

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian tentang klasifikasi citra buah kopi sudah pernah dilakukan Narko dan Andono (2015) dengan melakukan penelitian mengenai analisis CBIR (Content Based Image Retrieval) untuk menentukan tingkat kematangan biji kopi jenis robusta. Penelitian ini menggunakan CBIR untuk mengekstraksi ciri citra buah kopi dan menggunakan jarak *Euclidean* untuk melakukan klasifikasi. Tingkat akurasi yang dihasilkan dalam klasifikasi buah kopi untuk 20 citra uji baru didapat 90%. Dimana hanya menggunakan satu citra sebagai data acuan.

Maria (2013) melakukan penelitian menggunakan pengolahan citra dan *fuzzy logic* untuk mengklasifikasi kualitas biji kopi. Penelitian ini menggunakan logika *Fuzzy Quantization* (FLVQ) untuk melakukan klasifikasi dan dihasilkan tingkat akurasi 92,86% untuk vector masukan 6 (enam) variable dan 85,71% dengan 9 (sembilan) variabel.

Wang Zhi-hai, ddk., (2003) melakukan penelitian tentang pendekatan BPR (Pengenalan Pola Biomimetik) terhadap pengenalan wajah. Pada penelitian ini, Jaringan Saraf Tiruan (JST) bobot dua atau yang lebih dikenal dengan *Hyper Sausage Neuron* (HSNN) digunakan sebagai representasi dari metode BPR dan ekstraksi ciri wajah yang didasarkan pada perbedaan dan penempatan posisi mata digunakan sebagai metode pembangkit ciri dari citra

wajah yang akan diidentifikasi. Jumlah sampel pelatihan yang digunakan sebanyak 91 sampel dengan tingkat keberhasilan yang didapatkan dari pengenalan wajah menggunakan metode BPR sebesar 97% dan tingkat kegagalannya hanya sebesar 3% dari 226 sampel yang diujikan.

Pada penelitian yang dilakukan oleh Samsuryadi dan Primanita (2015) menunjukkan bahwa HSNN dapat mengenali citra telinga manusia untuk mengamankan informasi. Penelitian ini menerapkan *Geometrical Moment Invariant* (GMI) sebagai ekstraksi ciri dan HSNN untuk melakukan pengenalan. Dengan data *training* sampel 75 diperoleh tingkat akurasi 97,33% dan 0,023 detik dari masing – masing gambar untuk diproses.

Dari penelitian yang dilakukan diatas menunjukkan bahwa dengan menggunakan metode HSNN tingkat akurasi yang dihasilkan lebih tinggi dalam melakukan pengenalan terhadap sampel yang terlatih maupun yang tidak terlatih dan memiliki waktu pengenalan yang lebih singkat.

2.2 Buah Kopi

Buah kopi merupakan salah satu komoditas perkebunan bernilai tinggi dan termasuk kedalam kategori industri. Indonesia sendiri menempati peringkat empat dunia dalam ekspor buah kopi pada tahun 2015 yang dicatat oleh *International Coffee Organization*. Pada praktik dilapangan untuk melakukan penyeleksian buah kopi yang matang dilakukan secara manual oleh manusia. Namun metode ini dinilai kurang optimal dan memiliki

kemungkinan ada perbedaan kualitas kopi yang dihasilkan apabila penyeleksian dilakukan dalam kapasitas yang besar.

Sebuah pohon kopi yang berkualitas dalam satu pohon dapat menghasilkan kurang lebih 5000 buah kopi dibagi dengan beberapa gugus tiap tandan. Tiap gugus memiliki buah 20 sampai 50 buah. Tingkat kematangan buah kopi dapat dilihat secara langsung dari citra warna buah kopi itu sendiri, yaitu buah kopi mentah warna hijau, berubah menjadi hijau tua, kemudian menguningsetengah matang dan buah kopi matang berwarna merah atau merah tua. Untuk ukuran buah kopi jenis arabika memiliki panjang antara 12 – 18 mm dan buah kopi jenis robusta 7 – 16 mm (Panggabean, 2011).

Karakteristik buah kopi matang pada citra memiliki perbedaan yang sangat terlihat sehingga dapat mudah dibedakan. Namun pengelola perkebunan kopi pada musim panen seringkali memasukan kopi yang masih berwarna hijau tetapi memiliki tekstur yang keras atau setengah matang (Panggabean, 2011). Namun dari pemilahan buah kopi yang seperti ini memiliki kualitas kurang baik dibandingkan dengan buah yang dipetik berwarna merah dan merah kehitaman. Citra buah kopi matang dapat dilihat pada gambarII-1.



