

BAB II LANDASAN TEORI

2.1. Citra Digital

Putra (2010, p.19) mendefinisikan “Citra digital adalah sebuah larik (*array*) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu”.

Menurut Permadi & Murinto (2015, p.1029) Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi.

Awalludin & Irmawati (2016, p.2) mendefinisikan “Citra atau *image* merupakan salah satu komponen multimedia yang memiliki peran penting sebagai informasi dalam bentuk visual”.

Menurut Muhtadan & Harsono (2008, p.468) “Pengolahan citra adalah pemrosesan citra, khususnya menggunakan komputer digital untuk menghasilkan citra manipulasi yang kualitasnya lebih baik dari sebelumnya, sehingga citra tersebut dapat diinterpretasikan baik oleh manusia maupun mesin”.

2.1.1. Citra Berwarna (*True Color*)

Mengacu pada pendapat Nurtatio dan Sutojo (2015) Setiap piksel 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 berarti mempunyai gradasi warna sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak $2^8 \cdot 2^8 \cdot 2^8 = 2^{24} = 16$ juta warna lebih.

Menurut Wibowo (2011) Dasar model warna RGB (Red Green Blue) tidak hanya mewakili warna tetapi juga intensitas cahaya. Pencahayaan yang berbeda-beda pada warna kulit orang karena pencahayaan yang ada disekitarnya (p.118).

Menurut Kumaseh, Latumakulita, & Naiggolan (2013, p.75) “Suatu citra RGB (Red, Green, Blue) terdiri dari tiga bidang citra yang saling lepas, masing masing terdiri dari warna utama, yaitu : merah, hijau dan biru di setiap pixel”.

2.1.2. Citra Keabuan (*Grayscale*)

Putra (2010, p.40) mendefinisikan “Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk menunjukkan tingkat intensitasnya”. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Citra *grayscale* berikut memiliki kedalaman warna 8 *byte* (256 kombinasi warna keabuan).

(Noor, 2011. p.14) menyatakan “Pada pengubahan sebuah gambar menjadi grayscale dapat dilakukan dengan cara mengambil semua pixel pada gambar kemudian warna tiap pixel akan diambil informasi mengenai 3 warna dasar yaitu merah, biru dan hijau (melalui fungsi `warnatoRGB`), ketiga warna dasar ini akan dijumlahkan kemudian dibagi tiga sehingga didapat nilai rata-rata. Nilai rata-rata inilah yang akan dipakai untuk memberikan warna pada pixel gambar sehingga warna menjadi grayscale”.

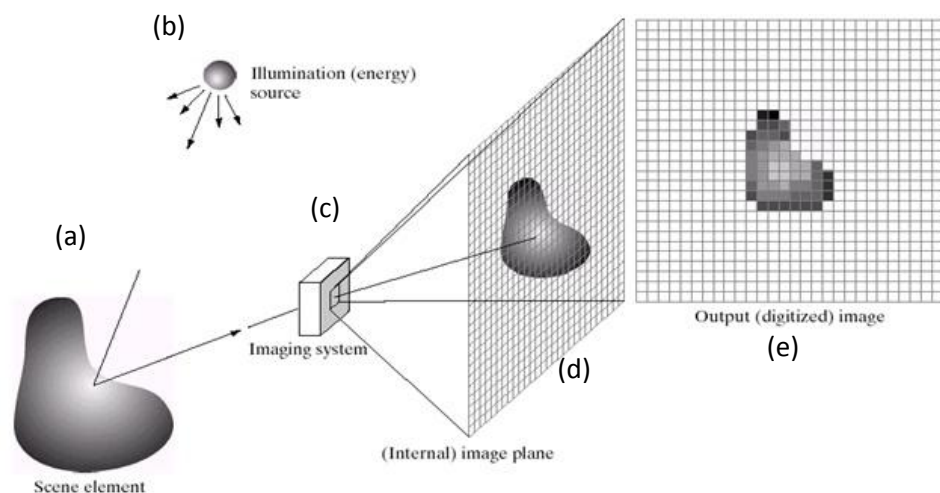
2.1.3. Citra Hitam Putih

Putra (2010, p.40) mendefinisikan “citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra biner juga disebut sebagai citra B&W (*black and white*) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel dari citra biner”.

Santi (2011, p. 102) menyatakan ”Citra monokrom atau citra hitam putih merupakan citra satu kanal, dengan citra $f(x,y)$ merupakan fungsi tingkat keabuan dari hitam ke putih: x adalah variabel baris atau garis jelajah dan y adalah variabel kolom atau posisi piksel di garis jelajah”.

2.2. Akuisisi Citra

Putra (2010, p. 28) menyatakan “proses akuisisi citra adalah pemetaan suatu pandangan (*scene*) menjadi citra kontinu dengan menggunakan sensor.



Gambar 2.1 proses akuisisi citra digital

Gambar 2.1 proses akuisisi citra digital (a) objek yang akan diambil (b) sumber cahaya (c) sistem pencitraan (*imaging*) (d) bidang citra (e) citra digital. Sumber gambar : Gonzales, Rafael C and Woods, Richard E., Digital Image Processing, Addison-Wesley publishing Company, 1993.

