

## BAB II

### KAJIAN LITERATUR

#### 2.1 Tinjauan Pustaka

Pada bagian ini akan menjelaskan hal – hal yang berhubungan dengan topik penelitian dan teori pendukungnya. Bagian ini nantinya juga akan membahas mengenai penghilangan derau, pengembalian citra yang kabur dan penghilangan kabut pada citra serta pendekatan lainnya yang berhubungan dengan objek penelitian.

##### 2.1.1 Citra

Citra (*image*) – istilah lain dari gambar, yang merupakan suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek (Sutojo, 2004). Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh objek. Ketika sumber cahaya menerangi objek, objek memantulkan kembali sebagian cahaya tersebut. Pantulan ini ditangkap oleh alat-alat pengindera optik, misalnya mata manusia, kamera, scanner, dan sebagainya (Thanki & Kothari, 2019). Citra terbagi atas 2 jenis, yaitu citra yang bersifat analog dan citra yang bersifat digital (Sutojo, 2004).

Citra dapat didefinisikan sebagai fungsi  $f(x,y)$  berukuran M baris dan N kolom dengan x dan y adalah koordinat spasial dan amplitudo f di titik koordinat (x, y) dinamakan intensitas atau tingkat keabuan dari citra (Putra, 2010).

Ada banyak cara untuk menyimpan citra digital di dalam memori. Cara penyimpanan menentukan jenis citra digital yang terbentuk. Beberapa jenis citra digital yang sering digunakan adalah citra biner, citra grayscale, dan citra warna (Sutoyo, 2009).

#### 1. Citra Warna

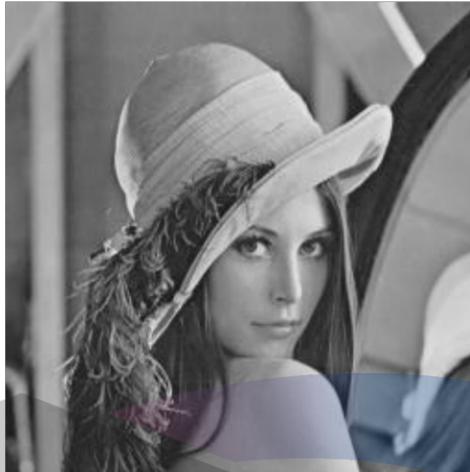
Setiap piksel (pixel) pada citra warna (true color) mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = Red Green Blue). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 256 warna. Berarti setiap piksel (pixel) mempunyai kombinasi warna sebanyak  $2^8 \times 2^8 \times 2^8 = 2^{24} = 16$  juta lebih warna. Itulah sebabnya format ini dinamakan true color karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna yang ada di alam (Hernawati, 2013).



Gambar II-1 Citra Warna  
(Sumber: cwabali.)

## 2. Citra Grayscale

Citra grayscale merupakan citra digital yang hanya memiliki satu nilai pada setiap pikselnya, dengan kata lain nilai bagian red = green = blue. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Citra grayscale memiliki 8 bit (256 kombinasi warna keabuan) dimana terdiri dari rentang nilai 0 sampai 1 (putih hingga mendekati kehitaman) (Munir, 2004).

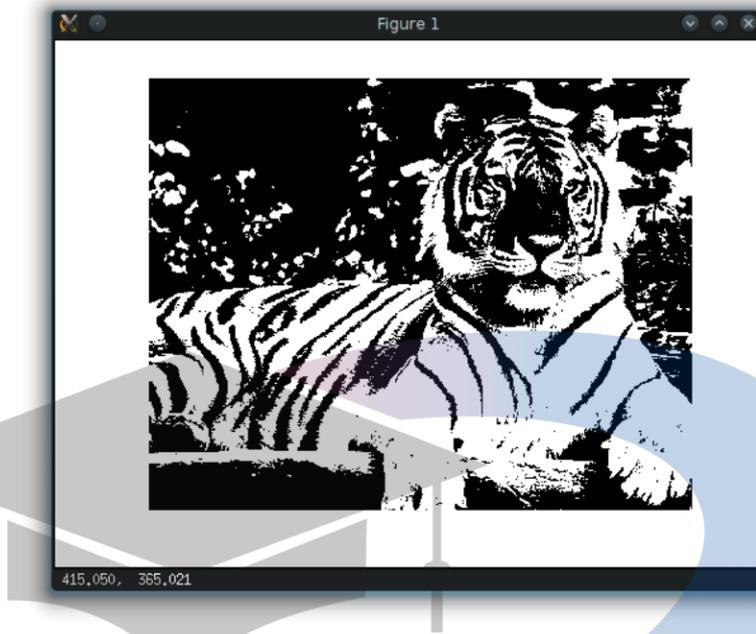


Gambar II-2 Citra *Grayscale*  
(Sumber: Munir, 2004)

### 3. Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai piksel (pixel) yaitu hitam dan putih. Citra biner juga disebut sebagai citra B & W (Black and White) atau monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel (pixel) dari citra biner. Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengembangan, morfologi, ataupun difhering (Munir, 2004).

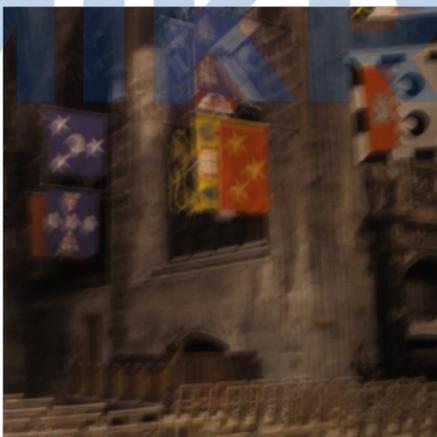
UNIVERSITAS  
MIKROSKIL



Gambar II-3 Citra Biner  
(Sumber: Munir, 2004)

### 2.1.2 Citra *Blur* (Kabur)

Blur adalah daerah gambar yang tidak tajam disebabkan oleh kamera atau gerakan subyek, fokus tidak akurat, atau penggunaan aperture yang memberikan kedalaman medan yang dangkal. Efek blur adalah filter yang membuat transisi halus dan kontras menurun oleh rata-rata pixel di sebelah tepi garis atau wilayah yang ditentukan dimana ada warna transisi yang signifikan (Pan, et al., 2018).



Gambar II-4 Citra *Blur* (Kabur)  
(Sumber: Pan, et al., 2018)

### 2.1.3 Kabut

Secara umum, kabut merupakan tetes – tetes air yang sangat kecil dan berkumpul melayang di udara. Kabut biasanya terlihat di wilayah yang memiliki udara dingin dan terletak di dataran tinggi. Kabut umumnya hampir sama dengan awan, bedanya awan tidak menyentuh tanah, sedangkan kabut menyentuh tanah (Mukono, 1997). Terdapat berbagai jenis kabut, yaitu kabut *advection*, kabut *frontal*, kabut radiasi dan kabut gunung (Ahrens & Henson, 2018).

