BAB II
TINJAUAN PUSTAKA

2.1 Persamaan Diferensial Biasa

Persamaan diferensial adalah persamaan matematika untuk fungsi satu variabel atau lebih yang menghubungkan nilai fungsi itu sendiri dan turunannya dalam berbagai orde. Persamaan diferensial biasa (PDB) merupakan persamaan diferensial dari suatu fungsi yang bergantung pada satu variabel bebas. Orde dari suatu persamaan diferensial merupakan turunan tertinggi pada persamaan tersebut. Bentuk umum dari PDB orde-n dapat dituliskan sebagai berikut:

$$F\left(t, u(t), u'(t), \ldots, u^{(n)}(t)\right) = 0,$$

dimana F adalah PDB yang diberikan dan

$u'(t) = \frac{du}{dt}$,

$u^{(n)}(t) = \frac{d^{(n)}u}{dt^{(n)}}$.

(Boyce & Di Prima, 2001)

2.2 Persamaan Diferensial Parsial

Persamaan diferensial parsial (PDP) merupakan persamaan diferensial dari suatu fungsi yang memiliki lebih dari satu variabel bebas serta melibatkan turunan parsial. Misalkan u merupakan fungsi dari variabel bebas yang dinotasikan oleh $u = u(x_1, x_2, x_3, \ldots, x_n)$ maka secara umum bentuk PDP yaitu

$$F\left(x_1, \ldots, x_n, u, u_{x_1}, \ldots, u_{x_n}, u_{x_1 x_2}, \ldots, u_{x_i x_j}, \ldots\right) = 0,$$

dimana F adalah PDP yang diberikan dan

$$u_{x_j} = \frac{\partial u}{\partial x_j} = \partial_{x_j} u,$$

$$u_{x_i x_j} = \frac{\partial^2 u}{\partial x_i \partial x_j} = \partial_{x_i x_j} u,$$

$i, j = 1,2, \ldots n$

(Strauss, 2008)

2.3 Aproksimasi Turunan Pertama

Dengan asumsi $f(x)$ terdiferensial untuk setiap nilai x, turunan pertama dari $f(x)$ dapat diaproksimasi menggunakan formula beda maju (*forward difference*), beda mundur (*backward difference*), dan beda pusat (*centerd difference*) yang dapat dituliskan sebagai berikut:

1. Formula beda maju:
$$f'(x) \approx \frac{f(x+h)-f(x)}{h}.$$
2. Formula beda mundur:
$$f'(x) \approx \frac{f(x)-f(x-h)}{h}.$$
3. Formula beda pusat:
$$f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}.$$

(Hernadi, 2015)

2.4 Aproksimasi Turunan Parsial Pertama

Diberikan suatu fungsi $u(x,y)$ yang bergantung pada dua variabel bebas yaitu x dan y. Turunan parsial $\frac{\partial u}{\partial x}$ dapat didekati dengan ketiga formula berikut:

1. Formula beda maju:
$$\frac{\partial u}{\partial x} \approx \frac{u(x+h,y)-u(x,y)}{h}.$$
2. Formula beda mundur:
$$\frac{\partial u}{\partial x} \approx \frac{u(x,y)-u(x-h,y)}{h}.$$
3. Formula beda pusat
$$\frac{\partial u}{\partial x} \approx \frac{u(x+h,y)-u(x-h,y)}{2h}.$$

Pendekatan untuk turunan parsial $\frac{\partial u}{\partial y}$ dapat menggunakan formula berikut:

1. Formula beda maju:
$$\frac{\partial u}{\partial y} \approx \frac{u(x,y+h)-u(x,y)}{h}.$$
2. Formula beda mundur:
$$\frac{\partial u}{\partial y} \approx \frac{u(x,y)-u(x,y-h)}{h}.$$

3. Formula beda pusat

$$\frac{\partial u}{\partial y} \approx \frac{u(x,y+h)-u(x,y-h)}{2h}.$$

(Argo, 2010)

2.5 Gradien

Misalkan terdapat suatu fungsi skalar $f(x,y)$ terdefinisi dan terdiferensial pada setiap titik (x,y) di $\boldsymbol{R}^2$. Gradien dari f dinotasikan sebagai fungsi vektor $grad\ f$ atau ∇f (nabla f). Operator ∇ (nabla) didefinisikan oleh:

$$\nabla = \frac{\partial}{\partial x}\boldsymbol{i} + \frac{\partial}{\partial y}\boldsymbol{j},$$

sehingga gradien dari $f(x,y)$ didefinisikan oleh:

$$grad\ f = \nabla f = \frac{\partial f}{\partial x}\boldsymbol{i} + \frac{\partial f}{\partial y}\boldsymbol{j}.$$

Gradien dari f merepresentasikan tingkat kemiringan atau perubahan nilai fungsi f pada setiap arah di ruang vektor $\boldsymbol{R}^2$.

