BAB II

LANDASAN TEORI

Pada bab ini akan dijelaskan tentang berbagai teori yang digunakan untuk melakukan penelitian ini. Karena pada penelitian ini menggunakan pengukuran *fuzzy*, maka akan dijelaskan logika *fuzzy*, himpunan *fuzzy*, pengukuran *fuzzy*, *fuzzy* integral, deteksi objek wajah, dan *boosting*. Untuk mendeteksi wajah akan dijelaskan tentang deteksi objek wajah dan algoritma boosting yang digunakan dalam proses pendeteksian wajah.

2.1 LOGIKA FUZZY

Logika *Fuzzy* adalah logika yang berbasiskan pada teori himpunan *fuzzy* dan diperkenalkan oleh Lotfi Zadeh (1965) [1]. Pada logika *fuzzy*, terdapat proses pemetaan dari suatu ruang input ke dalam suatu ruang output [11]. Logika *fuzzy* berguna sebagai fondasi untuk *approximate reasoning* dengan preposisi yang tidak pasti [1]. Fokus utama dari logika *fuzzy* adalah pada bahasa natural, dimana *approximate reasoning* dengan ketidakpastian sering terjadi. Logika *fuzzy* memungkinkan penggunaan predikat *fuzzy* (mahal, tua, muda, jarang, berbahaya, dan seterusnya), kuantifier *fuzzy* (banyak, sedikit, hampir semua, biasanya, mirip, dan sebagainya), nilai kebenaran *fuzzy* (hampir benar, kurang benar, dan sebagainya), dan *fuzzy modifiers* (sepertinya, hampir mustahil, dan sebagainya). Logika *fuzzy* terdiri dari tiga operator, yaitu *fuzzy negation*, t-norm, dan s-norm yang akan dijelaskan lebih lanjut.

2.1.1 Fuzzy Negation

Fuzzy negation adalah operasi negasi yang digunakan di logika *fuzzy* dan dituliskan dengan notasi $^{(n)}$. Berdasarkan definisi, *fuzzy negation* adalah sebuah fungsi $^{(n)}$: [0,1] → [1,0] yang memenuhi sifat – sifat berikut:

1. $0^{(n)} = 1$
2. $x_1^{(n)} > x_2^{(n)}$ jika $x_1 < x_2$
3. $(x^{(n)})^{(n)} = x$

2.1.2 T-Norm

T-norm adalah operasi konjungsi yang digunakan di logika *fuzzy*. Pada laporan ini, t-norm dituliskan dengan simbol T. Selain dapat melakukan operasi konjungsi di logika *fuzzy*, t-norm juga dapat digunakan sebagai basis untuk operator agregasi pada operasi himpunan *fuzzy*. Berdasarkan definisi, t-norm adalah sebuah fungsi T: [0,1] x [0,1] → [1,0] yang memenuhi sifat – sifat berikut:

1. $T(x,0) = 0$ dan $T(x,1) = x$
2. $T(x_1, x_2) = T(x_2, x_1)$
3. $T(x_1, T(x_2, x_3)) = T(T(x_1, x_2), x_3)$
4. $T(x_1, x_3) \leq T(x_2, x_3)$ jika $x_1 \leq x_2$

2.2.3 S-norm

S-norm (juga dikenal sebagai T-conorm) adalah operasi disjungsi yang digunakan di logika *fuzzy*. Pada laporan ini, s-norm dituliskan dengan simbol ⊥.

Berdasarkan t-norm, S-norm dapat didefinisikan sebagai $\perp(a,b) = 1 - T(1-a , 1-b))$. Berdasarkan definisi, s-norm adalah sebuah fungsi $\perp$: $[0,1] \times [0,1] \rightarrow [1,0]$ yang memenuhi sifat – sifat berikut:

1. $\perp(x, 0) = x$, $\perp(x, 1) = 1$

2. $\perp(x_1, x_2) = \perp(x_2, x_1)$

3. $\perp(x_1, \perp(x_2, x_3)) = \perp(\perp(x_1, x_2), x_3)$

4. $\perp(x_1, x_3) \leq \perp(x_2, x_3)$ jika $x_1 \leq x_2$

Alasan digunakannya Logika *fuzzy* adalah sebagai berikut [11]:

- Konsep logika *fuzzy* mudah dimengerti, karena konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
- Logika *fuzzy* sangat fleksibel
- Logika *fuzzy* memiliki toleransi terhadap data - data yang tidak tepat.
- Logika *fuzzy* mampu memodelkan fungsi – fungsi non linear yang sangat kompleks.
- Logika *fuzzy* dapat mengaplikasikan pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
- Logika *fuzzy* dapat bekerja sama dengan teknik – teknik kendali secara konvensional.
- Logika *fuzzy* didasarkan pada bahasa natural.