Gambar II-1. Buah Kopi Matang dengan (a) Matang, (b) Mentah

2.3 Pengolahan Citra

Pengolahan citra adalah modifikasi atau proses mempresentasikan suatu objek secara kontinuis. Citra didefinisikan sebagai dua fungsi yaitu $f(x,y)$, dimana x dan y merupakan kordinat special dan f merupakan sepasang kordinat yang disebut intesitas atau tingkat keabuan (Gonzales dan Woods, 2002).

Tujuan dari pengolahan citra adalah untuk modifikasi citra melalui piksel sehingga hasil citra baru sesuai kebutuhan. Dimana proses citra dapat dilakukan dengan media seperti kamera digital, scanner, dan perangkat input gambar lainnya. Proses umum pengolahan citra digital dapat dilihat pada gambar II-2.



Gambar II-2. Skema Umum Pengolahan Citra Digital

2.4 Ekstraksi Ciri *Fuzzy Color Histogram*

Warna adalah ciri yang paling dasar untuk mendiskriminasikan metadata yang terdapat pada suatu citra. Informasi-informasi pada warna juga dapat berperanpenting dalam penelitian ini karena kematangan buah kopi ditentukan dari warna. Maka penulis menggunakan *Fuzzy Color Histogram* (FCH) sebagai pemroses untuk mempresentasikan informasi warna kedalam bentuk histogram (Herdiyeni, Buono, & Noorniawati, 2007). Color Histogram banyak digunakan dalam proses pengindeksan pada bidang *Image Retrieval*,

Video Retrieval, dan *Object Tracking*. FCH dapat mempertimbangkan persamaan dari warna-warna tiap piksel yang dialokasikan ke semua bin histogram melalui Fuzzy set membership function (Han & Ma, 2002). Oleh karena itu, FCH memiliki banyak bin yang dapat menampung tiap-tiap warna yang berbeda sehingga dapat dilihat warna yang sering muncul pada suatu citra. Pada FCH, satu warna bisa masuk ke dalam dua atau lebih bin histogram dengan derajat keanggotaan yang berbeda. Dalam penelitian ini, peneliti menggunakan ruang warna CIELAB dimana Ruang warna tersebut mendekati bagaimana manusia menerima warna melewati penglihatannya. Perubahan warna dari RGB ke CIELAB menggunakan operasi tertentu. Menurut (Zhao et al., 2014), Perubahan ruang warna RGB tidak dapat langsung diubah menjadi CIELAB.

Perubahan memerlukan dua proses, yaitu:

1. Mengubah ruang warna RGB menjadi Nilai Tristimulus XYZ;
2. Mengubah nilai Tristimulus XYZ menjadi CIELAB.

Pada FCH, tiap warna dipresentasikan dengan himpunan Fuzzy. Hubungan warna dimodelkan dengan fungsi keanggotaan terhadap himpunan fuzzy. Pada FCH, citra dapat diartikan sebagai $F(I) = [f_1, f_2, \dots, f_n]$ dimana:

$$f_i = \sum_{j=1}^N \mu_{ij} P_j = \frac{1}{N} \sum_{j=1}^N \mu_{ij} \quad (\text{II.1})$$

Dimana:

P_j = Nilai probabilitas tiap piksel yang terdapat pada citra

N = Jumlah piksel dari citra

μ_{ij} = Nilai keanggotaan dari piksel j di dalam piksel ke i pada bin

warna (Han & Ma, 2002)

Model warna fuzzy ini memungkinkan pembesaran pengaruh warna yang diberikan ke warna Lainnya (Zhang & Zhang, 2004) . Pada FCH, setiap warna yang didapati pada citra akan menimbulkan pengaruh pada seluruh kuantisasi warna berdasarkan kemiripan warna dengan warna citra tersebut.

Pada proses FCH, dilakukan 5 tahapan Proses (Joshi, Bade, & Joshi, 2015), yaitu:

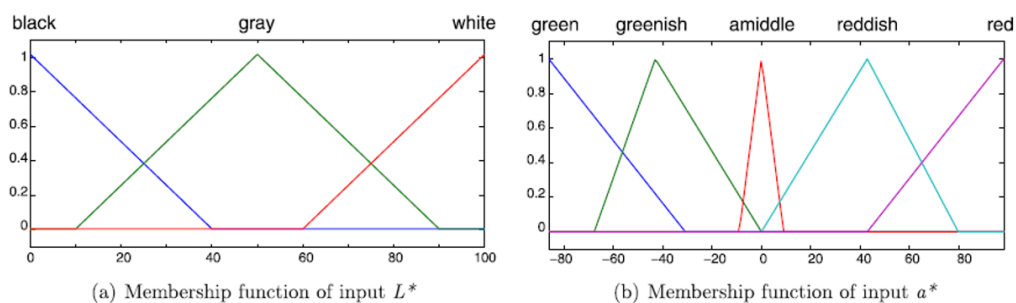
1. Mengkonversi RGB ke $L^*a^*b^*$ komponen warna fuzzy. L^* melambangkan Luminance, a^* melambangkan kehijauan dan kemerahan, b^* melambangkan kebiruan dan kekuningan. Komponen a^* dan b^* memiliki nilai yang lebih tinggi daripada komponen L^*
2. Fuzzifikasi (Proses mengubah suatu masukan dari bentuk tegas mejadi variabel linguistik (fuzzy) yang biasanya disajikan dalam bentuk himpunan himpunan fuzzy dengan suatu fungsi keanggotaan masing-masing) tiap pixel menjadi $L^*a^*b^*$ menggunakan fungsi triangular membership ('trimf'). Ambil inputan dan menentukan nilai derajat dimana tiap inputan merupakan milik tiap-tiap nilai fuzzy. Fungsi keanggotaan L^*