Magnitude atau besaran dari vektor ∇f didefinisikan sebagai berikut:

$$\|\nabla f\| = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}.$$

(Schey, 1997)

2.6 Divergensi

Misalkan vektor $\boldsymbol{V}(x,y) = V_1\boldsymbol{i} + V_2\boldsymbol{j}$ terdefinisi dan terdiferensial pada setiap titik (x,y) dalam ruang vektor $\boldsymbol{R}^2$. Divergensi dari $\boldsymbol{V}$ atau $div\ \boldsymbol{V} = \nabla.\boldsymbol{V}$, didefinisikan oleh:

$$\begin{aligned}\nabla.\boldsymbol{V} &= \left(\frac{\partial}{\partial x}\boldsymbol{i} + \frac{\partial}{\partial y}\boldsymbol{j}\right).(V_1\boldsymbol{i} + V_2\boldsymbol{j}),\\ &= \frac{\partial V_1}{\partial x} + \frac{\partial V_2}{\partial y}.\end{aligned}$$

Divergensi dari vektor $\boldsymbol{V}(x,y)$ merepresentasikan penyebaran nilai intensitas $\boldsymbol{V}$ pada titik (x,y).

(Schey, 1997)

2.7 Citra

Citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto. Citra dapat dibedakan menjadi dua yaitu citra analog dan citra digital. Citra analog adalah citra yang bersifat kontinyu, seperti gambar pada monitor televisi, foto sinar-X, foto yang tercetak di kertas foto, hasil CT scan. Citra analog tidak dapat direpresentasikan dalam komputer sehingga tidak bisa diproses secara langsung. Suatu citra digital terdiri dari sejumlah elemen yang berhingga. Masing-masing elemen mempunyai lokasi dan nilai tertentu. Elemen-elemen ini disebut sebagai *pixels*.

(Hermawati, 2013)

2.7.1 Representasi Citra Digital

Citra digital dapat diwakili oleh sebuah fungsi dua dimensi $f(x, y)$, dimana harga x (baris) dan y (kolom) adalah koordinat posisi. Harga fungsi f di setiap pasangan koordinat (x, y) disebut intensitas atau *pixel* dari citra di titik tersebut. Sebuah citra digital berukuran $M \times N$ dapat ditulis dalam bentuk matriks berikut.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & \cdots & \cdots & f(1,N-1) \\ \cdots & \cdots & \cdots & \cdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}.$$

Pada saat proses digitalisasi (sampling dan kuantisasi) diperoleh besar baris M dan kolom N hingga citra membentuk matriks $M \times N$ dan tingkat gradasi warna dalam 1 *pixel* berjumlah J.

Jumlah bit yang diperlukan untuk menyimpan J gradasi warna dalam 1 *pixel* adalah l bit dimana $J = 2^l$ dan l merupakan bilangan bulat positif. Jumlah bit yang diperlukan untuk menyimpan citra digital berukuran $M \times N$ dalam memori adalah $M \times N \times l$ bit.

(Sutoyo, 2009)

2.7.2 Jenis-jenis Citra Digital

Citra digital dibagi menjadi tiga jenis, yaitu citra biner, citra *grayscale*, dan citra warna. Citra biner adalah citra digital yang memiliki memiliki 2 gradasi warna dalam 1 *pixel* yaitu hitam dan putih. Dibutuhkan 1 bit di memori untuk menyimpan kedua warna ini. Pada citra biner, warna hitam dinyatakan dengan 0 dan warna putih dinyatakan dengan 1. Berbeda dengan citra biner, citra *grayscale* merupakan citra digital dengan 256 kemungkinan nilai *pixel*. Nilai tersebut dimulai dari nol untuk warna hitam dan 255 untuk warna putih. Dibutuhkan memori sebesar 8 bit untuk menyimpan *pixel* citra *grayscale*. Citra warna merupakan citra yang dibentuk oleh tiga warna, yaitu *Red, Green*, dan *Blue* (RGB). Citra ini direpresentasikan sebagai tiga buah matriks berbeda, yaitu matriks R menyatakan nilai *red*, matriks G menyatakan nilai *green*, dan matriks B menyatakan nilai *blue*. Setiap *pixel* pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = *Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 *byte*, yang berarti setiap warna memiliki gradasi sebanyak 256 warna. Contoh masing-masing jenis citra dapat dilihat pada Gambar 2.1.