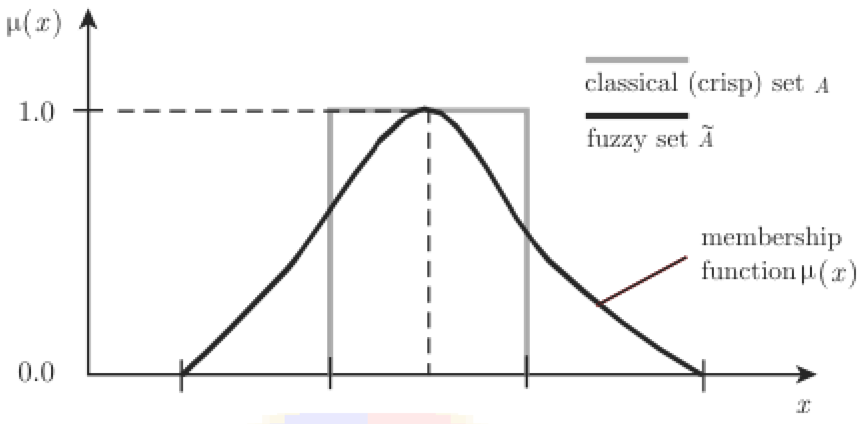
2.2 HIMPUNAN FUZZY

Himpunan *fuzzy* adalah himpunan yang setiap elemennya memiliki derajat keanggotaan tertentu [1]. Derajat keanggotaan tersebut ditentukan oleh fungsi keanggotaan (*membership function*). Fungsi keanggotaan, yang dikenal juga sebagai fungsi definisi atau fungsi karakteristik, adalah fungsi yang memetakan setiap elemen dari suatu himpunan ke interval nilai tertentu yang mengindikasikan derajat keanggotaan elemen tersebut. Hal ini berbeda dengan pengertian teori himpunan *crisp* yang memetakan setiap elemen dari himpunan hanya ke nilai nol atau satu.

Fungsi keanggotaan didefinisikan sebagai berikut: Jika X adalah himpunan semesta, maka fungsi keanggotaan μ_A (fungsi definisi / fungsi karakteristik A pada X) yang didefinisikan oleh himpunan *fuzzy* A memiliki ketentuan berikut:

$\mu_A : X \rightarrow [0,1]$

dimana [0,1] adalah interval bilangan real dari nol sampai dengan satu (inklusif). Jika $\mu_A(x)$ bernilai nol, berarti x bukan anggota dari himpunan *fuzzy* A. Nilai satu menunjukkan x adalah anggota penuh dari himpunan *fuzzy* A [16]. Sementara nilai antara nol sampai satu menunjukkan bahwa x merupakan anggota dari himpunan *fuzzy* A secara parsial.



Gambar 1. Perbandingan Fungsi Keanggotaan Himpunan *Fuzzy* Terhadap Himpunan Klasik

Himpunan *fuzzy* berguna untuk memodelkan suatu kelas konseptual atau label linguistic yang bergantung pada fungsi keanggotaannya.

2.3 PENGUKURAN FUZZY

Pengukuran *fuzzy* telah banyak diterapkan dalam bidang *image understanding* yang membutuhkan kombinasi informasi dari berbagai sumber [2]. Pendekatan yang seringkali dilakukan, setiap sumber informasi diberikan nilai kesesuaian (*grades of compatibility*), dan setiap informasi diberikan bobot dan saling dikombinasikan untuk keperluan tertentu. Pengukuran *fuzzy* adalah representasi dari ketidakpastian dengan fungsi yang didefinisikan sebagai berikut [1, 3]:

$g : P(X) \rightarrow [0,1]$

P(X) adalah *power set* dari *crisp set* X.

fungsi tersebut memetakan setiap *crisp subset* ke sebuah angka dalam interval [0,1]. Jika subset A ε P(X), maka g(A) merepresentasikan tingkat kepercayaan bahwa sebuah elemen yang sedang dipertimbangkan memang seharusnya milik dari subset A. Subset dari P(X) yang memiliki nilai g(A) tertinggi merupakan terkaan terbaik terhadap adanya elemen dari fakta-fakta yang sedang dipertanyakan[1]. Pada pengukuran *fuzzy* berlaku aksioma sebagai berikut:

1. $g(\emptyset) = 0$ dan $g(X) = 1$ (kondisi batasan)
2. untuk setiap A, B $\in$ P(X), jika $A \subseteq B$, maka $g(A) \leq g(B)$ (monotonik)
3. Untuk setiap rangkaian ($A_i \in P(X) \mid i \in N$) dari subset X, jika $A_1 < A_2 < \ldots$ atau $A_1 > A_2 > \ldots$ maka $\lim_{i \to \infty} g(A_i) = g(\lim_{i \to \infty} A_i)$ (kontinu)

Fungsi g disebut juga λ-*fuzzy measure*, ditulis dengan notasi g_λ jika memenuhi aksioma pengukuran *fuzzy* dan persamaan:

$$g_\lambda(A \cup B) = g_\lambda(A) + g_\lambda(B) + \lambda\, g_\lambda(A)\, g_\lambda(B)$$

Pada penerapannya, nilai λ perlu ditetapkan berdasarkan nilai dari *singleton* di X. Jika $X = \{x_1, x_2, \ldots, x_n\}$ dan $g^i = g_\lambda(\{x_i\})$ maka nilai $\{g^i : i = 1, 2, \ldots, n\}$ adalah nilai *fuzzy densities* dan diinterpretasikan sebagai bobot atau derajat kepentingan dari sebuah sumber informasi (*singleton*) [7].

Untuk $A \subset X$, dimana $A = \{x_{i1}, \ldots, x_{im}\}$ merupakan sekumpulan informasi, $g_\lambda(A)$ adalah bobot dari subset suatu kumpulan informasi [2].

$$g_\lambda(A) = \sum_{j=1}^{m} g^{i_j} + \lambda \sum_{j=1}^{m-1} \sum_{k=j+1}^{m} g^{i_j} g^{i_k} + \ldots + \lambda^{m-1} g^{i_1} g^{i_2} \ldots g^{i_m}$$

Untuk $\lambda \neq 0$ dan A = X, diperoleh

$$g_\lambda(X) = \frac{1}{\lambda} \left[\prod_{i=1}^{n} \left(1 + \lambda g^i\right) - 1 \right]$$

Nilai λ dapat ditemukan dengan mengganti $g_\lambda(X) = 1$ (berdasarkan definisi pengukuran *fuzzy*), sehingga diperoleh persamaan berikut:

$$1 + \lambda = \prod_{i=1}^{n}\left(1 + \lambda g^{i}\right)$$

Solusi unik dapat diperoleh jika $\lambda > -1$ dan $\lambda \neq 0$

Jika g adalah g_λ, nilai $g_\lambda(A_i)$ dihitung secara rekursif sebagai berikut [17]:

$g_\lambda(A_1) = g_\lambda(\{x_1\}) = g^1$

$g_\lambda(A_i) = g^i + g_\lambda(A_{i-1}) + \lambda g^i g_\lambda(A_{i-1})$ dimana $1 < i \leq n$ dan $A_i = x_1, \ldots, x_n$

g_λ merupakan derajat kepercayaan.

2.4 FUZZY INTEGRAL

Fuzzy integral dapat dilihat sebagai sebuah operator agregasi. Jika X adalah sebuah himpunan berisi sejumlah element dan fungsi h: X → [0,1], maka h(x) menyatakan derajat kepercayaan (*confidence value*) dari elemen x (keanggotaan dari sebuah data yang ditentukan oleh pengklasifikasi). Selanjutnya, *fuzzy* integral dari h terhadap E (subset dari X) dapat dihitung sebagai berikut [13]:

$$\int_E h(x) \circ g = \sup_{\alpha \in [0,1]} \left[\alpha \wedge g\left(E \cap H_\alpha\right)\right];$$

Dengan

$$H_\alpha = \{x \mid h(x) \geq \alpha\}.$$

Misalkan himpunan berhingga X = $\{x_1, x_2, \ldots, x_n\}$. Selanjutnya, kita perlu mengurutkan $h(x_i)$ secara menurun, lalu menghitung *fuzzy* integral Sugeno [13] sebagai berikut:

$$\int h(x) \circ g = \bigvee_{i=1}^{n} \left[h(x_i) \wedge g(H_i) \right] ;$$

Dengan

$$H_i = \{x_1, x_2, \cdots, x_i\}$$

Disamping *fuzzy* integral Sugeno, juga terdapat *fuzzy* integral Choquet [4]. Namun dalam penelitian ini, hanya digunakan *fuzzy* integral Sugeno yang mampu merefleksikan pemikiran *non-linier* manusia [5, 9] dan telah diterapkan dengan sukses untuk pada [2]. Pengukuran *fuzzy* dan *fuzzy* integral telah diterapkan pada bidang klasifikasi data, segmentasi citra, dan penyatuan citra.