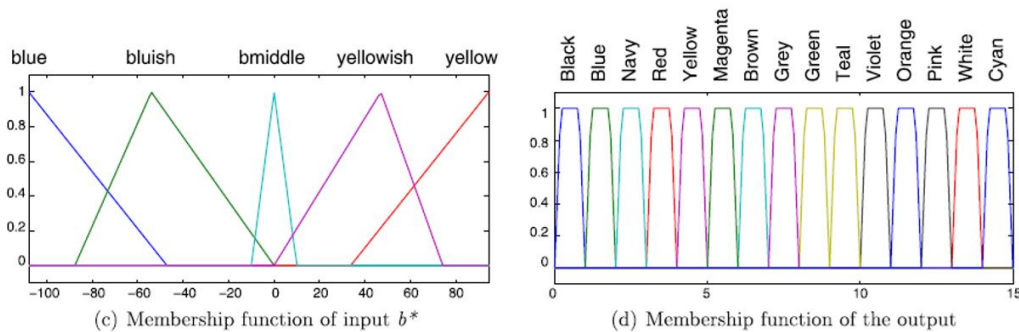
terbagi menjadi 3 bagian yaitu Hitam, Abu, dan Putih. a^* menjadi 5 bagian yaitu merah, kemerahan, a_middle , kehijauan, dan hijau. b^* menjadi 5 bagian yaitu Biru, Kebiruan, b_midde , kekuningan, kuning.

3. Pengumpulan, lalu aturan fuzzy tersebut dievaluasi dan hasilnya dikombinasi dengan operator *MAX aggregation*. Hal ini dilakukan untuk mengubah nilai *aggregate fuzzy* menjadi suatu bentuk tegas. Pengumpulan merupakan proses dari penyatuan dari keluaran semua aturan.

4. *Defuzzification* (langkah terakhir dalam suatu sistem logika fuzzy) dimana tujuannya adalah mengkonversi setiap hasil dari *inference engine* yang diekspresikan dalam bentuk fuzzy set kesuatu bilangan real. hasil konversi tersebut merupakan aksi yang diambil oleh sistem kembali logika fuzzy) digunakan untuk merubah ke bentuk tegas.

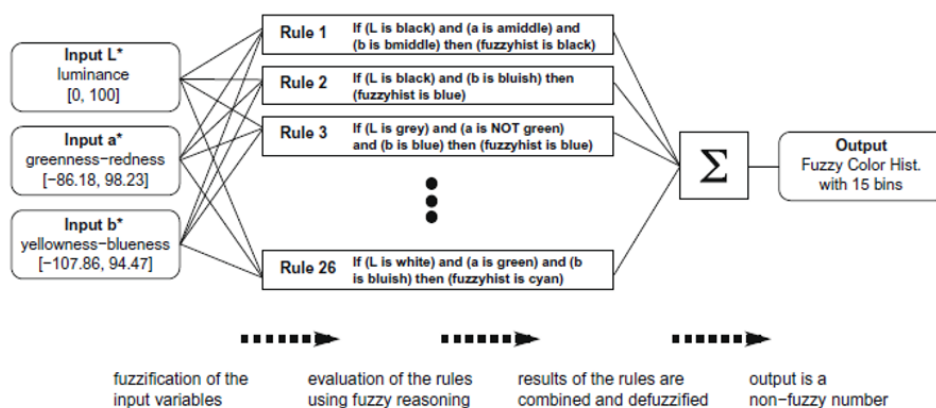
5. Mengkalkulasi *Fuzzy Histogram* dan *Histogram error*. Nilai histogram eror didapatkan dengan cara melakukan peninjauan terhadap citra kueri dan citra database, dilakukan rotasi, pencerminan terhadap citra kueri. Jika terdapat perbedaan hasil histogram pada citra yang sudah di transformasi tersebut, maka terdapat histogram eror yang menyebabkan histogram tersebut kurang baik. Hal ini dilakukan untuk tinjauan hasil ekstraksi ciri yang akurat.