Gambar 2.1 adalah contoh proses akuisisi citra digital. Proses ini dapat dijelaskan sebagai berikut. Ada sebuah obyek yang akan diambil gambarnya untuk dijadikan citra digital. Sumber cahaya diperlukan untuk menerangi obyek, ini berarti ada intensitas cahaya yang diterima oleh obyek. Oleh obyek, intensitas cahaya sebagian diserap dan sebagian lagi dipantulkan ke lingkungan sekitar obyek secara radial. Sistem pencitraan menerima sebagian dari intensitas yang dipantulkan oleh obyek tadi. Di dalam sistem pencitraan terdapat sensor optik yang digunakan untuk mendeteksi intensitas cahaya yang masuk ke dalam sistem.

Keluaran dari sensor ini berupa arus yang besarnya sebanding dengan intensitas cahaya yang mengenainya. Arus tersebut kemudian di konversi menjadi data digital yang kemudian dikirimkan ke unit penampil atau unit pengolah lainnya. Secara keseluruhan hasil keluaran sistem pencitraan berupa citra digital.

Menurut Nurcahyani & Saptono (2015, p.65) “Dalam proses akuisisi data citra digital, data gambar digital dari *image* yang telah dikumpulkan untuk tahap testing diproses menjadi data tekstual”.

Ada beberapa macam sensor untuk akuisisi citra, yaitu sensor tunggal (*single sensor*), sensor garis (*sensor strip*), dan sensor larik (*sensor array*).

1) Sensor Tunggal (*single sensor*)

Sensor tunggal yang paling familiar adalah *photodiode*. *Photodiode* terbentuk dari silicon yang memiliki tegangan keluaran yang sebanding dengan cahaya. Untuk menciptakan citra 2 dimensi dengan menggunakan sensor ini, harus ada proses pemindahan relative di sumbu x dan y antara sensor dan objek.

2) Sensor Garis (*sensor strip*)

Sensor garis melakukan pencitraan satu arah. Sensor ini berupa deretan sensor yang disatukan dalam satu baris sehingga dapat melakukan akuisisi sumbu x secara bersamaan. Untuk mengakuisisi citra keseluruhan, sensor digerakkan searah sumbu y. sensor ini sering dijumpai dalam mesin *scanner*.

3) Sensor Larik (*sensor array*)

Sensor jenis ini banyak sekali ditemukan dalam kamera digital. Sensor ini berbentuk larik 2 dimensi. Sensor larik yang terdapat pada kamera digital disebut sensor CCD dengan ukuran sensor yang rata-rata mencapai 4.000 x 4.000 elemen.

2.2.1. Kamera

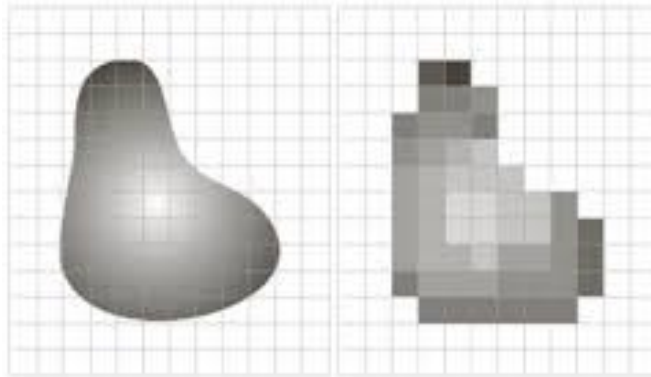
Pencitraan adalah proses untuk mentransformasi citra analog menjadi citra digital. Salah satu alat yang dapat digunakan untuk pencitraan adalah kamera.



Gambar 2.2 salah satu contoh kamera DSLR

2.2.2 Digitalisasi Citra

Pulung dan Sutojo (2015, p.15) menyatakan bahwa “citra analog tidak bisa diproses langsung oleh komputer. Agar dapat diproses oleh komputer citra analog harus terlebih dahulu diubah menjadi citra digital (pencitraan). Suatu proses yang mengubah citra analog menjadi citra digital disebut digitalisasi citra”.



Gambar 2.3 digitalisasi citra

2.2.3 Format Berkas Citra

Mengacu pada pendapat Dr. Pulung Nurtantio dan T. Sutojo (2015) Ada dua jenis format file citra yang sering digunakan dalam pengolahan citra, yaitu bitmap dan citra vektor. Istilah ini biasanya digunakan pada saat kita melakukan desain grafis.

Menurut Wiharja & Harjoko (2014, p.58) “Pengolahan citra digital adalah salah satu teknologi yang dikembangkan untuk mendapatkan informasi dari citra dengan cara memodifikasi bagian dari citra yang diperlukan sehingga menghasilkan citra lain yang lebih informatif”.

2.2.3.1 Format File Citra Bitmap

Citra bitmap sering disebut juga dengan citra *raster*. Citra bitmap menyimpan data kode citra secara digital dan lengkap (cara penyimpanannya adalah per piksel). Citra bitmap dipresentasikan dalam bentuk matrik atau dipetakan dengan menggunakan bilangan biner atau sistem bilangan lain, memiliki kelebihan untuk memanipulasi warna, namun untuk merubah obyek lebih sulit. Tampilan bitmap mampu menunjukkan kehalusan gradasi bayangan dan warna dari sebuah gambar, karena itu bitmap merupakan media elektronik yang paling tepat untuk gambar-gambar dengan perpaduan gradasi warna yang rumit seperti foto dan lukisan digital.