#### 1. Kabut Advection

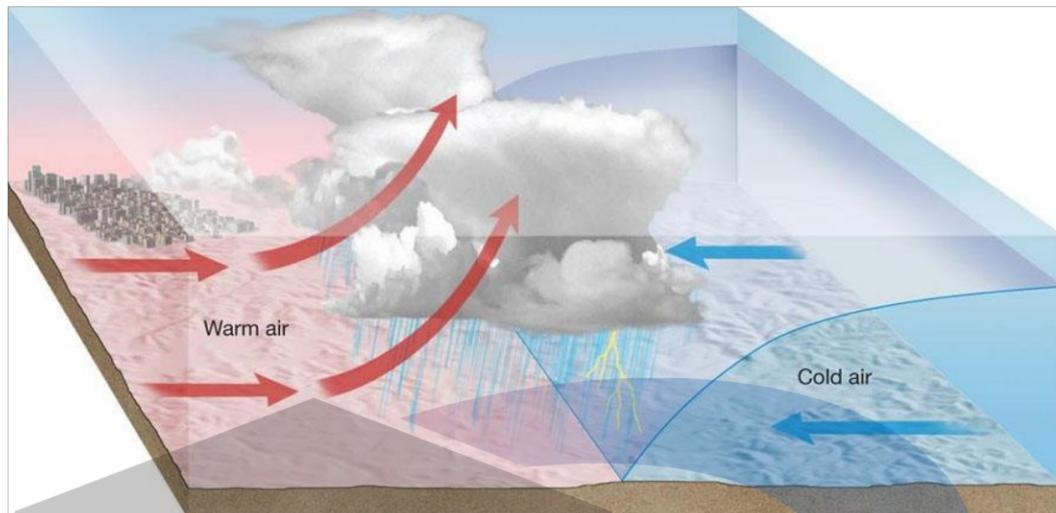
Kabut *advection* adalah kabut yang terbentuk dari aliran udara yang melalui suatu permukaan yang memiliki suhu yang berbeda. Salah satu contoh kabut ini adalah kabut Laut yang terjadi ketika udara yang basah dan hangat mengalir di atas suatu permukaan yang dingin. Kabut laut sering muncul di sepanjang pesisir pantai dan di tepi-tepi danau (Ahrens & Henson, 2018).



Gambar II-5 Kabut Advection  
(Sumber: Ahrens & Henson, 2018)

#### 2. Kabut Frontal

Kabut *frontal* terbentuk melalui suatu pertemuan antara dua masa udara yang berbeda temperaturnya. Kabut ini terbentuk ketika hujan turun dari masa udara yang hangat ke dalam masa udara yang dingin tempat uap air menguap. Dengan demikian akan menyebabkan uap air pada udara dingin melampaui titik jenuh (Ahrens & Henson, 2018).



Gambar II-6 Kabut Frontal  
(Sumber: Ahrens & Henson, 2018)

### 3. Kabut Radiasi

Kabut radiasi terbentuk pada malam yang tenang dan bersih, ketika tanah memancarkan kembali panas ke dalam udara. Satu lapis kabut terbentuk di seluruh permukaan tanah, dan secara bertahap bertambah menjadi tebal. Kabut Radiasi sering muncul di lembah-lembah yang dalam (Ahrens & Henson, 2018).



Gambar II-7 Kabut Radiasi  
(Sumber: Ahrens & Henson, 2018)

### 4. Kabut Gunung

Kabut gunung terbentuk ketika uap air bergerak menuju ke atas melewati lereng-lereng gunung. Udara dingin bergerak ke atas lereng sampai tidak sanggup menahan uap air. Titik-titik kabut kemudian terbentuk di sepanjang lereng gunung (Ahrens & Henson, 2018).



Gambar II-8 Kabut Gunung  
(Sumber: Ahrens & Henson, 2018)

#### 2.1.4 Konvolusi Citra

Konvolusi pada citra sering disebut dengan konvolusi dua-dimensi (konvolusi 2-D) yang diartikan sebagai proses untuk mendapatkan sebuah piksel yang didapatkan berdasarkan hasil piksel itu sendiri dengan piksel tetangganya dengan melibatkan sebuah matrik yang digunakan untuk pembobotannya (Kadir & Susanto, 2013).

$$g(y, x) = \sum_{p=-m_2}^{m_2} \sum_{q=-n_2}^{n_2} h(p + m_2 + 1, q + n_2 + 1) f(y - p, x - q) \dots \dots (2.1)$$

Keterangan:

- $m_2$  = separuh dari tinggi kernel.
- $n_2$  = separuh dari lebar kernel.
- $h$  = kernel, indeks dimulai dari 1.
- $f$  = citra yang akan diolah.

#### 2.1.5 Dark Channel

Sesuai dengan namanya *dark channel* (lapisan kegelapan) merupakan lapisan dari citra yang memiliki peran untuk mengembalikan citra yang *blur* (kabur). Sebagai contoh *dark channel* pada citra dapat dijelaskan dengan rumus berikut (Pan, et al., 2018):

$$D(I)(x) = \min_{y \in N(x)} \left( \min_{c \in \{r, g, b\}} I^c(y) \right) \dots \dots \dots (2.2)$$

Keterangan rumus:

$x$  dan  $y$  merupakan lokasi dari citra.

$N(x)$  merupakan citra yang berpusat pada  $x$ .

$I^c$  merupakan lapisan warna.

Sesuai dengan tahapan yang dijelaskan oleh (Pan, et al., 2018), pengembalian citra yang *blur* (kabur) dibagi ke dalam dua bagian, yaitu (Pan, et al., 2018):

a. Tahapan Pencarian Lapisan *Blur* (Kabur)

Tahapan untuk pencarian lapisan *blur* (kabur) terbagi ke dalam beberapa tahapan, yaitu:

- Masukkan citra yang *blur* (kabur) ( $B$ ).
- Lakukan inisialisasi lapisan  $k$  dimulai dari tingkat kekasarannya.
- Lakukan perulangan sampai iterasi yang ditentukan.
- Lakukan pencarian lapisan  $I$  dengan menggunakan rumus (2.3).

$$B = \sum_t k_t H_t I + n \dots \dots \dots (2.3)$$

Keterangan rumus:

$I$  merupakan citra yang asli.

$n$  merupakan vektor dari  $I$ .

$t$  merupakan indeks dari kamera.

$H_t$  merupakan matriks yang dihasilkan dari matriks homograf.

$K_t$  merupakan bobot kamera ke- $t$ .

- Lakukan pencarian lapisan  $k$  dengan menggunakan rumus (2.4).

$$B = KI + n = Ik + n \dots \dots \dots (2.4)$$

Keterangan rumus:

$k$  merupakan sebuah vektor yang dihasil dari bobot  $K_t$ .

- Keluaran berupa lapisan  $k$  yang akan digunakan untuk pengembalian citra *blur* (kabur).

b. Pengembalian Citra *Blur* (Kabur)

Tahapan untuk pengembalian citra *blur* (kabur) terbagi lagi ke dalam beberapa tahapan setelah selesai pencarian kernel  $k$ , yaitu:

- Masukkan berupa citra yang *blur* (kabur) dan lapisan  $k$  yang telah dicari sebelumnya pada tahapan sebelumnya.
- Lakukan perulangan nilai  $\beta$  sampai dengan nilai maksimumnya.
- Lakukan pencarian nilai  $D(I)$  sesuai dengan definisi dari *dark channel* (lapisan kegelapan) dan membuat matriks  $M$  dengan menggunakan rumus (2.5).

$$M(x, z) = \begin{cases} 1, & z = x, \\ 0, & \text{lainnya.} \end{cases} \dots\dots\dots(2.5)$$

Keterangan rumus:  $x$  dan  $z$  merupakan posisi piksel pada citra.

