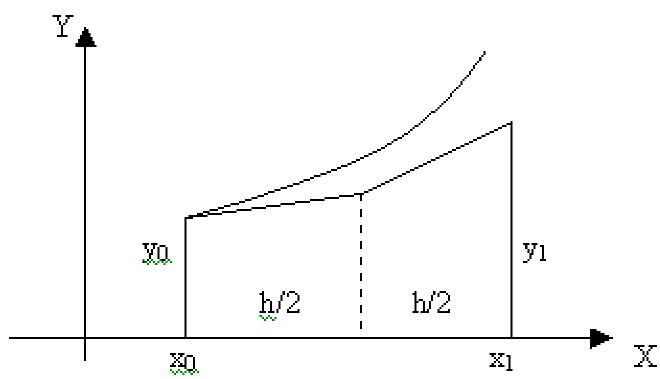
Predictor : $y^{(0)}_{r+1} = y_r + hf(x_r, y_r)$

Corrector : $y_{r+1} = y_r + {}^h/_2\,[f(x_r,y_r) + f(x_{r+1}, y^{(0)}_{r+1})]$ (P.6.12)

atau ditulis dalam satu kesatuan,

$$y_{r+1} = y_r + {}^h/_2\,[f(x_r,y_r) + f(x_{r+1}, y_r + hf(x_r,y_r)] \qquad (P.6.13)$$

6.4.1. Tafsiran Geometri Metode Heun



6.4.2. Galat Metode Heun

Dari persamaan (P.6.12), suku $h/2\,[f(x_r,y_r) + f(x_{r+1}, y^{(0)}_{r+1})]$ bersesuaian dengan aturan trapesium pada integrasi numerik. Dapat dibuktikan bahwa galat per langkah metode Heun sama dengan galat aturan trapesium, yaitu

$$E_p \approx \frac{-h^3}{12} y''(t) \;,\quad x_r < t < x_{r+1} \qquad (P.6.13)$$

$$= O(h^3)$$

Galat per langkah = nilai sejati - nilai hampiran

$$= Y_{r+1} - y_{r+1}$$

$$= (y_r + hf_r + \frac{h^2 f_r'}{2} + \frac{h^3 f_r'''}{6} + \ldots) - (y_r + hf_r + \frac{h^2 f_r'}{2} + \frac{h^3 f_r''}{4} + \ldots)$$

$$= \frac{h^3 f_r'''}{6} - \frac{h^3 f_r'''}{4} + \ldots$$

$$= \frac{-h^3 f_r'''}{12} + \ldots$$

$$= \frac{h^3 f_r'''(t)}{12}\ , \quad x_r < t < x_{r+1} \qquad \text{(terbukti)}$$

$$= O(h^3)$$

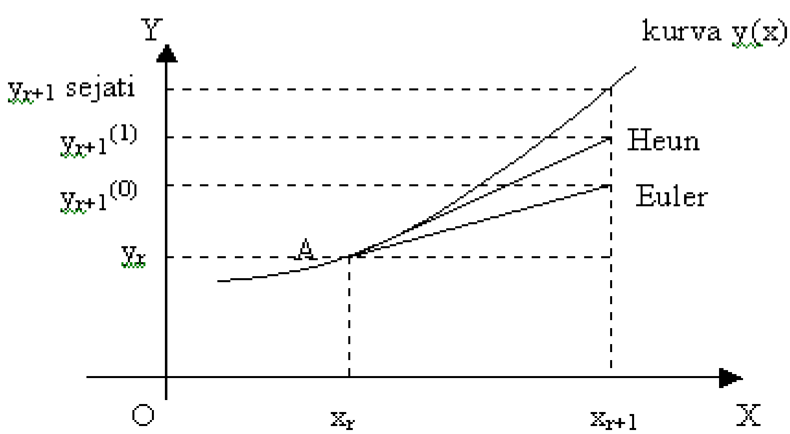
Galat longgokannya adalah,

$$E_L = \sum_{r=1}^{n} {}^{-1}/_{12}\, h^3\, y''(t)$$

$$= -\frac{(b - a)}{12} h^2\, y''(t) \qquad \text{(P.6.16)}$$

$$= O(h^2)$$

Jadi, galat longgokan metode Heun sebanding dengan h^2.