Tabel 2.1 Format file citra bitmap

Nama Format	Ekstensi	Kegunaan
Microsoft Windows Bitmap	BMP	Format umum untuk menyimpan citra bitmap yang dikembangkan oleh Microsoft.
CompuServe Graphics Interchange Format	GIFF	Format umum citra yang dirancang untuk keperluan transmisi melalui modem.
Aldus Tagged Image File Format	TIF	Format kompleks dan multiguna yang dikembangkan Aldus bersama Microsoft.
WordPerfect Graphics Format	WPG	Format vektor yang juga mendukung citra bitmap
GEM Image Format	IMG	Format bitmap yang dikembangkan untuk riset digital dilingkungan GEM

2.2.3.2 Format File Citra Vektor

Citra vektor dihasilkan dari perhitungan matematis dan tidak berdasarkan piksel, yaitu data tersimpan dalam bentuk vektor posisi, dimana yang tersimpan hanya informasi vektor posisinya dengan bentuk sebuah fungsi. Pada citra vektor lebih sulit dalam merubah warna tapi lebih mudah membentuk obyek dengan cara merubah nilai. Karena itu apabila citra diperbesar atau diperkecil kualitas citra relatif tetap baik dan tidak berubah. Citra vektor biasanya dibuat menggunakan aplikasi-aplikasi citra vektor seperti Corel Draw, Adobe Illustrator, Macromedia Freehand, Autocad dan lain-lain.

Format file yang citra vektor yaitu *AutoCAD Drawing Format (DWG)*, *AutoCAD Drawing Exchange Format (DXF)*, *Microstation Drawing Format (DGN)*, dan *Scalabel Vektor Graphics (SVG)*.

2.3. Noise

Mengacu pada pendapat Eko Prasetyo (2011) sumber noise pada citra digital bisa terjadi sejak pengambilan atau transmisi citra. Kinerja dari sensor citra dipengaruhi oleh banyak faktor seperti kondisi lingkungan selama pengambilan citra dan oleh kualitas sensitivitas elemen itu sendiri. Sebagai contoh, dalam pengambilan citra dengan kamera CCD, level pencahayaan dan suhu sensor adalah faktor utama yang mempengaruhi tingkat noise pada citra yang dihasilkan. Dalam proses transmisi misalnya citra yang terkorupsi.

Menurut pendapat Winarno (2011, p.44) menyatakan “Meskipun sebuah citra kaya informasi, namun seringkali citra yang kita miliki mengalami penurunan intensitas mutu, misalnya mengandung cacat atau derau (noise), warnanya terlalu kontras atau kabur tentu citra seperti ini akan sulit di representasikan sehingga informasi yang ada menjadi berkurang. Agar citra yang mengalami gangguan mudah direpresentasikan maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik”.

2.3.1. Jenis-jenis Noise

Berdasarkan pada asumsi sebelumnya, descriptor noise spasial yang akan dibahas adalah perilaku statistic nilai intensitas komponen noise dari model proses degradasi/restorasi citra. Dalam hal ini perlu diperhatikan variabel random yang dikarakteristikan oleh probability density function (PDF). Berikut ini adalah diantara PDFs yang paling umum ditemukan dalam aplikasi pengolahan citra.

1) Gaussian Noise

Model noise Gaussian (disebut juga normal) yang paling sering digunakan dalam prakteknya. Faktanya, *tractability* ini tepat dan sering menghasilkan model Gaussian yang digunakan dalam situasi dimana secara garis besar dapat diaplikasikan dengan paling baik.

PDF variabel random Gaussian z , diberikan oleh:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$$

dimana z merepresentasikan intensitas, \bar{z} adalah nilai rata-rata z dan σ adalah standar deviasi. Kuadrat dari standar deviasi disebut dengan varian dari z . fungsi plot dari fungsi ini ditunjukkan oleh gambar. Ketika z digambarkan oleh formula diatas diperkirakan 70% nilainya dalam range $[(\bar{z} - 2\sigma), (\bar{z} + 2\sigma)]$.

2) Rayleigh Noise

PDF noise Rayleigh diberikan oleh:

$$p(z) = \begin{cases} \frac{2}{b}(z - a)e^{-(z-a)^2/b} & \text{Untuk } z \geq a \\ 0 & \text{Untuk } z < a \end{cases}$$

rata-rata dan varian diberikan oleh formula:

$$\bar{z} = a + \sqrt{\pi b/4}$$

Dan

$$\sigma^2 = \frac{b(4 - \pi)}{4}$$

3) Erlang (Gamma) Noise

PDF noise Erlang diberikan oleh:

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} \\ 0 \end{cases}$$

dimana parameter $a > 0$, b integer positif dan “!” mengindikasikan factorial. Rata-rata dan varian dari kepadatan ini diberikan oleh:

$$\bar{z} = \frac{b}{a} \text{ dan } \sigma^2 = \frac{b}{a^2}$$

4) Exponential Noise

PDF exponential noise diberikan oleh:

$$p(z) = \begin{cases} ae^{-az} & \text{Untuk } z \geq 0 \\ 0 & \text{Untuk } z < 0 \end{cases}$$

dimana $a > 0$. Rata-rata dan varian dari kepadatan ini adalah:

$$\bar{z} = \frac{1}{a} \text{ dan } \sigma^2 = \frac{1}{a^2}$$

5) Uniform Noise

PDF uniform noise diberikan oleh:

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{Jika } a \leq z \leq b \\ 0 & \text{lainnya} \end{cases}$$

rata-rata dan varian dari kepadatan ini adalah:

$$\bar{z} = \frac{a+b}{2} \text{ dan } \sigma^2 = \frac{(b-a)^2}{12}$$

6) Impulse (Salt-and-Pepper) Noise

PDF noise (bipolar) *impulse* diberikan oleh:

$$p(z) = \begin{cases} Pa & \text{untuk } z = a \\ Pb & \text{untuk } z = b \\ 0 & \text{lainnya} \end{cases}$$

jika $b > a$ akan tampak sebagai titik terang pada citra. Sebaliknya, level a akan tampak seperti titik gelap. Jika selain P_a atau P_b nol, impulse noise disebut juga *unipolar*. Jika probability selain nol, dan khususnya diperkirakan sama, nilai impulse noise akan mirip butiran *salt-and-pepper* secara acak yang terdistribusi pada citra.

2.4. Pengolahan Citra Digital

Secara umum, istilah pengolahan citra menyatakan “pemrosesan gambar berdimensi-dua melalui computer digital” (Jain, 1998). “Pada umumnya tujuan dari image processing adalah mentransformasikan atau menganalisis suatu gambar sehingga informasi baru tentang gambar dibuat lebih jelas.” (Darujati dkk, 2014. p.33).

2.4.1. Perbaikan Citra (*Image Restoration*)

Kadir (2013, p.412) menyatakan “istilah restorasi mempunyai perbedaan makna dengan peningkatan citra. Restorasi citra merupakan proses untuk membuat citra yang kualitasnya turun akibat adanya tambahan derau agar menjadi mirip dengan keadaan aslinya.

2.4.1.1. Restorasi Pada Citra Degradasi yang Hanya Terkena Noise Saja

Bila degradasi citra hanya disebabkan oleh noise saja, maka persamaan berubah menjadi :

$$g(x,y) = f(x,y) + \eta(x,y)$$

fungsi noise $\eta(x,y)$ tidak diketahui, sehingga untuk mencari citra $\hat{f}(x,y)$, sebagai pendekatan dari citra semula $f(x,y)$, dengan cara mengurangi citra terdegradasi dengan fungsi noise $g(x,y) - \eta(x,y)$, sangat tidak realistis. Untuk itu diperlukan teknik-teknik tertentu yang bisa digunakan untuk mendapatkan citra pendekatan $\hat{f}(x,y)$.

2.4.1.2. Reduksi Noise Menggunakan Filter Rata-Rata

Citra yang mengandung noise, bila langsung diproses dan diekstrak fitur-fitur pentingnya akan menimbulkan masalah akurasi. Jadi sebaiknya citra tersebut harus dibersihkan dari noise terlebih dahulu, kemudian baru diproses untuk diekstrak fitur-fitur pentingnya. Pada bagian ini akan dibahas beberapa teknik untuk mereduksi noise menggunakan filter rata-rata, yaitu filter rata-rata aritmatik, filter rata-rata geometrik, filter rata-rata harmonik, dan filter rata-rata kontra harmonik.

1) Filter Rata-rata Aritmatik

Filter rata-rata aritmatik adalah metode paling mudah dari filter rata-rata. Misalkan S_{xy} mewakili himpunan koordinat dalam sebuah window berukuran $m \times n$, berpusat di titik (x,y) . Proses filter rata-rata aritmatik menghitung rata-rata nilai dari citra yang rusak $g(x,y)$ pada area yang didefinisikan oleh S_{xy} . Nilai dari citra $f(x,y)$ yang diperbaiki pada tiap titik (x,y) hanya dihitung dengan menggunakan piksel dalam daerah yang didefinisikan oleh S_{xy} . Dengan kata lain

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t)$$

operasi ini dapat diimplementasikan dengan menggunakan konvolusi.

2) Filter Rata-rata Geometri

Sebuah citra diperbaiki dengan menggunakan filter rata-rata geometri yang diberikan oleh persamaan berikut:

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in S_{xy}} g(s, t) \right] \frac{1}{mn}$$

Disini, setiap piksel yang diperbaiki diberikan oleh hasil kali masing-masing piksel dalam *subimage window*, kemudian dipangkatkan dengan $1/mn$

3) Filter Rata-rata Harmonik

Operasi filter rata-rata harmonik diberikan oleh persamaan:

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}}$$

Filter rata-rata harmonik bekerja dengan baik untuk noise *salt*, tetapi gagal untuk *noise pepper*. Filter ini juga bekerja baik dengan tipe noise yang lain seperti *noise gaussian*

4) Filter Rata-rata Kontraharmonik

Persamaan :

$$\hat{f}(x, y) = \frac{\sum_{(s,t) \in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s, t)^Q}$$

Dimana Q disebut order dari filter. Filter ini sangat cocok untuk mengurangi atau secara virtual mengeliminasi efek *noisesalt-and-pepper*. Untuk nilai Q positif filter mengeliminasi *noise pepper*. Untuk nilai Q negatif filter mengeliminasi *noise salt*. Filter ini tidak dapat melakukan keduanya secara bersamaan.