- Mencari nilai  $u$  menggunakan rumus (2.6).

$$\min_I \|T_k I - B\|_2^2 + \alpha \|\nabla I - g\|_2^2 + \beta \|MI - u\|_2^2, \dots\dots\dots(2.6)$$

Keterangan rumus:

$T_k$  merupakan konvolusi matriks  $k$ .

$B, g,$  dan  $u$  merupakan vektor dari  $B$ .

- Lakukan perulangan nilai  $\alpha$  sampai dengan nilai maksimumnya.
- Mencari nilai  $g$  dengan menggunakan rumus (2.7).

$$\min_u \beta \|D(I) - u\|_2^2 + \lambda \|u\|_0, \dots\dots\dots(2.7)$$

$$\min_g \alpha \|\nabla I - g\|_2^2 + \mu \|g\|_0, \dots\dots\dots(2.7)$$

- Mencari nilai  $I$  dengan menggunakan rumus (2.8).

$$u = \begin{cases} D(I), & |D(I)|^2 \geq \frac{\lambda}{\beta}, \\ 0 & \text{lainnya,} \end{cases} \dots\dots\dots(2.8)$$

- Menghasilkan citra  $I$  yang merupakan citra yang terbebas dari *blur* (kabur).

Kedua tahapan ini juga akan dilakukan proses terhadap penghilangan kabut pada citra.

### 2.1.6 Pengembalian Citra Kabur dan Penghilangan Kabut

Dalam beberapa dekade terakhir telah banyak penelitian yang telah melakukan pengembalian citra yang kabur seperti yang dilakukan oleh (Fergus, et al., 2006) menggunakan pengabungan *gaussians* untuk mempelajari *gradien* dari citra dengan kerangka kerja variasi *bayesian*. Pada penelitian yang lain

menunjukkan bahwa metode yang dipakai oleh (Fergus, et al., 2006) dapat bekerja dengan baik, namun pendekatan ini terlalu banyak melakukan komputasi dan membutuhkan waktu yang lama untuk melakukan eksekusi.

Pada penelitian yang dilakukan oleh (Lou, et al., 2011) menggunakan metode MAP (*Maximum A Posterior*). Metode tersebut berfokus untuk menentukan batas tepi sehingga memerlukan penyaringan yang heuristik dan ambang batas yang menghilangkan struktur citra yang halus.

Pada penelitian yang dilakukan oleh (Chen, et al., 2014) menggunakan metode berbasis statistik untuk mengembalikan citra yang kabur. Metode tersebut menghasilkan hasil yang lebih baik daripada algoritma pengembalian citra yang kabur pada umumnya, namun untuk penerapan pada aplikasi memerlukan operasi tertentu.

Penelitian yang dilakukan oleh (Pan, et al., 2018) menggunakan metode *dark channel* (lapisan kegelapan) yang memberikan hasil yang lebih baik dibandingkan dengan metode yang lainnya. Pada penelitian yang dilakukan, memberikan hasil 27.94 dB (desibel) dibandingkan dengan metode yang lainnya. Pada penelitian ini nantinya akan dilakukan pengujian terhadap metode yang digunakan terhadap beberapa skenario yang telah diterapkan. Skenario tersebut terdiri dari penghilangan derau, pengembalian citra kabur, dan penghilangan kabut. Hasil dari skenario tersebut diharapkan nantinya bisa diterapkan dalam penelitian lebih lanjut.

### 2.1.7 Derau

Derau adalah sebuah atau kumpulan piksel yang tidak dikehendaki dan mengganggu kualitas sebuah citra. Derau dapat disebabkan oleh gangguan fisik (optik) pada alat akuisisi atau faktor lingkungan eksternal baik akibat proses pengolahan yang tidak sesuai. Selain itu, derau juga dapat disebabkan oleh kotoran – kotoran yang terdapat pada citra. Derau umumnya variasi intensitas suatu piksel dengan piksel – piksel tetangganya. Secara visual, gangguan mudah dilihat oleh mata karena tampak berbeda dengan piksel tetangganya. Piksel yang mengalami gangguan umumnya memiliki frekuensi tinggi. Komponen citra yang berfrekuensi

rendah umumnya mempunyai piksel konstan atau berubah sangat lambat (Munir, 2004).

Reduksi derau adalah suatu proses menghilangkan atau mengurangi derau dari suatu piksel. Reduksi derau secara konsep hampir sama penerapannya pada setiap jenis signal. Namun berdasarkan fakta dari proses implementasinya, reduksi derau tergantung dari jenis signal yang akan diproses (Munir, 2004).

Derau terbagi ke dalam beberapa jenis, diantaranya derau *gaussian*, derau garam dan merica dan derau *speckle*.

### 1. Derau Gaussian

Derau *gaussian* adalah model derau yang memiliki kerapatan probabilitas (*probability density function* / PDF) yang diberikan kurva *Gaussian*. PDF yang mewakili sifat paling acak dalam bentuk satu dimensi seperti berikut (Munir, 2004):

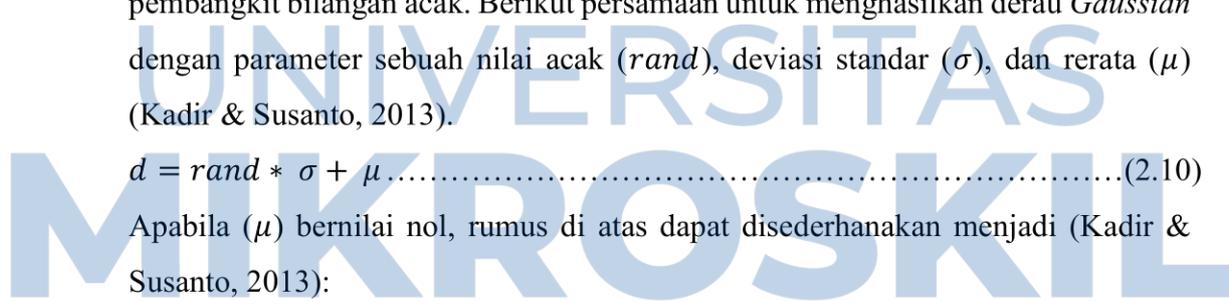
$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \dots\dots\dots(2.9)$$

Derau *gaussian* umumnya muncul akibat efek optis, misalnya *blur* akibat kamera lensa yang tidak fokus, pencahayaan yang singkat dan gangguan atmosfer namun, derau *gaussian* juga dapat dibangkitkan dengan menggunakan fungsi pembangkit bilangan acak. Berikut persamaan untuk menghasilkan derau *Gaussian* dengan parameter sebuah nilai acak (*rand*), deviasi standar ( $\sigma$ ), dan rerata ( $\mu$ ) (Kadir & Susanto, 2013).