Gambar 6.5: Perbandingan metode Euler dengan metode Heun

Program 6.2: Metode Heun

```
function y_Heun (x0, y0, b, h:real) :real;
{menghitung y(b) dengan metode Heun pada PDB
       y' = f(x,y);      y(x0)=y0
   }
var
     r, n: integer;
     x, y, y_s : real;
begin
     n:=(b-x0)/h;  {jumlah langkah}
     y:=y0;        {nilai awal}
     x:=x0;
     for r:=1 to n do begin
         y_s:=y;                     { y dari langkah r-1 }
         y:=y + h*f(x,y);            { y(xr) dengan Euler }
         y:=y_s + h/2 *(f(x,y_s) + f(x+h,y));
                                      { y(xr) dengan Heun }
         x:=x+1;                      {titik berikutnya}
     end;
     y_Heun:=y;
end;
```

6.5 Metode Runge-Kutta

Metode Runge-Kutta adalah alternatif lain dari metode deret Taylor yang tidak membutuhkan perhitungan turunan. Metode ini berusaha mendapatkan derajat ketelitian yang lebih tinggi, dan sekaligus menghindarkan keperluan mencari turunan yang lebih tinggi dengan jalan mengevaluasi fungsi *f(x, y)* pada titik terpilih dalam setiap selang langkah. Metode Runge-Kutta adalah metode PDB yang paling populer karena banyak dipakai dalam praktek.

Bentuk umum metoda Runge-Kutta orde-*n* ialah:

$$y_{r+1} = y_r + a_1k_1 + a_2k_2 + \ldots + a_nk_n \qquad \text{(P.6.17)}$$

dengan $a_1, a_2, \ldots, a_n$ adalah tetapan, dan

$k_1 = hf(x_r, y_r)$

$k_2 = hf(x_r + p_1h, y_r + q_{11}k_1)$

$k_3 = hf(x_r + p_2h, y_r + q_{21}k_1 + q_{22}k_2)$

...

$$k_n = hf(x_r + p_{n-1}h,\ y_r + q_{n-1,1}k_1 + q_{n-1,2}k_2 + \ldots + q_{n-1,n-1}\ k_{n-1})$$

6.5.1. Metode Runge-Kutta Orde Satu

Metode Runge-Kutta yang pertama berbentuk

$$k_1 = hf(x_r\ ,\ y_r)$$

$$y_{r+1} = y_r + (a_1k_1) \qquad \text{(P.6.18)}$$

Galat per langkah metode R-K orde satu adalah $O(h^2)$

Galat longgokan metode R-K orde satu adalah $O(h)$.

Yang termasuk ke dalam metode R-K orde satu ialah metode Euler:

$$k_1 = hf(x_r,\ y_r)$$

$$y_{r+1} = y_r + k_1 \qquad \text{(dalam hal ini } a_1 = 1)$$

6.5.2. Metode Runge-Kutta Orde Dua

Metode Runge-Kutta yang pertama berbentuk

$$k_1 = hf(x_r,\ y_r)$$

$$k_2 = hf\ (x_r + p_1h,\ y_r + q_{11}k_1)$$

$$y_{r+1} = y_r + (a_1k_1 + a_2k_2) \qquad \text{(P.6.19)}$$

Galat per langkah metode R-K orde dua adalah $O(h^3)$.

Galat longgokan metode R-K orde dua adalah $O(h^2)$.