Gambar II-3. *Fuzzy Membership Function* untuk input($L^*a^*b^*$) dan Output

(Küçüktunç, Güdükbay, & Ulusoy, 2010)



Gambar II-4. Struktur *Fuzzy Color Histogram* (Küçüktunç et al., 2010)

Pada gambar II-3 menjelaskan tentang rentang nilai masing-masing inputan L^* , a^* , dan b^* . Pada gambar tersebut, gambaran Fuzzy Membership Function dijabarkan menjadi keluaran dengan 15 bins warna. Pada Gambar II-3, dijelaskan proses bagaimana ciri masukan L^* , a^* , dan b^* akan menjadi keluaran dengan warna tertentu dengan rentang 15 bin warna dengan aturan fuzzy yang terdapat pada Gambar II-5.


```

If (L is black) and (a is amiddle) and (b is bmiddle) then (fuzzyhist is black)
If (L is black) and (b is blueish) then (fuzzyhist is blue)
If (L is grey) and (a is NOT green) and (b is blue) then (fuzzyhist is blue)
If (L is white) and (a is amiddle) and (b is blueish) then (fuzzyhist is blue)
If (L is white) and (a is greenish) and (b is blueish) then (fuzzyhist is blue)
If (L is black) and (a is reddish) and (b is blue) then (fuzzyhist is navy)
If (L is grey) and (a is red) and (b is NOT blue) then (fuzzyhist is red)
If (L is grey) and (a is reddish) and (b is bmiddle) then (fuzzyhist is red)
If (L is black) and (a is reddish) and (b is yellowish) then (fuzzyhist is red)
If (L is grey) and (a is reddish) and (b is yellow) then (fuzzyhist is yellow)
If (L is white) and (a is amiddle) and (b is yellow) then (fuzzyhist is yellow)
If (L is white) and (a is greenish) and (b is yellow) then (fuzzyhist is yellow)
If (L is grey) and (a is reddish) and (b is blueish) then (fuzzyhist is magenta)
If (L is white) and (a is reddish) and (b is blueish) then (fuzzyhist is magenta)
If (L is grey) and (a is amiddle) and (b is yellowish) then (fuzzyhist is brown)
If (L is white) and (a is reddish) and (b is yellow) then (fuzzyhist is brown)
If (L is grey) and (a is amiddle) and (b is bmiddle) then (fuzzyhist is grey)
If (L is grey) and (a is greenish) and (b is yellow) then (fuzzyhist is green)
If (L is white) and (a is green) and (b is yellowish) then (fuzzyhist is green)
If (L is grey) and (a is greenish) and (b is bmiddle) then (fuzzyhist is teal)
If (L is grey) and (a is green) and (b is blue) then (fuzzyhist is violet)
If (L is white) and (a is red) and (b is yellow) then (fuzzyhist is orange)
If (L is white) and (a is reddish) and (b is bmiddle) then (fuzzyhist is pink)
If (L is white) and (a is amiddle) and (b is bmiddle) then (fuzzyhist is white)
If (L is white) and (a is greenish) and (b is bmiddle) then (fuzzyhist is cyan)
If (L is white) and (a is green) and (b is bluish) then (fuzzyhist is cyan)

```

Gambar II-5. *Fuzzy Rules for L*a*b** (Küçüktunç et al., 2010)

2.5 *Hyper Sausage Neuron Network*

Hyper Sausage Neuron Network (HSNN) diajukan sebagai representasi dari *Biomimetic Pattern Recognition* (BPR). Ini digunakan sebagai kesatuan jaringan saraf dari hyper sausage per kelas (Shojue dan Xingtao, 2004, Samsuryadi, 2013). BPR sendiri dikenalkan oleh Wang Shoujue pada tahun 2002 sebagai pengenalan pola baru. Konsep dasar dari BPR adalah mengenali sampel satu kelas, dan bukan mengklasifikasikannya kedalam banyak kelas seperti yang terdapat dalam *Statistical Pattern Recognition* (SPR) (Wu, dkk., 2011).

BPR sendiri menggunakan *Principle of Homology Continuity* (PHC) yang berarti bahwa perbedaan antara dua sampel pada kelas yang sama harus berubah secara bertahap. Perubahan barisan bertahap ini harus ada di antara kedua sampel tersebut. Setiap sampel dalam barisan tersebut harus dipetakan secara berkelanjutan. Prinsip berkelanjutan antar sampel sejenis pada ruang ciri ini dinamakan *principle of homology continuity* (Shou-jue, dkk, 2003).

PHC dapat digambarkan secara matematis sebagai berikut. Misalkan, A merupakan suatu kumpulan sampel sejenis pada kelas A . Bila x dan y adalah dua titik yang berada pada kelas A dan terdapat bilangan kecil ($\varepsilon > 0$), maka akan ada himpunan B yang memenuhi persamaan (II.2).