$$d = rand * \sigma + \mu \dots\dots\dots(2.10)$$

Apabila ( $\mu$ ) bernilai nol, rumus di atas dapat disederhanakan menjadi (Kadir & Susanto, 2013):

$$d = rand * \sigma \dots\dots\dots(2.11)$$

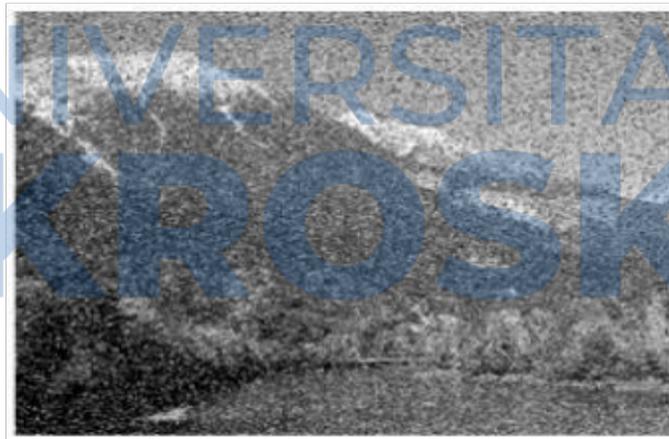




Gambar II-9 Derau Gussian  
(Sumber: Nascimento, J., dan Marques, J., 2006)

## 2. Derau Garam dan Merica (*Salt and Papper Noise*)

Derau garam dan merica biasa dinamakan sebagai derau implus positif dan negatif, derau tembakau atau derau biner. Derau ini biasa disebabkan oleh gangguan yang tiba – tiba dan tajam pada proses pengambilan citra. Bentuknya berupa bintik – bintik putih dan hitam di dalam citra. Derau garam dan merica sering muncul pada citra yang diperoleh melalui kamera (Myler & Weeks, 1993).



Gambar II-10 Derau *Salt and Papper*  
(Sumber: Kadir & Susanto, 2013)

## 3. Derau Speckle

Derau *speckle* merupakan derau ganda. Derau ini ditambahkan pada citra menggunakan persamaan  $J = I + n * V$ , dimana  $n$  terdistribusi random seragam

dengan *mean* 0 dan *variance*  $V$ .  $V$  adalah konstanta *non-negative* yang besarnya dapat berubah – ubah. Default nilai untuk  $V$  adalah 0.04. Makin besar nilai  $V$  maka citra akan semakin kabur (Thanki & Kothari, 2019).

### 2.1.8 Penapisan Derau (Penghilangan Derau)

Penapisan derau merupakan proses yang bertujuan menekan gangguan (noise) pada citra. Gangguan tersebut biasanya muncul sebagai akibat dari hasil pencahayaan yang tidak bagus (*sensor noise, photographic grain noise*) atau akibat saluran transmisi. Operasi pelembutan dapat dilakukan pada ranah spasial maupun pada ranah frekuensi. Pada ranah spasial, operasi pelembutan dilakukan dengan mengganti intensitas suatu pixel dengan rata – rata dari nilai pixel tersebut dengan nilai pixel – pixel tetangganya (Thanki & Kothari, 2019).

Operasi perata – rata dapat dipandang sebagai konvolusi antara citra  $f(x,y)$  dengan penapis  $h(x,y)$  (Thanki & Kothari, 2019):

$$g(x,y) = f(x,y)*h(x,y).....(2.12)$$

penapis  $h$  disebut penapis rerata (*mean filter*). Dalam ranah frekuensi, operasi konvolusi tersebut adalah (Thanki & Kothari, 2019):

$$G(u,v) = F(u,v)*H(u,v).....(2.13)$$

penapis  $H(x,y)$  pada operasi pelembutan citra disebut juga penapis lolos rendah (*low pass filter*), karena penapis tersebut menekan komponen yang berfrekuensi tinggi (misalnya pixel gangguan, pixel tepi) dan meloloskan komponen yang berfrekuensi rendah. Penapisan derau pada citra digital terdapat beberapa macam, yaitu (Thanki & Kothari, 2019):

#### a. Penapis lolos rendah

Penapis rata – rata adalah salah satu penapis lolos rendah yang paling sederhana. Aturan untuk penapis lolos rendah adalah:

- Semua koefisien penapis harus positif, dan
- Jumlah semua koefisien harus sama dengan 1.

Jika jumlah semua koefisien lebih besar dari 1, maka konvolusi menghasilkan peningkatan (tidak diinginkan). Jika jumlah koefisien kurang dari 1, maka yang

dihasilkan adalah penurunan, dan nilai mutlak setiap pixel di seluruh bagian citra berkurang. Akibatnya, citra hasil pelembutan tampak lebih gelap.

Jika citra hasil penapisan lolos rendah dikurangi dari citra semula (yang mengandung derau), maka yang dihasilkan adalah peningkatan relatif komponen citra yang berfrekuensi tinggi tanpa peningkatan komponen derau.

Penapis lolos rendah merupakan penapis linjar (*linear*). Operasi pelembutan dapat juga dilakukan dengan menggunakan beberapa penapis lain, yaitu (Thanki & Kothari, 2019):

- 1) Penapis minimum,
- 2) Penapis maksimum, dan
- 3) Penapis median.

b. Penapis median

Penapis ini dikembangkan oleh Tukey. Pada penapis median, suatu *window* memuat sejumlah *pixel*. *Window* digeser titik demi titik pada seluruh daerah citra. Pada setiap penggeseran dibuat *window* baru. Titik tengah dari jendela ini diubah dengan nilai median dari *window* tersebut.

Jadi penapis median menghilangkan nilai *pixel* yang sangat berbeda dengan *pixel* tetangganya. Dalam beberapa kasus dapat dibuktikan bahwa penapis median memberikan hasil yang lebih baik dibandingkan penapis rerata untuk citra yang mengalami gangguan dalam bentuk *spike* berupa bercak – bercak putih.