(a) (b) (c)

Gambar 2.1 (a) Citra warna, (b) Citra *grayscale*, dan (c) Citra biner. (Putra, 2010)

2.7.3 Ketetanggaan *Pixel*

Sebuah *pixel* p pada koordinat (x, y) mempunya 4 tetangga horizontal dan vertikal dengan koordinat $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$. Himpunan *pixel-pixel* ini disebut 4-tetangga dari p, dinotasikan dengan $N_4(p)$. Ilustrasi 4-tetangga dari *pixel* p dapat dilihat pada Gambar 2.2.

	$(x-1,y)$	
$(x,y-1)$	(x,y)	$(x,y+1)$
	$(x+1,y)$	

Gambar 2.2 4-tetangga dari *pixel* p

Keempat tetangga diagonal (*diagonal neighbors*) dari p, dinotasikan $N_D(p)$. $N_D(p)$ mempunyai koordinat $(x-1,y-1)$, $(x+1,y-1)$, $(x-1,y+1)$, dan $(x+1,y+1)$. Ilustrasi 4-tetangga diagonal dari *pixel* p dapat dilihat pada Gambar 2.3.

$(x-1,y-1)$		$(x-1,y+1)$
	(x,y)	
$(x+1,y-1)$		$(x+1,y+1)$

Gambar 2.3 4-tetangga diagonal *pixel* p

$N_D(p)$ digabungkan dengan $N_4(p)$ disebut 8-tetangga dari p, dinotasikan $N_8(p)$. Ilustrasi 8-tetangga dari *pixel* p dapat dilihat pada Gambar 2.4.

$(x-1,y-1)$	$(x-1,y)$	$(x-1,y+1)$
$(x,y-1)$	(x,y)	$(x,y+1)$
$(x+1,y-1)$	$(x+1,y)$	$(x+1,y+1)$

Gambar 2.4 8-tetangga *pixel* p

(Hermawati, 2013)

2.7.4 *Noise*

Munculnya *noise* pada citra digital terjadi pada saat akuisisi citra (digitisasi) dan atau transmisi citra. Sifat *noise* pada domain spasial mengacu pada ada atau tidaknya hubungan *pixel* noise dengan *pixel*-*pixel* lain dalam citra tersebut. *Noise* biasanya berdiri sendiri dan tidak terhubung dengan *pixel* lain.

Berdasarkan bentuk dan karakteristiknya, *noise* pada citra dapat dibedakan menjadi beberapa jenis, antara lain *noise Gaussian*, *noise speckle*, dan *noise 'Salt & Pepper'*.

Noise Gaussian dibangkitkan dengan cara membangkitkan bilangan acak berdistribusi Gauss dengan suatu nilai ragam (σ^2) dan rata-rata (μ). Pada skripsi ini digunakan *noise Gaussian* dengan σ^2 bernilai positif dan $\mu = 0$. Kemudian untuk *pixel* yang terkena *noise*, nilai citra ditambahkan dengan nilai *noise* yang ada, atau dirumuskan dengan:

$$F(x,y) = f(x,y) + p(x,y).\sigma + \mu,$$

dimana:

$p(x,y)$: bilangan acak berdistribusi normal baku,
$F(x,y)$: citra ber-*noise*, dan
$f(x,y)$: nilai citra sebelum terkena *noise.*

Noise speckle dibangkitkan dengan cara membangkitkan bilangan acak berdistribusi Gauss dengan nilai rata-rata 0 dan suatu nilai ragam *noise* yang bernilai positif. Citra yang terkena *noise* '*Speckle'* dirumuskan dengan:

$$F(x,y) = f(x,y) + f(x,y).\sigma.p(x,y),$$

dimana:

$F(x,y)$: citra ber-*noise*,
$p(x,y)$: probabilitas acak di titik (x,y) berdistribusi Gauss.