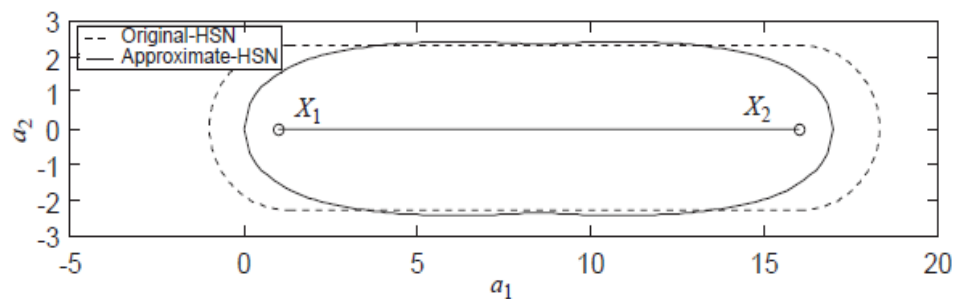
$$B = \{x_1, x_2, \dots, x_n \mid x_1 = x, x_n = y, n \in N, \rho(x_m, x_{m+1}) < \varepsilon, \varepsilon > 0, 1 \leq m \leq n - 1, m \in N\}, B \subset A \quad (\text{II.2})$$

Dimana $\rho(x_m, x_{m+1})$ adalah jarak antara sampel x_m dan x_{m+1} .

Melalui PCH, diketahui bahwa semua sampel sejenis dari kelas A harus terdistribusi pada daerah terkait pada ruang ciri R^n . Daerah ini dinotasikan dengan P . Dengan mempertimbangkan gangguan kecil, semua sampel yang berhampiran dengan P harus dipertimbangkan sebagai sampel kelas A . Sehingga tujuan BPR adalah mencari pelingkupan yang sesuai pada ruang ciri kelas A (Shou-jue, dkk., 2003). Pelingkupan sempurna dari himpunan P pada kelas A yang dinotasikan dengan P_a dinyatakan oleh Persamaan (II.3).

$$P_a = \{x \mid \rho(x, y) \leq k, y \in A, x \in R^n\} \quad (\text{II.3})$$

Dimana k sebagai jarak ambang batas konstanta (Wang, dkk., 2012). Dalam ruang ciri n -dimensi, P_a memiliki n -dimensi yang kompleks, yang memisahkan jarak keseluruhan ruang menjadi dua bagian, yaitu satu kelas sebagai kelas A dan yang lainnya bukan. Oleh karena itu, pengenalan pola dengan metode HSNN dilakukan dengan menentukan pemetaan dalam ruang ciri dari suatu objek yang melingkupi himpunan P_a atau bukan (Zhai, dkk., 2009).



Gambar II-6. Original HSNN dan yang mendekati HSNN dalam bentuk 2D(Shoujue,dkk., 2004)

Misalkan jumlah sampel pelatihan setiap citra buah kopi adalah N , maka himpunan pelatihan sampel dinyatakan dengan Persamaan (II.4):

$$S = \{x \mid x = S_1, S_2, \dots, S_n\} \quad (\text{II.4})$$

Agar dapat menggunakan jaringan syaraf untuk melingkupi P_a , gunakan beberapa garis segmen baru yang dinotasikan dengan B untuk mendekati A . Maka, pelingkupannya adalah P_b . Dikarenakan jumlah sampel

pelatihannya ada sejumlah N , maka akan ada $N-1$ garis yang akan mendekati

A. Maka B dapat dinyatakan dengan Persamaan (II.5).

$$B = \bigcup_{i=1}^{n-1} B_i$$

$$B_i = \{x | x = aS_i + (1 - a)S_{i+1}, a \in [0,1], S_i \in S, x \in R^d\} \quad (\text{II.5})$$

Pelengkupan setiap jaringan syarafnya dapat direpresentasikan dengan persamaan (II.6).

$$P_b = \bigcup_{i=1}^{n-1} P_i \quad (\text{II.6})$$

$$P_i = \{x | x = \min(\rho(x, y)) \leq Th, y \in B_i, x \in R^d\} \quad (\text{II.7})$$

Dimana P_i dilingkupi oleh struktur neuron pada Persamaan (II.19).

$$y_i = f[\phi(S_i, S_{i+1}, x)] \quad (\text{II.8})$$

$$\phi(S_i, S_{i+1}, x) = \min_{y \in \{z | z = as_i + (1 - a)S_{i+1}, a \in [1,0]\}}(\rho(x, y)) \quad (\text{II.9})$$

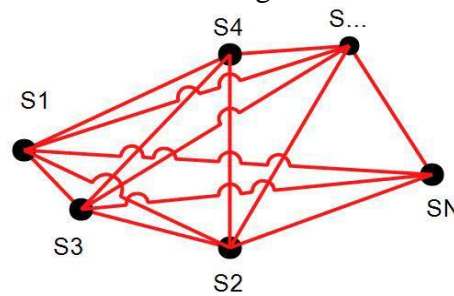
2.5.1 Algoritma Pelengkupan

Dalam BPR, proses pelatihan dan pengenalannya berbeda dengan JST. Dimana BPR dilakukan bersamaan sedangkan JST hanya dengan satu bobot. Selain itu JST menggunakan bobot acak sebagai pengetahuan awal, BPR

menggunakan sampelnya sendiri sebagai pengetahuan awal untuk melakukan pelatihan sekaligus pengenalan.