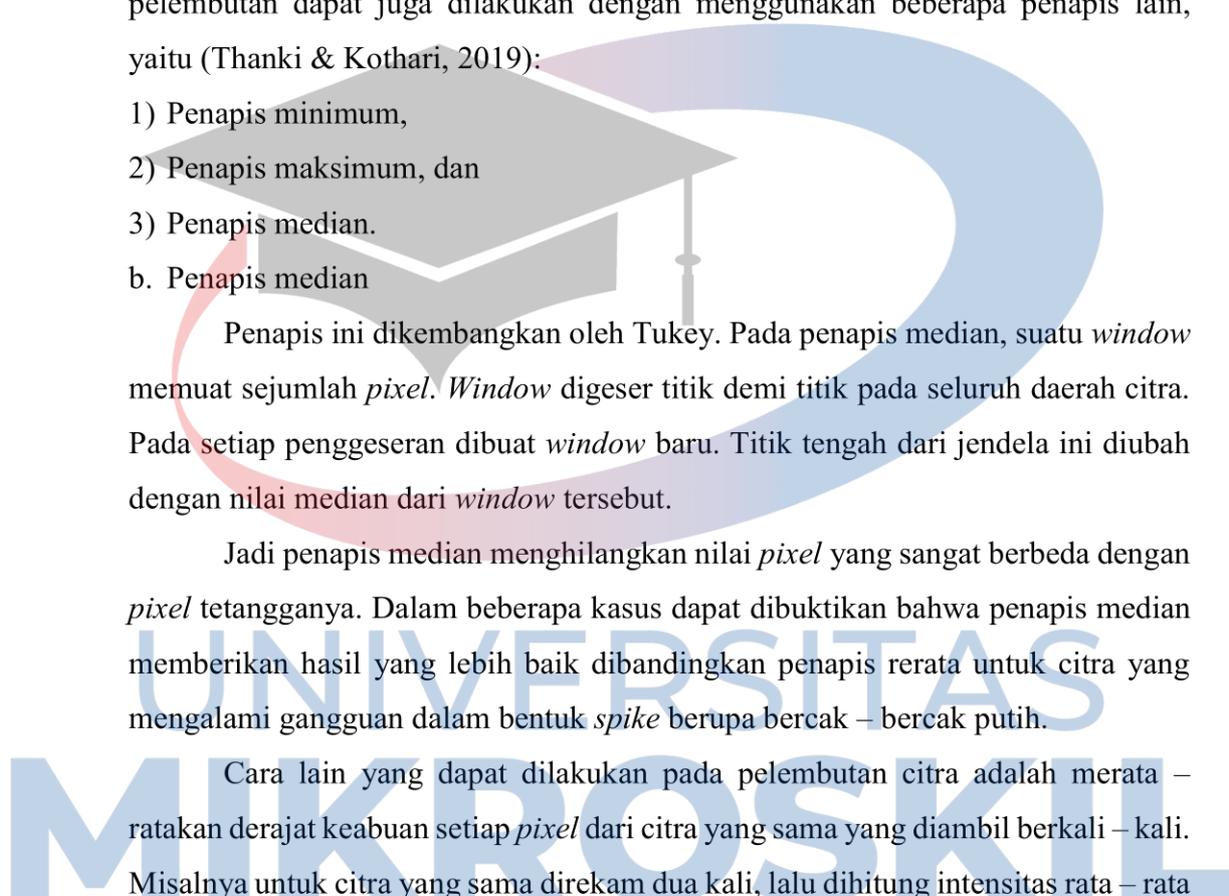
Cara lain yang dapat dilakukan pada pelembutan citra adalah merata – ratakan derajat keabuan setiap *pixel* dari citra yang sama yang diambil berkali – kali. Misalnya untuk citra yang sama direkam dua kali, lalu dihitung intensitas rata – rata untuk setiap *pixel*:

$$f^*(x,y) = \frac{1}{2} \{f_1(x,y) + f_2(x,y)\} \dots \dots \dots (2.14)$$

c. Penapis lolos tinggi

Aturan penapis lolos tinggi:

- 1) Koefisien penapis boleh positif, negative, atau nol, dan
- 2) Jumlah semua koefisien adalah 0 atau 1.



### 2.1.9 Adaptive Fuzzy Filter

Metode AFF (*Adaptive Fuzzy Filter*) adalah sebuah teknik *fuzzy* yang di terapkan dalam masalah restorasi citra. Terdapat 4 bagian utama dalam AFF, yaitu *fuzzy rank selection*, *fuzzy weighted*, *fuzzy network* (FNN) dan *softswitching*. Teknik ini dilakukan pada signal yang terdapat banyak dimensi yang dikonversi menjadi *vector* (Choi & Krishnapuram, 1997).

#### FNN Filter

Baik RCFS *filter* dan *switching fuzzy filter* tidak bisa menyesuaikan diri mereka dengan variasi lingkungan. Mereka kurang akan kemampuan beradaptasi. FNN *filter* dapat meningkatkan dengan sangat banyak kemampuan menyesuaikan diri *filter* dengan menggunakan teknik jaringan saraf (Choi & Krishnapuram, 1997).

#### a. Neuro-Fuzzy Filter

Diberi jendela operasi  $X$  dengan  $3 \times 3$ , hitung perbedaan  $\Delta x_i = x_p - x_i$  ( $i = 1, \dots, 8$ ),  $\Delta x'_j = x_p - x'_j$  ( $j = 1, \dots, 16$ ) antara  $x_p$  dan *pixel* yang berdekatan di  $X$ ,  $x_p$  dan 16 piksel luar yang berdekatan dari  $X$ , masing – masing. Memakai  $\{\Delta x_i\}$  dan  $\{\Delta x'_j\}$ , penulis dalam (Choi & Krishnapuram, 1997) memberikan aturan inferensi *fuzzy* seperti (Choi & Krishnapuram, 1997). Tujuan utama *neuro-fuzzy filter* adalah untuk menentukan nilai  $\Delta y$  yang disesuaikan untuk pusat tingkat abu – abu  $x_p$ . Jadi, *output*  $y$  dari *filter* dihitung sebagai berikut:  $y = x_p + \Delta y$ .  $\Delta y$  adaptif disesuaikan dengan FNN. Ini adalah jaringan *feedforward* lima *level*. Input adalah  $\Delta x_i$  ( $i = 1, \dots, 8$ ),  $\Delta x'_j$  ( $j = 1, \dots, 16$ ). Lapisan *fuzzification* terdiri dari empat bagian, yang bagian pertama dan ketiga dilengkapi dengan fungsi keanggotaan  $\tilde{L}_p$  dari himpunan *fuzzy* “*positively large*”. *Output* yang sesuai didefinisikan sebagai berikut (Choi & Krishnapuram, 1997):

$$O_{1i}^{(2)} = \tilde{L}_p(\Delta x_i) (i = 1, \dots, 8) \dots\dots\dots(2.15)$$

$$O_{3j}^{(2)} = \tilde{L}_p(\Delta x'_j) (j = 1, \dots, 16) \dots\dots\dots(2.16)$$

Dan bagian kedua dan keempat dengan fungsi keanggotaan  $\tilde{L}_n$  dari himpunan fuzzy “negatively large” mempunyai *output* masing – masing sebagai berikut (Choi & Krishnapuram, 1997):

$$O_{2_i}^{(2)} = \tilde{L}_n(\Delta x_i) (i = 1, \dots, 8) \dots\dots\dots(2.17)$$

$$O_{4_j}^{(2)} = \tilde{L}_n(\Delta x'_j) (j = 1, \dots, 16) \dots\dots\dots(2.18)$$

Dengan demikian, dalam kesimpulan lapisan I, *output* dari bagian-*k* adalah sebagai berikut (Choi & Krishnapuram, 1997):

$$O_{k_i}^{(3)} = \bigvee_{i'=1}^8 (O_{k_i}^{(2)} \cdot w_{i,i'}^k) (k = 1, 2) \dots\dots\dots(2.19)$$

$$O_{k_j}^{(3)} = \bigvee_{j'=1}^{16} (O_{k_j}^{(2)} \cdot w_{j,j'}^k) (k = 3, 4) \dots\dots\dots(2.20)$$

Output dari dua unit di inferensi lapisan II adalah (Choi & Krishnapuram, 1997):

$$O_1^{(4)} = \left( \bigwedge_{i=1}^8 \{O_{1_i}^{(3)}\} \right) \wedge \left( \bigwedge_{j=1}^{16} \{O_{3_j}^{(3)}\} \right) \dots\dots\dots(2.21)$$