Noise salt & pepper memberikan tampilan titik terang (*salt*) dan titik gelap (*pepper*) pada citra. Intensitas *noise* bernilai 0 untuk noise *pepper* dan 255 untuk noise *salt*. Citra yang terkena *noise 'Salt & Pepper'* dapat dirumuskan dengan:

$$F(x,y) = \begin{cases} 0, & a(x,y) < d/2 \\ 255, & d/2 \leq a(x,y) < d \\ f(x,y), & a(x,y) \geq d \end{cases},$$

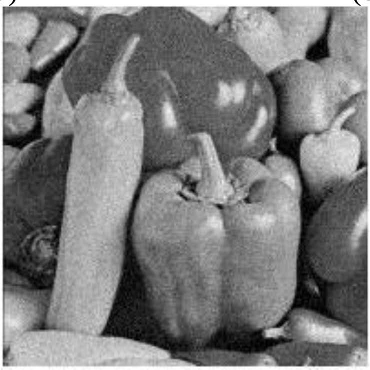
dimana:

$F(x,y)$: citra ber-*noise*,
$a(x,y)$: probabilitas acak di titik (x,y) berdistribusi seragam,
d : persentase *noise* '*Salt & Pepper'*,

Contoh citra yang terkena *noise* dapat dilihat pada Gambar 2.5.

(a) (b)



(c)

Gambar 2.5 (a) Citra ber-*noise' salt & pepper'*, (b) Citra ber-*noise speckle*, dan (c) Citra ber-noise *Gaussian*

(Makandar, dkk., 2012)

2.8 Konvolusi Citra

Konvolusi pada citra biasa disebut konvolusi dua dimensi (konvolusi 2D). Konvolusi 2D didefinisikan sebagai proses untuk memperoleh suatu *pixel* berdasarkan nilai *pixel* itu sendiri dan tetangganya. Operasi konvolusi pada suatu citra $f(x, y)$ didefinisikan sebagai berikut:

$$h(x, y) = f(x, y) * g(x, y) = \sum_{a=x-1}^{x+1} \sum_{b=y-1}^{y+1} f(a, b)g(x - a, y - b),$$

dimana $h(x,y)$ merupakan citra hasil konvolusi dan fungsi $g(x,y)$ merupakan filter konvolusi. Filter konvolusi pada citra digital umumnya berupa matriks 3×3 dengan $g(0,0)$ sebagai elemen tengah atau kernel dari filter $g(x,y)$. Bentuk filter g(x,y) adalah sebagai berikut:

$$g(x,y) = \begin{bmatrix} g(-1,-1) & g(-1,0) & g(-1,1) \\ g(0,-1) & g(0,0) & g(0,1) \\ g(1,-1) & g(1,0) & g(1,1) \end{bmatrix}, \qquad ,$$

misalkan dilakukan konvolusi pada citra $f(x,y)$ dengan filter $g(x,y)$. Pada pelaksanaan konvolusi, filter digeser sepanjang baris dan kolom pada citra unuk setiap subcitra 3×3 dari $f(x,y)$ sehingga diperoleh nilai *pixel* baru pada citra keluaran $h(x,y)$. Pada citra hasil konvolusi pada koordinat (1,1) akan didpatkan nilai *pixel* dengan perhitungan sebagai berikut:

$$\begin{aligned} h(1,1) = {} & f(0,0)g(1,1) + f(0,1)g(1,0) + f(0,2)g(1,-1) \\ & + f(1,0)g(0,1) + f(1,1)g(0,0) + f(1,2)g(0,-1) \\ & + f(2,0)g(-1,1) + f(2,1)g(-1,0) + f(2,2)f(-1,-1). \end{aligned}$$

(Hermawati, 2013)

2.9 Persamaan *Anisotropic Diffusion*

Persamaan difusi merupakan persamaan diferensial parsial yang menggambarkan kepadatan bahan menjalani difusi. Sementara persamaan *anisotropic diffusion* merupakan salah satu bentuk persamaan difusi nonlinear yang terdapat unsur koefisien difusi yang tidak konstan didalamnya.

Bentuk umum persamaan *anisotropic diffusion* adalah sebagai berikut:

$$\frac{\partial u(\boldsymbol{r},t)}{\partial t} = div(D(u(\boldsymbol{r},t))\boldsymbol{\nabla} u(\boldsymbol{r},t)), \tag{2.1}$$

dimana $u(\boldsymbol{r},t)$: densitas material pada ruang $\boldsymbol{r}(x,y)$ dan waktu t,
$D(u(\boldsymbol{r},t))$: koefisien difusi untuk densitas material u di lokasi $\boldsymbol{r}$ dan waktu t.