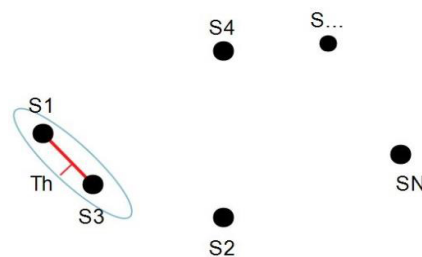
Misalkan himpunan sampel dalam suatu kelas dinyatakan dengan S , dan jumlah sampelnya sebanyak N . Dalam himpunan pelatihan ini akan dibangun sebanyak $N-1$ pelingkupan jaringan HSNN (Hyper Sausage Neuron) dan membentuk sebuah closed set dalam ruang ciri yang melingkupi kelas tersebut. Proses pelingkupannya adalah sebagai berikut:

Langkah 1: Cari dua sampel (i dan j) diantara N sampel yang memiliki jarak terdekat menggunakan Jarak Euclidean dengan Persamaan (II.14).



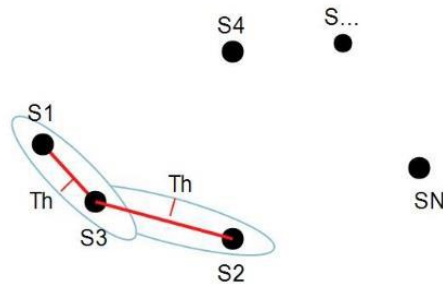
Gambar II-7. Langkah 1 HSNN

Langkah 2: Bangun sebuah jaringan HSNN menggunakan sampel i dan j tersebut menggunakan Persamaan (II.6) dan (II.7). Kemudian cek apakah ada sampel yang jaraknya kurang dari nilai Threshold (TH), Persamaan (II-13). Jika ada, maka sampel ini dianggap tidak pantas dan harus diabaikan.



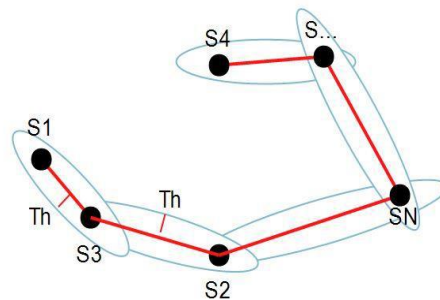
Gambar II-8. Langkah HSNN 2

Langkah 3: Setelah mengabaikan sampel yang tidak pantas, hitung lagi jarak antara sampel yang tersisa dan pilih sebuah sampel (sampel k) yang memiliki jarak terdekat antara sampel i atau sampel j.



Gambar II-9. Langkah HSNN 3

Langkah 4: Ulangi langkah 2 dan langkah 3 dengan sampel j dan sampel k hingga sebuah pelingkupan rantai HSNN dari kelas tersebut terbentuk.



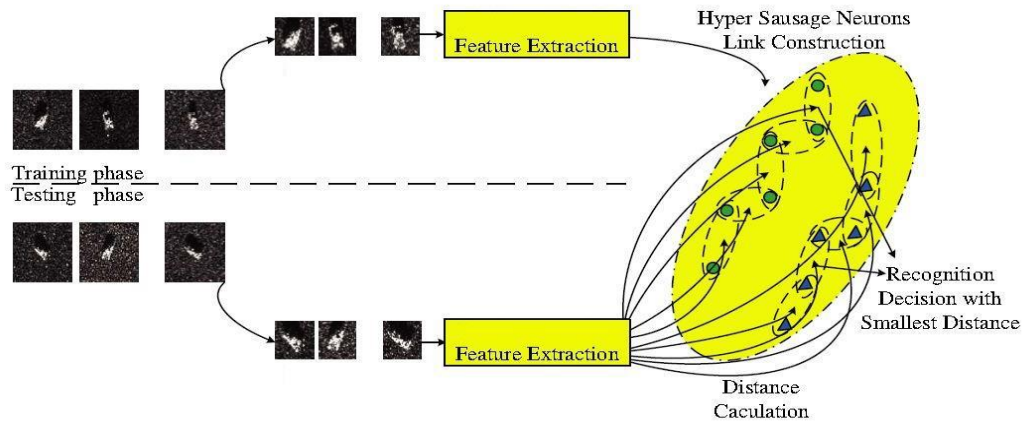
Gambar II-10. Langkah 4 HSNN

Langkah 5: Tetap ulangi langkah 1 sampai 4 hingga semua kelas dalam himpunan pelatihan membentuk rantai-rantai HSNN.