2.4.2. Peningkatan Kualitas Citra (*Image enhancement*)

Putra (2010, p.33) mendefinisikan “peningkatan kualitas citra adalah suatu proses untuk mengubah sebuah citra menjadi citra baru sesuai dengan kebutuhan yang diinginkan melalui berbagai cara (*transformation, mathematic operation, filtering, etc.*). Tujuan utama dari peningkatan kualitas citra adalah untuk memproses citra sehingga citra yang dihasilkan lebih baik dari pada citra aslinya. Beberapa operasi yang digunakan dalam peningkatan kualitas citra antara lain:

1) Penyesuaian Kecerahan Citra (*Brightness Adjustment*)

Metode ini dilakukan untuk menganalisis tingkat kecerahan suatu citra yaitu dengan menggunakan histogramnya. Penyesuaian tingkat *brightness* suatu citra dapat dinyatakan sebagai:

$$U' = U + c$$

Dengan U' dan U berturut-turut menyatakan citra setelah dan sebelum penyesuaian kecerahan sedangkan c adalah suatu konstanta yang merupakan faktor penyesuaian. Proses penyesuaian di atas dilakukan dengan menambahkan (mengurangkan) nilai setiap piksel dengan suatu konstanta. Apabila nilai piksel setelah penyesuaian melebihi nilai maksimum intensitasnya yang mungkin untuk citra *grayscale*, nilai maksimum intensitas adalah 255 maka nilai piksel tersebut akan dijadikan 255. Demikian pula sebaliknya, bila nilai piksel hasil penyesuaian lebih kecil dari 0 maka nilai piksel tersebut dijadikan 0.

2) Koreksi Gamma

operasi ini digunakan untuk memperbaiki kecerahan citra (*brightness*). Bentuk umum dari transformasi gamma adalah:

$$U' = U^{\frac{1}{\gamma}}$$

Dengan U' dan U berturut-turut adalah citra setelah dan sebelum mengalami koreksi gamma, sedangkan γ adalah faktor koreksi gamma, dengan kisaran nilai $0 < \gamma < 1$. Bila $\gamma = 1$ maka hasil transformasi akan sama dengan citra masukan.

3) Peregangan Kontras (*Contrast Stretching*)

Mengacu pada pendapat Munawaroh & andreas (2010, p.35) Citra kontras-rendah dicirikan dengan sebagian besar komposisi citranya adalah terang atau sebagian besar gelap.

Operasi ini digunakan untuk memperbaiki kontras citra terutama citra yg memiliki kontras rendah. Pada peregangan kontras, setiap piksel pada citra U ditransformasi dengan menggunakan fungsi sebagai berikut:

$$o(i,j) = \frac{u(i,j) - c}{d - c} (L - 1)$$

Dengan $o(i,j)$ dan $u(i,j)$ berturut-turut piksel sesudah dan sebelum ditransformasi pada koordinat (i,j) , c dan d berturut-turut menyatakan nilai maksimum dan minimum dari piksel pada citra input, dan L menyatakan nilai *grayscale* maksimum. Bila nilai piksel lebih kecil dari 0 maka akan dijadikan 0 dan bila lebih besar dari $(L-1)$ maka akan dijadikan $(L-1)$.

Menurut Yulianto (2008) Tujuan perentangan ini didasarkan pertimbangan bahwa informasi atas objek yang ingin diambil dari data citra terdistribusi pada aras keabuan yang berbeda- beda. (p.90).s

4) Deteksi Tepi

Deteksi tepi berfungsi untuk memperoleh tepi objek. Deteksi tepi memanfaatkan perubahan nilai intensitas yang drastis pada batas dua area. Menurut Watiningsih (2012, p.20) “Deteksi tepi merupakan topik penting dalam pengolahan citra. Tepi memberikan porsi yang signifikan atas informasi yang dikandung dalam citra, sehingga berguna untuk mengekstraksi ciri-ciri atas citra lengkap”.

Umumnya deteksi tepi menggunakan dua macam detektor, yaitu detektor baris (H_y) dan detektor kolom (H_x). Beberapa contoh yang tergolong jenis ini adalah operator *Robert*, *Prewitt*, *Sobel* dan *Frei-chen*. Deteksi tepi dapat dibagi menjadi dua golongan. Golongan pertama disebut deteksi tepi orde pertama (operator *Robert*, *Prewitt* dan *Sobel*) golongan kedua dinamakan deteksi tepi orde kedua (*Laplacian of Gaussian*).

Tabel 2.2 Turunan orde pertama dan kedua pada bentuk turunan kedua.

Turunan	Bentuk Kontinu	Bentuk Diskrit
$\frac{df}{dx}$	$\lim_{\Delta x \rightarrow 0} \frac{f(y, x + \Delta x) - f(y, x)}{\Delta x}$	$f(y, x + 1) - f(y, x)$
$\frac{df}{dy}$	$\lim_{\Delta y \rightarrow 0} \frac{f(y + \Delta y, x) - f(y, x)}{\Delta y}$	$f(y + 1, x) - f(y, x)$
$\nabla f(y, x)$	$\left[\frac{df}{dy}, \frac{df}{dx} \right]$	$[f(y, x + 1) - f(y, x), f(y + 1, x) - f(y, x)]$
$\frac{d^2 f}{dx^2}$	$\lim_{\Delta x \rightarrow 0} \frac{\left(\frac{df}{dx}\right)(y, x + \Delta x) - \left(\frac{df}{dx}\right)(y, x)}{\Delta x}$	$f(y, x + 1) - 2f(y, x) + f(y, x - 1)$
$\frac{d^2 f}{dy^2}$	$\lim_{\Delta y \rightarrow 0} \frac{\left(\frac{df}{dy}\right)(y + \Delta y, x) - \left(\frac{df}{dy}\right)(y, x)}{\Delta y}$	$f(y + 1, x) - 2f(y, x) + f(y - 1, x)$
$\Delta^2 f(y, x)$	$\frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}$	$f(y, x + 1) + f(y, x - 1) - 4f(y, x) + f(y - 1, x) + f(y + 1, x)$

5) Penajaman Citra (*Sharping*)

Menurut Ahmad & Hadinegoro (2012) “Penajaman citra bertujuan untuk meningkatkan gambar dari perspektif visual manusia. Gambar fitur seperti tepi, batas, dan kontras tajam dengan cara yang mereka rentang dinamis meningkat tanpa ada perubahan dalam isi informasi yang melekat dalam data”.