(Yunita, 2013)

2.10 Model *Anisotropic Diffusion Filter* (ADF)

Perona & Malik (1990) memperkenalkan metode *non-linear filtering* menggunakan persamaan *anisotropic diffusion* yang bertujuan untuk mereduksi *noise* sekaligus menjaga detail objek citra pada citra yang terkorupsi *noise*. Model *Anisotropic Diffusion Filter* (ADF) didefinisikan sebagai berikut:

$$\frac{\partial \boldsymbol{I}(x,y,t)}{\partial t} = div(g.\nabla \boldsymbol{I}(x,y,t)), \quad \boldsymbol{I}(x,y,0) = \boldsymbol{I}_0 \tag{2.2}$$

dimana:

t	:	parameter iterasi,
$\boldsymbol{I}(x,y,t)$	:	citra pada waktu t,
$\boldsymbol{I}_0$	:	citra awal pada $t = 0$,
div	:	operator divergensi,
$\boldsymbol{\nabla I}$	:	gradien citra,
g	:	fungsi koefisien difusi.

Fungsi koefisien difusi ditentukan sebagai fungsi positif yang monoton turun dan memenuhi kondisi:

$$\lim_{x\to 0} g(x) = 1, \quad \lim_{x\to\infty} g(x) = 0 \tag{2.3}$$

terdapat dua opsi fungsi koefisien difusi pada model ADF yang didefinisikan pada persamaan (2.4) dan (2.5).

$$g_1(\|\boldsymbol{\nabla I}\|) = \frac{1}{1+(\frac{\|\boldsymbol{\nabla I}\|}{k})^2}, \tag{2.4}$$

$$g_2(\|\boldsymbol{\nabla I}\|) = e^{\left(-\left(\frac{\|\boldsymbol{\nabla I}\|}{k}\right)^2\right)}, \tag{2.5}$$

dimana:

k: parameter gradien *threshold*,

$\|\boldsymbol{\nabla I}\|$: besaran gradien citra.

(Tebini, dkk., 2013)

2.11 Estimasi Parameter Gradien *Threshold*

Parameter gradien *threshold* (k) memiliki peranan penting dalam dalam model ADF. Parameter k menyatakan ambang antara nilai citra nilai gradien citra yang berkaitan dengan *noise* dan nilai gradien citra yang berkaitan dengan tepi objek citra.

Black, dkk., (1998) mendefinisikan nilai estimasi parameter gradien *threshold* sebagai berikut:

$k = 1.4826MAD(\nabla\mathbf{I}(\mathrm{x}, \mathrm{y})),$ (2.6)

dimana:

$\boldsymbol{I}(x, y)$**:** citra ber-*noise*,

$MAD = median(|(\|\nabla\boldsymbol{I}(x, y)\| - median(\|\nabla\boldsymbol{I}(x, y)\|)|).$

(Black, dkk., 1998)

2.12 Kriteria Evaluasi Citra

Kualitas citra hasil *denoising* dapat diukur melalui beberapa kriteria evaluasi, antara lain adalah *Mean Square Error* (MSE) dan *Peak Signal to Noise Ratio* (PSNR).

MSE menyatakan rata-rata kuadrat nilai kesalahan antara citra asli dengan citra hasil *denoising* yang secara matematis dapat dinyatakan sebagai berikut:

$$\mathrm{MSE} = \frac{1}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}([\boldsymbol{I}(x, y) - \boldsymbol{I_d}(x, y)]^2), \quad (2.7)$$

dimana:

$\boldsymbol{I}$: citra asli,

$\boldsymbol{I_d}$: citra hasil *denoising*,

$M \times N$: ukuran citra.

PSNR merupakan nilai perbandingan antara nilai maksimum intensitas citra hasil *denoising* dengan kuantitas *noise*, yang dinyatakan dalam satuan *decibel* (dB). Secara matematis PSNR dinyatakan sebagai berikut:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{{MAX_{\boldsymbol{I_d}}}^2}{MSE}\right). \quad (2.8)$$

dimana MAX_{I_d} merupakan nilai maksimum intensitas citra hasil *denoising*. Semakin baik kualitas citra hasil *denoising*, maka nilai PSNR semakin tinggi dan nila MSE semakin rendah.

(Tebini, dkk., 2013)