Nilai a_1, a_2, p_1, dan q_{11} ditentukan sebagai berikut:

Misalkan

$$f_r = f(x_r,y_r)$$

$$f_x = \frac{\partial f}{\partial x}(x_r\ , y_r),\ \text{dan}\ f_y = \frac{\partial f}{\partial y}(x_r\ , y_r)$$

Contoh metode Runge-Kutta orde dua adalah metode Heun, yang dalam hal ini

$a_2 = 1/2$,

$a_1 = 1/2$,

$p_1 = q_{11} = 1$

Dalam bentuk Runge-Kutta orde 2, metode Heun dapat ditulis sebagai

$k_1 = hf(x_r, y_r)$

$k_2 = hf(x_r + h, y_r + k_1)$

$y_{r+1} = y_r + ½ (k_1 + k_2)$

Program 6.4: Metode Heun dalam bentuk Runge-Kutta orde 2

```
function y_Heun(x0, y0, b, h:real) :real;
{menghitung y(b) dengan metode Heun pada PDB
       y' =f(x,y);    y(x0)=y0
  }
var
     r, n: integer;
     x, y, y_s, x_s : real;
begin
     n:=(b - x0)/h;   {jumlah langkah}
     y:=y0;           {nilai awal}
     x:=x0;
     for r:=1 to n do begin
        k1:=h*f(x, y);
        k2:=h*f(x+h, y+k1);
        y:=y + (k1 + k2)/2;
        x:=x+h;
     end;
     y_Heun:=y;
end;
```

6.5.3. Metoda Runge-Kutta Orde Tiga

Metode Runge-Kutta yang terkenal dan banyak dipakai adalah metode Runge- Kutta orde tiga dan metode Runge-Kutta orde empat. Metode Runge-Kutta orde tiga berbentuk:

$$k_1 = hf(x_r, y_r)$$

$$k_2 = hf(x_r + ½ h, y_r + ½ k_1)$$

$$k_3 = hf(x_r + h, y_r - k_1 + 2k_2)$$

$$y_{r+1} = y_r + {}^1/_6 (k_1 + 4k_2 + k_3) \qquad \text{(P.6.25)}$$

Galat per langkah metode R-K orde tiga adalah $O(h^4)$.

Galat longgokan metode R-K orde tiga adalah $O(h^3)$.

Program 6.5: Metode Runge_Kutta_Orde_3

```
function y_RK3(x0, y0, b, h:real):real;
{menghitung y(b) dengan metode Runge-Kutta orde tiga pada PDB
       y'=f(x,y);    y(x0)=y0
}
var
       r, n: integer;
       x, y, k1, k2, k3: real;
begin
    n:=(b - x0)/h;      {jumlah langkah}
    y:=y0;              {nilai awal}
    x:=x0;
    for r:=1 to n do begin
        k1:=h*f(x, y);
        k2:=h*f(x + h/2, y + k1/2);
        k3:=h*f(x + h, y - k1 + 2*k2);
        y:=y + (k1 + 4*k2 + k3)/6                    { nilai y(xr) }
        x:=x+h;                                      { titik berikutnya}
    end;
    y_RK3:=y;
end;
```

6.5.4. Metode Runge-Kutta Orde Empat

Metode Runge-Kutta orde empat adalah

$$k_1 = hf(x_r, y_r)$$

$$k_2 = hf(x_r + ½ h, y_r + ½ k_1)$$

$$k_3 = hf(x_r + ½ h, yr + ½ k_2)$$

$$k_4 = hf(x_r + h, y_r + k_3)$$

$$y_{r+1} = y_r + {}^1/_6 (k_1 + 2k_2 + 2k_3 + k_4) \qquad (P.6.26)$$

Galat per langkah metode R-K orde empat adalah $O(h^3)$.

Galat longgokan metode R-K orde empat adalah $O(h^2)$.

Program 6.6: Metode Runge_Kutta_Orde_4

```
function y_RK4(x0, y0, b, h:real):real;
{menghitung y(b) dengan metode Runge-Kutta orde empat pada PDB
     y' = f(x,y);     y(x0)=y0
}
var
     r, n: integer;
     x, y, k1, k2, k3, k4: real;
begin
   n:=(b - x0)/h;     {jumlah langkah}
   y:=y0;             {nilai awal}
   x:=x0;
   for r:=1 to n do begin
      k1:=h*f(x, y);
      k2:=h*f(x + h/2, y + k1/2);
      k3:=h*f(x + h/2, y + k2/2);
      k4:=h*f(x + h, y + k3);
      y:=y + (k1 + 2*k2 + 2*k3 + k4)/6    { nilai y(xr) }
      x:=x+h;                             { titik berikutnya}
   end;
   y_RK4:=y;
end;
```