Proses penajaman pada citra adalah menjumlahkan citra asli dengan citra hasil dari operasi deteksi tepi yaitu operator laplacian.

Banyaknya penambahan komponen citra tepi diatur dengan suatu nilai yang disebut sebagai derajat ketajaman α . Besar kecilnya tingkat ketajaman biasa disesuaikan dengan keinginan kita dengan cara mengatur nilai α . Secara matematis ditulis:

$$h(x, y) = f(x, y) + \alpha \nabla^2 f(x, y)$$

6) Penghalusan (*Smoothing*) Citra Menggunakan Filter Rata-rata

Membuat efek penghalusan citra dengan filter rata-rata adalah melakukan proses pemfilteran citra $f(x, y)$ dengan filter rata-rata $g(x, y)$ untuk beberapa ukuran filter.

7) Penghalusan (*Smoothing*) Citra Menggunakan Filter Gaussian

Melakukan proses pemfilteran citra $f(x, y)$ dengan filter Gaussian $g(x, y)$ untuk berbagai ukuran filter.

8) Image Enhancement Menggunakan Filter Maksimum

Memilih intensitas paling terang diantara intensitas piksel yang tercakup dalam filter.

9) Image Enhancement Menggunakan Filter Minimum

Memilih intensitas paling gelap diantara intensitas piksel yang tercakup dalam filter.

10) Image Enhancement Menggunakan Filter Median

Memilih intensitas piksel yang paling tengah, setelah piksel-piksel yang tercakup dalam filter diurutkan.

11) *High Pass Filter*

Mempertahankan (mempertajam) komponen frekuensi tinggi dan menghilangkan (mengurangi) komponen frekuensi rendah. Cara kerja *spasial high pass filter* sama dengan filter penajaman citra. Bedanya, pada *spasial high pass filter* tidak dilengkapi dengan derajat ketajaman α .

12) *Low Pass Filter*

“*Low Pass Filter* adalah proses filter yang mengambil citra dengan gradiasi intensitas yang halus dan perbedaan intensitas yang tinggi akan dikurangi atau dibuang.” (Darujati et al . 2014. p. 35)

Bertujuan mempertahankan komponen frekuensi rendah (perubahan nilai piksel yang bertahap) dan menghilangkan komponen frekuensi tinggi (perubahan nilai piksel secara mendadak) pada citra. Cara kerja *Spasial Low Pass Filter* sama dengan filter rata-rata. Bedanya, pada *Spasial Low Pass Filter* 5 tetangga 4 buah piksel yang berada dipojok tidak diperhitungkan, dan posisi piksel.

13) Transformasi Log

Menurut Wijaya & Kanata (2004. p.47) “transformasi merupakan suatu proses pengubahan data kedalam bentuk lain agar mudah dianalisis”. Transformasi ini sangat berguna dalam menggambarkan grafik ketika pada deretan nilai, di samping ada selisih nilai yang sangat kecil, juga ada selisih nilai yang sangat besar, sehingga ketika digambar dalam grafik maka selisih yang sangat kecil akan sulit terlihat. Transformasi ini meningkatkan nilai citra yang gelap dan mengompres nilai citra yang sangat tinggi.

14) Transformasi *Power Law*

Transformasi *power law* mempunyai dasar sebagai berikut :

$$s = cr^{\gamma}$$

dimana s dan r merupakan tingkat keabuan piksel dari citra output dan input, sedangkan c dan γ adalah konstanta positif. Transformasi ini dapat diaplikasikan pada citra yang mempunyai kontras rendah. Kurva *power law* dengan nilai pecahan γ memetakan nilai-nilai daerah gelap masukan menjadi nilai-nilai *output* yang lebih luas.

15) Transformasi Fourier Diskrit

Transformasi fourier membawa suatu citra dari ruang spasial ke ruang frekuensi (sinyal). Melalui transformasi ini, suatu citra dapat dinyatakan sebagai penjumlahan sinyal sinus atau kosinus dengan amplitudo dan frekuensi yang bervariasi. Transformasi ini terbagi menjadi dua :

- a. DFT 1-Dimensi
- b. DFT 2- Dimensi

2.5. Pengembangan Perangkat Lunak

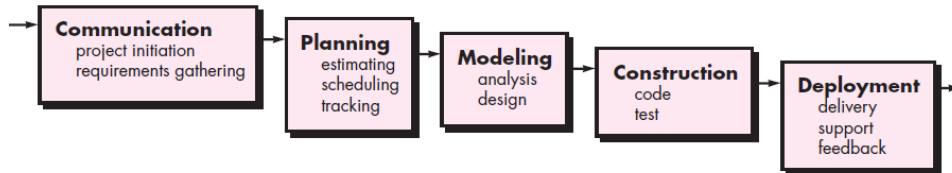
Pengembangan perangkat lunak (*Software development*) merupakan salah satu dari tahap rancangan system rinci/detail dari Siklus Hidup Pengembangan Sistem (*Software Development Life Cycle* atau *SDLC*).