Pada proses klasifikasi, kita hanya perlu melakukan pengecekan apakah sampel tersebut berada dalam pelingkupan kelas HSNN dengan menghitung jarak terkecil antara sampel yang akan diklasifikasi dan sampel yang terdapat dalam database menggunakan Persamaan (II.13) dan (II.14).

Jika ya, maka sampel tersebut dapat dikatakan merupakan kelas tersebut.

Adapun arsitekturnya dapat dilihat pada Gambar II-11.



Gambar II-11. Arsitektur HSNN (Zhai, dkk., 2013)

2.5.2 Nilai Ambang (Threshold)

Parameter *threshold* (Th) dalam pembentukan ruang pelingkupan HSNN menentukan tingkat akurasi dan mengeneralisasi algoritma. Samsuryadi (2013) mengusulkan dua buah Th yang didasarkan pada *Mean Absolute Error* (MAE) yang dinyatakan dalam Persamaan (II.10).

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - r_i| \quad (II.10)$$

Keterangan :

MAE : *Mean Absolute Error*;

n : jumlah citra;

x_i : citra itu sendiri;

r_i : referensi citra.

Dimana citra pertama adalah referensi citranya.

$$Th = \frac{2}{n(n-1)} \sum_{j \in A} \sum_{i \in A} d(S_i, S_j) \quad (\text{II.11})$$

Dimana $i > j$, dan n merupakan jumlah sampel. Dimana Persamaan (II.12) dimodifikasi menggunakan Persamaan (II.11) dihasilkan Threshold pertama dari Persamaan (II.12).

$$Th_1 = \frac{2}{n(n-1)} \sum_{i=1, i \neq k}^n \left(\frac{1}{d} \sum_{j=1}^d |S_{ij} - S_{kj}| \right) \quad (\text{II.12})$$

Keterangan :

Th : Threshold pertama;

n : jumlah sampel citra dalam himpunan S ;

d : jumlah ciri citra sampel;

S : $S = \{S_1, S_2, \dots, S_n\}$ sampel citra pada himpunan pembelajaran dan setiap sampel citra dinotasikan dengan $S = \{S_{i1}, S_{i2}, \dots, S_{id}\}$;

S_{ij} : ciri ke- j dari sampel ke- i ;

S_{kj} : ciri ke- j dari sampel ke- k .

Threshold kedua menggunakan perbedaan dari setiap ciri dari sampel (S_{ij}) untuk mendapatkan nilai rata-rata dari semua ciri sampel setiap himpunan S , yang terbagi menjadi (nd) yang ditunjukkan pada Persamaan (II.13).

$$Th_2 = \frac{1}{nd} \sum_{i=1}^n \sum_{j=1}^d |S_{ij} - \bar{S}|, \text{ dengan } \bar{S} = \frac{1}{nd} \sum_{i=1}^n \sum_{j=1}^d S_{ij} \quad (\text{II.13})$$

Dimana, n merupakan jumlah sampel citra dalam sebuah himpunan (S), d merupakan jumlah ciri sampel citra, dan S_{ij} merupakan ciri ke- j dari sampel ke- i .

2.5.3 Jarak Euclidean

Jarak Euclidean (*Euclidean Distance*) merupakan metrika yang paling sering digunakan untuk menghitung kesamaan 2 (dua) vektor. Jarak Euclidean adalah akar dari jumlah selisih kuadrat antara 2 (dua) vektor. Rumus dari Jarak Euclidean dinyatakan oleh Persamaan (II.14).

$$\rho_{ij} = \sqrt{\sum_{k=1}^n ((x_{ik} - x_{jk})^2)} \quad (\text{II.14})$$

Keterangan :

n = banyak ciri;

k = menyatakan ciri ke- k ;

ρ = jarak antara 2 vektor;

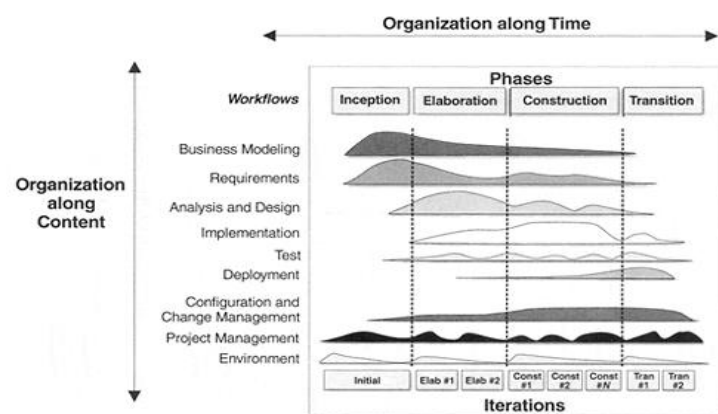
x_{ik} = sampel ke – i dari ciri ke – k ;

x_{jk} = sampel ke – j dari ciri ke – k .