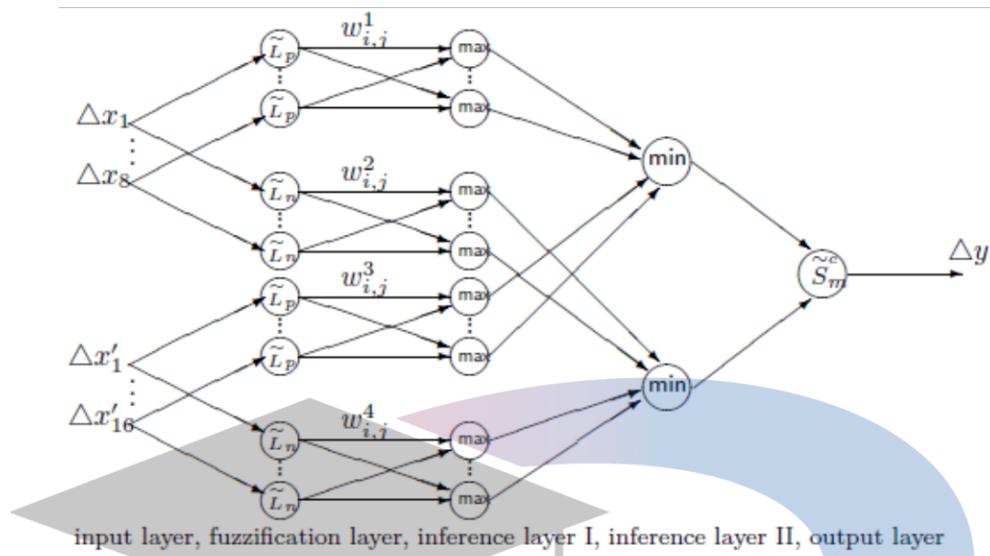
$$O_2^{(4)} = \left( \bigwedge_{i=1}^8 \{O_{2_i}^{(3)}\} \right) \wedge \left( \bigwedge_{j=1}^{16} \{O_{4_j}^{(3)}\} \right) \dots\dots\dots(2.22)$$

Output dari FNN sebagai  $\Delta y = (L - 1)$ .

$$\tilde{S}_m^c \left( (L - 1)(O_1^{(4)} - O_2^{(4)}) \right) \cdot \text{sign}(O_1^{(4)} - O_2^{(4)}) .$$

Dengan demikian, FNN *filter* dapat direpresentasikan sebagai  $Y_{fn}(X) = x_p + \Delta y$ . Dalam (Russo & Ramponi, 1994), bobot koneksi  $w_{i,i'}^k$ ,  $w_{j,j'}^k$  dan *fuzzy set*  $\tilde{S}_m$ , akibatnya pelengkap ditentukan oleh algoritma genetika. *Filter*  $Y_{fn}$  dapat menghapus intensitas *noise* yang tinggi. Selanjutnya mungkin juga dapat menyesuaikan dirinya dengan variasi lingkungan. Dengan menjamin kinerja yang baik dari *filter* di probabilitas impuls *noise* yang berbeda. Jika pasangan operator *fuzzy* (*max*, *min*) tidak langsung diberikan, tapi dipilih secara rasional untuk masalah – masalah yang berhubungan (Russo & Ramponi, 1994), dapat meningkatkan kinerja  $Y_{fn}$  lebih lanjut.

Namun, dalam aplikasi kinerja penyaringan *neuro-fuzzy filter* akan menjadi sangat buruk jika *noise* tidak impuls, tetapi campuran beberapa *noise*, seperti impuls dan *Gaussian noise*. Selanjutnya, banyak masalah tentang FNN sebagai Gambar. II-11 terkait dengan *neuro-fuzzy filter* tetap harus dipecahkan. Sebagai contoh:

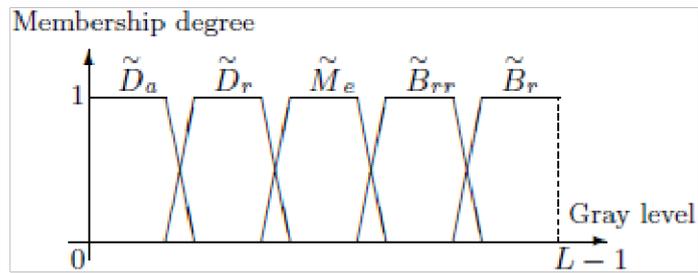


Gambar II-11 Arsitektur *neuro – fuzzy filter*  
(Sumber: Choi & Krishnapuram, 1997)

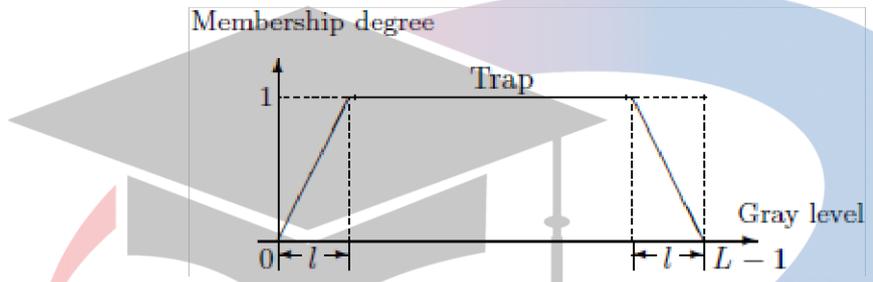
Apakah bisa pilih pasangan lainnya dari operator *fuzzy* daripada (*max*, *min*) untuk meningkatkan kinerja penyaringan *filter*? Bagaimana mencari beberapa fungsi keanggotaan yang cocok yang menjadi ciri *fuzzy set*  $\tilde{L}_n$ ,  $\tilde{L}_p$ , dan  $\tilde{S}_m$ ? Bagaimana mengembangkan beberapa aturan inferensi *fuzzy* yang lebih rasional untuk menentukan arsitektur FNN? dan seterusnya.

#### b. Selection Type FNN Filter

Asumsikan bahwa nilai – nilai tingkat abu – abu untuk citra dua dimensi  $F = \{F(t_1, t_2), 0 \leq t_1, t_2 \leq N\}$  milik  $[0, L - 1]$ , yaitu  $\forall (t_1, t_2) \in \{0, 1, \dots, N\}^2, F(t_1, t_2) \in [0, L - 1]$ . Memperkenalkan partisi *fuzzy*  $[0, L - 1]$  untuk mendeskripsikan citra  $F$  dengan *fuzzy sets*  $\tilde{G}_1, \dots, \tilde{G}_{k_0}$ . Sebagai contoh, *fuzzy sets* masing – masing menjadi “dark ( $\tilde{D}_a$ )”, “dark er ( $\tilde{D}_r$ )”, “medium ( $\tilde{M}_e$ )”, “brighter ( $\tilde{B}_{rr}$ )”, “bright ( $\tilde{B}_r$ )”, yang ditunjukkan pada Gambar. II-12.