Pressman dalam bukunya *Rekayasa Perangkat Lunak* (2010), menyatakan terdapat beberapa model dari pengembangan perangkat lunak diantaranya adalah Model Proses Prespektif.

1. Model Air Terjun (*Waterfall*)

Model air terjun kadang dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menyiratkan pendekatan yang sistematis dan berurutan pada pengembangan perangkat yang dimulai dengan spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*contruction*), serta penyerahan sistem/perangkat lunak ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak lengkap yang dihasilkan.

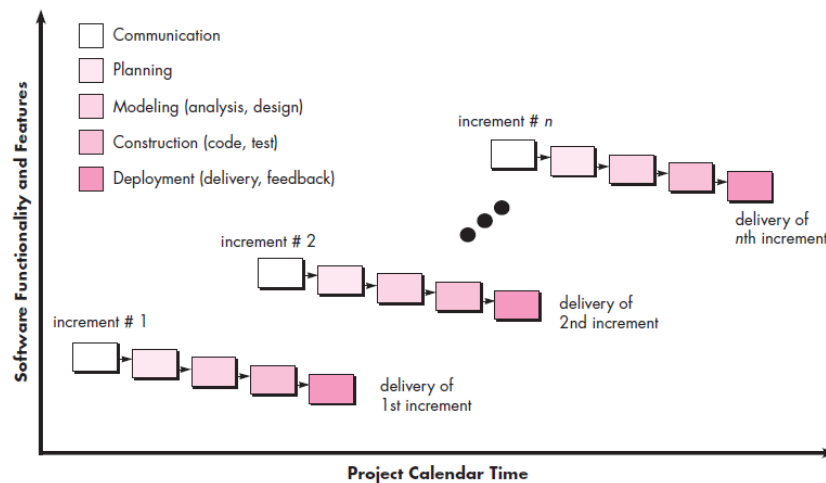
Menurut Irianto & Nugroho (2010, p.2) “Metoda waterfall banyak digunakan karena sangat mudah dikerjakan, dimana satu step dikerjakan setelah satu step selesai dikerjakan”.



Gambar 2.4 model air terjun

2. Model Proses Inkremental

Model penambahan sedikit demi sedikit (*incremental*) menggabungkan elemen-elemen aliran proses linier dan paralel, model incremental menerapkan pada aliran proses linier bentuk-bentuk perangkat lunak yang belum stabil di sepanjang waktu penyelesaian perangkat lunak. Masing-masing urutan linier menghasilkan bagian penambahan dari perangkat lunak dengan cara yang serupa dengan penambahan sedikit demi sedikit (*incremental*) yang dihasilkan oleh suatu aliran proses yang bersifat evolusioner.

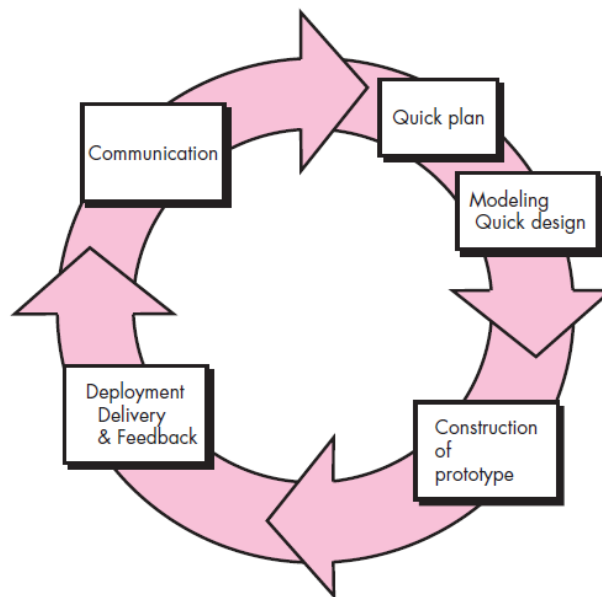


Gambar 2.5 model proses inkremental

3. Model Proses Evolusioner

Bisnis dan spesifikasi kebutuhan produk sering berubah saat pengembangan perangkat lunak berlangsung sehingga lintasan garis lurus ke produk akhir menjadi tidak realistis. Dalam situasi-situasi ini, kita memerlukan suatu model proses perangkat lunak yang telah secara eksplisit dirancang untuk mengakomodasi suatu produk yang akan berubah secara perlahan (berevolusi) sepanjang waktu. Model proses evolusioner ini dicirikan dalam bentuk yang memungkinkan kita mengembangkan perangkat lunak yang semakin kompleks pada versi-versi yang berikutnya.