2.6 *Rational Unified Process (RUP)*

Metode RUP merupakan paradigma pengembangan perangkat lunak yang bersifat dinamis dengan pemodelan menggunakan bahasa pemodelan *Unified Modeling Language (UML)*. Metode RUP dalam struktur alur kegiatannya dapat melakukan perulangan ke fase sebelumnya tanpa harus menunggu seluruh alur kegiatan dalam proses selesai dilakukan. Dengan demikian, alur kegiatan pengembangan perangkat lunak metode RUP dinamakan juga dengan struktur dinamis (*dynamic structure*) (Kruchten, 2000).

Dalam pengembangan perangkat lunak dengan metode RUP memiliki tiga karakteristik, yaitu *use-case driven*, *architecture-centric*, dan *iterative and incremental*. *Use case driven* merupakan pemodelan perangkat lunak yang dikembangkan berlandaskan pada *use case*. *Use case* berfungsi untuk menjembatani komunikasi antara pengguna dengan perangkat lunak. Sehingga, *use case* mampu menyajikan apa yang seharusnya perangkat lunak lakukan dari sudut pandang pengguna. *Architecture centric* merupakan komposisi dari struktur elemen-elemen yang memiliki hubungan keterkaitan satu dengan yang lainnya, baik antar elemen maupun elemen dengan perangkat lunak. Selain itu, arsitektur dapat juga menjelaskan fungsionalitas dan unjuk kerja (*performance*) dari sistem. Selanjutnya, *iterative* merupakan perulangan yang dilakukan pada fase-fase tertentu untuk memenuhi kebutuhan dari perangkat lunak. Sedangkan, *incremental* merupakan jumlah dari perulangan yang dilakukan pada fase-fase tertentu. Penjelasan di atas dapat digambarkan pada Gambar II-11.



Gambar II-12. Arsitektur RUP

(Sumber: Kruchten, 2000)

Pada gambar (II-12) dapat dilihat pada metode RUP memiliki 2 dimensi tahapan. Pada dimensi tahapan pertama digambarkan secara horizontal dan menjelaskan aspek proses pengembangan perangkat lunak yaitu; siklus, fase, iterasi, dan *milestones*. Dimensi ini terdiri dari beberapa tahapan yaitu; insepisi (*inception*), elaborasi (*elaboration*), konstruksi (*construction*), dan transisi (*transition*). Dimensi kedua ditunjukkan melalui garis vertikal. Dimensi ini menjelaskan aspek statis dari metode RUP yang meliputi; aktivitas, alur kerja, dan pekerja yang bertanggung jawab terhadap suatu proses. Alur kerja pada dimensi kedua ini terdiri atas; *business modeling, requirement, analysis and design, implementation, test, deployment, configuration and change management, project management* dan *environment*.

Dalam metode RUP ada 4 (empat) fase yang harus dicapai dalam pengembangan dan dokumentasi perangkat lunak. Tahapan-tahapan tersebut adalah :

1. Fase insepisi adalah tahap awal dalam proses pengembangan perangkat lunak pada metode RUP. Tahap ini berfokus pada permodelan bisnis. Permodelan bisnis yang dimaksud adalah pendefinisian masalah, batasan ruang lingkup proyek, dan membuat *business case*.
2. Fase elaborasi merupakan tahap kedua. Tujuan dari fase *elaboration* adalah untuk menetapkan arsitektur dasar dari sistem untuk menyediakan dasar yang stabil sebagian besar dari desain dan pelaksanaan upaya dalam tahap konstruksi.

3. Fase konstruksi merupakan tahapan yang mengimplementasikan kebutuhan perangkat lunak ke dalam modul-modul dalam bentuk kode program. Selain itu, aktivitas pada fase ini juga berfokus terhadap pengujian perangkat lunak. Sehingga, perangkat lunak yang dihasilkan berkualitas dan sesuai dengan kebutuhan.

4. Fase transisi adalah tahapan terakhir pada metode RUP. Aktivitas yang paling dominan dilakukan pada fase ini adalah merilis versi beta dari perangkat lunak, melakukan distribusi perangkat lunak, mempersiapkan lingkungan operasi perangkat lunak dan membuat buku panduan perangkat lunak serta melakukan pelatihan kepada user.