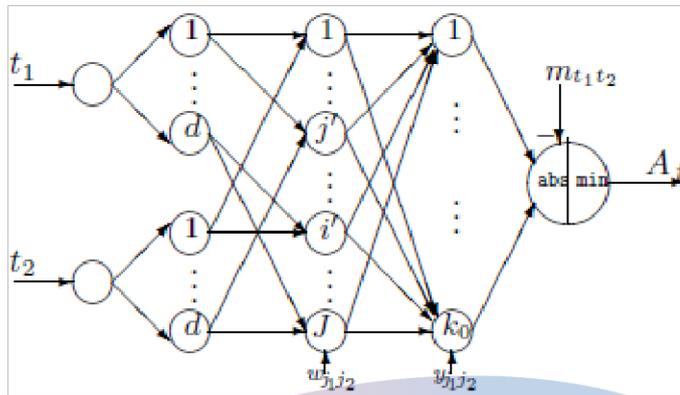
Gambar II-12 Tingkat abu – abu *fuzzy sets*  
(Sumber: Choi & Krishnapuram, 1997)



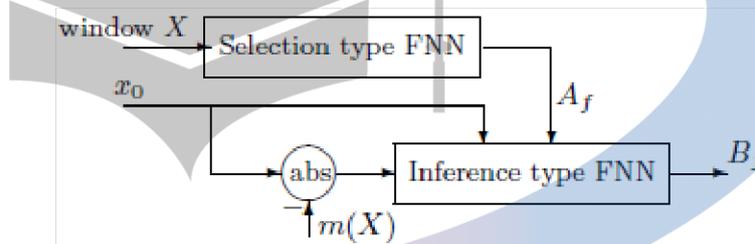
Gambar II-13 *Mean Fuzzy Set*  
(Sumber: Choi & Krishnapuram, 1997)

Dalam rangka untuk menekan impuls *noise*, diperkenalkan *Mean fuzzy set Trap*, yang bentuknya dapat diwakili sebagai bilangan *fuzzy* trapesium sebagai Gambar. II-13. Dengan tingkat abu - abu *fuzzy sets* dan *mean fuzzy set selection type* FNN dapat dikembangkan. Tetapkan jendela operasi  $X = \{x_1, \dots, x_p, \dots, x_d\}$ , dimana  $x_p$  adalah tingkat abu - abu *pixel* pusat yaitu  $x_p = F(t_1, t_2)$ . Struktur topologi *selection type* FNN ditunjukkan pada Gambar. II-14, dan hubungan I/O dinyatakan sebagai berikut:

UNIVERSITAS  
MIKROSKIL



Gambar II-14 Selection type FNN  
(Sumber: Choi & Krishnapuram, 1997)



Gambar II-15 Selection type FNN filter  
(Sumber: Choi & Krishnapuram, 1997)

$$\begin{cases} m^k(X) = \frac{\sum_{j=1}^d x_j \tilde{G}_k(x_j)}{\sum_{j=1}^d \tilde{G}_k(x_j)}, & m(X) = \frac{\sum_{j=1}^d x_j \text{Trap}(x_j)}{\sum_{j=1}^d \text{Trap}(x_j)} \\ A_f = m^{k'}(X), \text{ dimana} & \dots\dots\dots(2.23) \\ k' = \max_{k_1} \{k_1 : |m(X) - m^{k_1}(X)| = \min_k \{|m(X) - m^k(X)|\}\}. \end{cases}$$

Jika F adalah citra bebas dari *noise*, nilai – nilai tingkat abu – abu  $x_1, \dots, x_p, \dots, x_d$  di jendela X kira – kira sama (Kim & Cho, 1994), yaitu  $x_1 \approx \dots \approx x_p \approx \dots \approx x_d \approx m(X)$ . Jadi jika  $u$  adalah variabel *output*, dan  $v = |x_p - m(X)|$  adalah variabel *input*, maka diperoleh aturan inferensi Mamdani sebagai berikut (Liu & Li, 2000):

JIKA  $v$  adalah “*small*” MAKA  $v$  adalah  $\tilde{x}_p$ ;  
 JIKA  $v$  adalah “*large*” MAKA  $v$  adalah  $\tilde{A}_f$ ;

Dimana  $\tilde{x}_p, \tilde{A}_f$  masing – masing berarti *fuzzifications* dari  $x_p, A_f$ . Dan  $x_p \in \ker(\tilde{x}_p), A_f \in \ker(\tilde{A}_f)$ . Pada himpunan tingkat abu – abu  $[0, L - 1]$  mendefinisikan  $\tilde{S} = \text{“small”}, \tilde{L} = \text{“large”}$  (Choi & Krishnapuram, 1997).

$$B_f = \frac{x_p \tilde{S}(v) + A_f \tilde{L}(v)}{\tilde{S}(v) + \tilde{L}(v)} \dots\dots\dots(2.24)$$

$$= \frac{x_p \tilde{S}(|x_p - m(X)|) + A_f \tilde{L}(|x_p - m(X)|)}{\tilde{S}(|x_p - m(X)|) + \tilde{L}(|x_p - m(X)|)}.$$

Persamaan 2.23 dan 2.24 membentuk hubungan I/O dari *selection type* FNN *filter*, yang strukturnya ditampilkan sebagai Gambar. II-15.

Untuk mencari *selection type* FNN *filter* yang optimal, itu sudah cukup untuk merancang algoritma pembelajaran untuk masing – masing parameter dari *selection type* FNN dan *inference type* FNN. Algoritma pembelajaran untuk *selection type* FNN bertujuan untuk menentukan partisi dari himpunan tingkat abu – abu  $[0, L - 1]$ , yaitu menentukan nilai  $k_0$ , bilangan *fuzzy*  $\tilde{G}_1, \dots, \tilde{G}_{k_0}$  di partisi *fuzzy*. Algoritma dapat digambarkan sebagai berikut (Choi & Krishnapuram, 1997):

- i. Untuk  $n$ -bit citra digital, variasi tidak melebihi  $2^{n-4}$  di tingkat abu - abu citra tidak akan menyebabkan perubahan visual yang jelas (Kuo, et al., 2000). Jadi partisi  $[0, L - 1]$  menjadi  $2^{n-4}$  bagian – bagian yang identik.
- ii. Carilah daerah konsentrasi tinggi dari tingkat abu – abu citra  $F$ : Hitung jumlah  $\Gamma_k$  dari  $F$  milik  $I_k \triangleq [(k - 1)L/(2^{n-4}), (kL/(2^{n-4}))](k = 1, \dots, 2^{n-4})$ . Dan membedakan  $\Gamma_k \geq \eta$ ? Jika ya,  $I_k$  disebut daerah konsentrasi tingkat abu – abu  $F$ , dimana  $\eta$  adalah konstan yang diberikan sesuai dengan citra  $F$ .
- iii. Tentukan partisi *fuzzy*  $[0, L - 1]$ . Biarkan  $k_0$  menjadi sejumlah daerah konsentrasi tingkat abu – abu. Dan mereka adalah  $I_{i_1}, \dots, I_{i_{k_0}}$ . Jadi, bilangan *fuzzy*  $\tilde{G}_1, \dots, \tilde{G}_{k_0}$  dapat dibangun.