Model Prototipe



Gambar 2.6 model proses prototipe

Model prototipe dimulai dengan dilakukannya komunikasi antara tim pengembang perangkat lunak dengan pelanggan. Tim pengembang perangkat lunak akan melakukan pertemuan-pertemuan dengan *stakeholder* untuk mendefinisikan sasaran keseluruhan untuk perangkat lunak yang akan dikembangkan, mengidentifikasi spesifikasi kebutuhan apa pun saat ini diketahui, dan menggambarkan area-area dimana definisi lebih jauh pada iterasi selanjutnya merupakan keharusan. Suatu rancangan cepat berfokus pada representasi semua aspek perangkat lunak yang akan terlihat oleh para pengguna akhir. Rancangan cepat (*quick design*) akan memulai konstruksi pembuatan prototipe. Prototipe kemudian akan diserahkan kepada para *stakeholder* dan kemudian mereka akan melakukan evaluasi-evaluasi tertentu terhadap prototipe yang akan dibuat sebelumnya, kemudian akhirnya akan memberikan umpan-balik yang akan digunakan untuk memperhalus spesifikasi kebutuhan.

Model Spiral

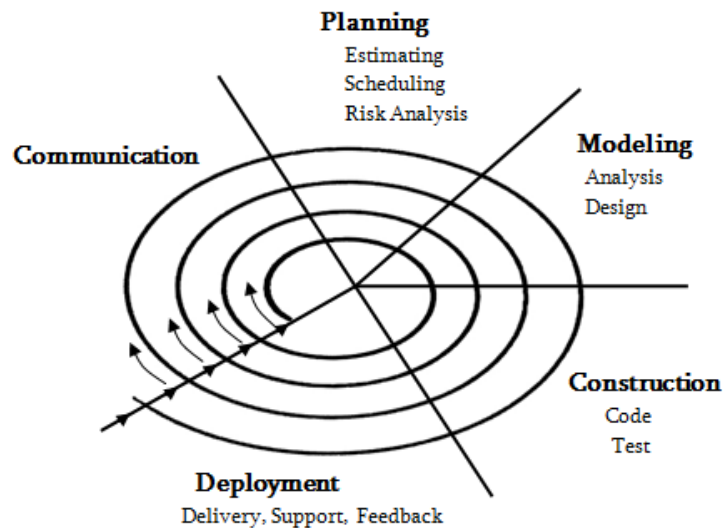


Figure : Spiral Model

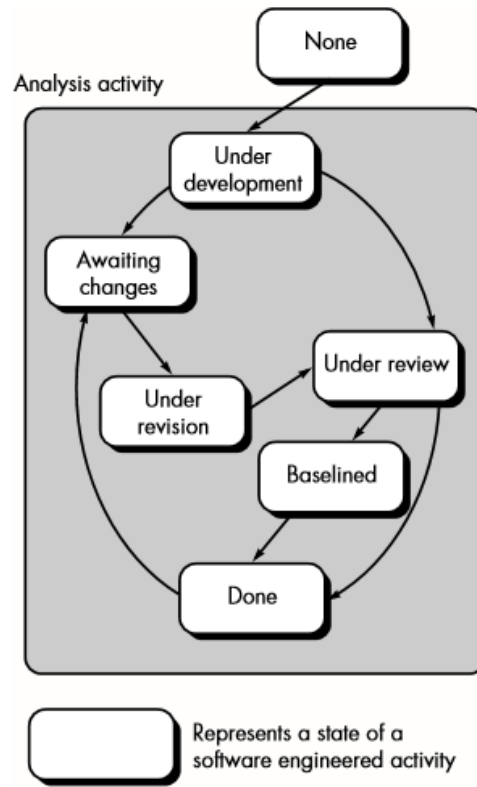
Gambar 2.7 model proses spiral

Model spiral merupakan pendekatan yang cukup realistis untuk diterapkan pada pengembangan sistem-sistem dan perangkat lunak berskala besar.

Tidak seperti model proses yang berakhir saat perangkat lunak dikirimkan ke pengguna, model spiral dapat diadaptasi untuk diterapkan di sepanjang kehidupan perangkat lunak. Lebih jauh, lintasan yang pertama disekitar spiral mungkin merepresentasikan suatu konsep proyek pengembangan yang berawal dari inti spiral dan berlanjut untuk iterasi berganda hingga konsep pengembangan lengkap. Jika konsep dikembangkan kedalam suatu produk nyata, proses akan berlanjut ke arah luar spiral dan suatu proyek pengembangan produk dimulai. Produk yang baru akan bergerak melewati sejumlah iterasi disekitar spiral. Selanjutnya, lintasan di sekitar spiral mungkin bisa digunakan untuk merepresentasikan proyek perbaikan produk perangkat lunak. Esensinya, spiral saat dikerjakan dengan cara seperti ini, tetap bersifat operatif hingga perangkat lunak tidak digunakan lagi. Ada waktu-waktu tertentu saat proses ini tidak aktif, tetapi kapanpun perubahan perubahan dimulai, proses akan bermula pada titik masuk yang sesuai.

4. Model model konkuren

Memungkinkan tim perangkat lunak untuk menggunakan unsur-unsur yang bersifat berulang (iteratif) dan konkuren (berjalan bersamaan) dalam setiap model proses yang dijelaskan dalam bab ini. Aktivitas pemodelan konkuren ini mungkin ada dalam salah satu keadaan pada suatu waktu tertentu. Dengan cara yang sama, aktivitas-aktivitas lain, tindakan lain, atau pekerjaan lain (misalnya **komunikasi** atau **kontruksi**) bisa saja berada pada keadaan yang serupa. Semua aktivitas mungkin hadir secara konkuren tetapi berada pada berbagai keadaan (*state*).



Gambar 2.8 model proses konkuren