Dalam definisi *mean fuzzy set* untuk *selection type* FNN, parameter  $l$  diasumsikan sekitar 3, dan seleksi nilai standar adalah  $m(X)$ , maka didapatkan hasil yang baik untuk menghilangkan impuls *noise* (Choi & Krishnapuram, 1997).

Berdasarkan algoritma pembelajaran untuk *selection type* FNN, dengan kembali menyebarkan algoritma pembelajaran tentang  $s_1, s_2$ , inferensi FNN dapat dibangun. Dengan demikian, *selection type* FNN *filter* dapat ditentukan (Choi & Krishnapuram, 1997).

Pada (Liu & Li, 2000) didapatkan citra restorasi dengan kualitas tinggi dengan *selection type* FNN *filter* dari citra *noise* terdegradasi oleh probabilitas

impuls *noise* yang tinggi atau rendah. Selanjutnya, jika *noise* adalah *hybrid*, kinerja penyaringan yang baik dari *filter* dapat juga dipastikan.

Karena sistem *fuzzy* umum dapat dilakukan dengan beberapa FNN (Liu & Li, 2000) itu adalah pelajaran utama untuk penelitian masa depan bahwa beberapa aturan *fuzzy* rasional harus dikembangkan, dan FNN sederhana yang mudah dipahami diterapkan untuk menghilangkan *noise*. Banyak masalah dasar yang berkaitan dengan subjek diselesaikan dalam (Liu & Li, 2000).

### 2.1.10 Penghilangan Derau

Penelitian yang dilakukan oleh (Hiremath, et al., 2012) menggunakan *contourlet harris detector* untuk menghilangkan derau pada citra medis. Metode tersebut berhasil menghilangkan derau dengan tingkat keberhasilan mencapai 19.05 dB (desibel), namun sesuai dengan (Welstead, 1999) bahwa citra yang baik adalah citra yang memiliki PSNR lebih dari 30, sehingga metode tersebut masih belum dikatakan baik untuk menghilangkan derau.

Pada penelitian yang dilakukan oleh (Lukac, et al., 2005) menggunakan *vector median filter* (VMF). VMF merupakan sebuah vektor yang melakukan proses dengan menerapkan *scalar median filter*. Metode tersebut berhasil menghilangkan derau dengan tingkat kesuksesan mencapai 17.75 dB (desibel), namun sesuai dengan (Welstead, 1999) bahwa citra yang baik adalah citra yang memiliki PSNR lebih dari 30, sehingga metode tersebut masih belum dikatakan baik untuk menghilangkan derau.

Penelitian yang dilakukan oleh (Halim, et al., 2017) menggunakan kombinasi dari adaptive fuzzy filter (AFF) dan fuzzy base enhancement technique (FBET) untuk meningkatkan kualitas pada citra. Penelitian tersebut berhasil menghilangkan derau pada citra hingga mencapai 28.37 dB (desibel), namun sesuai dengan (Welstead, 1999) bahwa citra yang baik adalah citra yang memiliki PSNR lebih dari 30, sehingga metode tersebut masih belum dikatakan baik untuk menghilangkan derau. Oleh karena itu, pada penelitian kali ini akan dilakukan kombinasi kembali antara AFF dengan *dark channel* (lapisan kegelapan).

Diharapkan dengan dilakukan kombinasi pada kedua metode tersebut dapat meningkatkan kualitas citra.

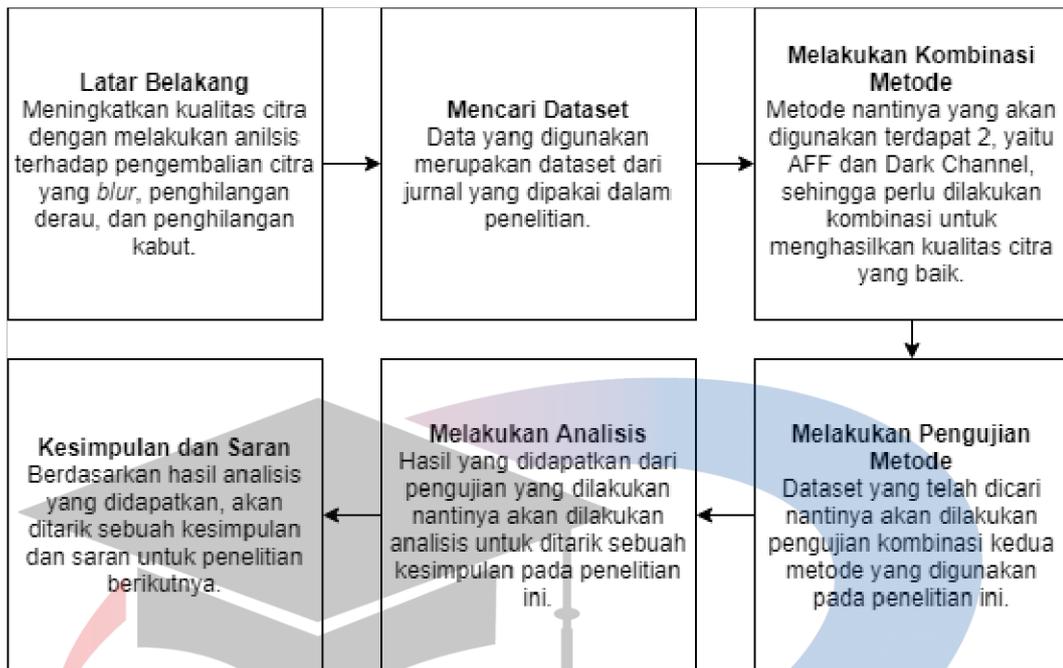
## 2.2 Kerangka Konsep / Pola Pikir Pemecahan Masalah

Penelitian ini dimulai dari ide untuk meningkatkan kualitas citra dengan dengan melakukan analisis terhadap pengembalian citra yang *blur*, penghilangan derau, dan penghilangan kabut. Analisis tersebut nantinya akan menggunakan dataset yang didapatkan dari sebuah jurnal. Dataset tersebut nantinya akan dilakukan pengujian terhadap pengembalian citra yang *blur*, penghilangan kabut, dan penghilangan derau.

Pengujian yang dilakukan nantinya dilakukan dalam beberapa skenario, yaitu citra yang *blur*, citra yang terdapat kabut, citra yang terdapat derau, citra yang *blur* ditambah dengan kabut, citra yang kabut ditambah derau, dan terhadap ketiga masalah tersebut secara bersamaan. Skenario tersebut nantinya akan diuji dengan melakukan pengujian terhadap dua metode, yaitu AFF (*Adaptive Fuzzy Filter*) dan *dark channel* (lapisan kegelapan).

Tingkat keberhasilan pada penelitian dilakukan menggunakan MSE (*Mean Square Error*) dan PSNR (*Peak Signal Noise Ratio*). Sesuai dengan yang dijelaskan (Welstead, 1999), rentang PSNR yang baik untuk citra dan video antara 30 dB (desibel) sampai dengan 50 dB (desibel).

UNIVERSITAS  
MIKROSKIL



Gambar II-16 Kerangka Konsep

# UNIVERSITAS MIKROSKIL