2.5 DETEKSI OBJEK WAJAH

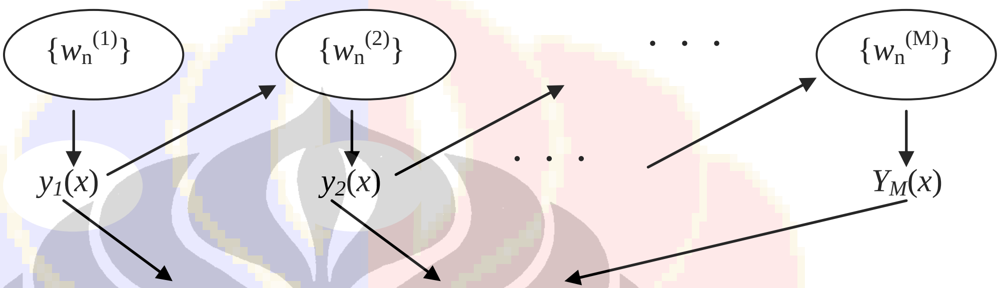
Deteksi objek wajah manusia merupakan proses untuk menentukan lokasi dan ukuran wajah manusia dari suatu citra [14]. Untuk dapat mendeteksi keberadaan wajah, sistem pendeteksi wajah harus dilatih terlebih dahulu agar bisa mengenali keberadaan wajah. Proses pelatihan melibatkan pengelompokan dua kelas citra. Citra yang mempunyai objek wajah dikelompokkan menjadi citra kelas positif, sedangkan citra yang tidak mempunyai objek manusia akan dikelompokkan ke dalam citra kelas negatif. Objek wajah digunakan pada tugas ini karena wajah dianggap mewakili identitas dari manusia.

Pada proses pelatihan, sistem pendeteksi wajah dilatih untuk membedakan citra yang mempunyai wajah manusia atau citra yang tidak mempunyai wajah manusia. Data pelatihan berupa citra kelas positif dan kelas negatif. Kelas positif merupakan citra yang mempunyai satu objek wajah, sedangkan kelas negatif merupakan citra yang tidak mempunyai objek wajah sama sekali. Wajah yang digunakan dalam proses pelatihan adalah wajah yang frontal. Proses pelatihan menggunakan metode *Boosting* yang akan dibahas lebih lanjut.

2.6 BOOSTING

Boosting adalah teknik yang handal untuk mengkombinasikan beberapa pengklasifikasi dasar (*multiple base classifiers*) untuk menghasilkan suatu *committee* (rata – rata prediksi dari beberapa pengklasifikasi dasar) yang memiliki kinerja yang lebih baik daripada pengklasifikasi dasar [12]. *Boosting* juga didefinisikan sebagai metode untuk menciptakan pengklasifikasi kuat (*strong classifier*) yang akurat dengan mengkombinasikan beberapa pengklasifikasi lemah (*weak classifier*). *Boosting* dikatakan memberikan hasil yang baik jika pengklasifikasi dasar atau pengklasifikasi lemah menghasilkan akurasi yang lebih baik daripada proses acak. Metode *Boosting* disebut juga sebagai algoritma AdaBoost (singkatan dari *Adaptive Boosting*).

Klasifikasi dua kelas dilakukan dengan menggunakan input vector x_1, x_2, x_3, … , x_N yang masing – masing mempunyai target t_1, t_2, t_3, … , t_N. Nilai $t_n \in \{-1,1\}$. Setiap data diberikan inisialisasi bobot w_n dengan nilai 1/N untuk setiap bobotnya. Pada proses pelatihan, pengklasifikasi dasar menggunakan bobot data untuk fungsi $y(x) \in \{-1,1\}$. Selanjutnya, untuk setiap tahapan, AdaBoost melatih pengklasifikasi baru menggunakan data set dengan koefisien bobot yang diatur berdasarkan hasil pelatihan sebelumnya dengan cara memberikan bobot yang besar pada data yang salah diklasifikasikan. Selanjutnya, hasil pelatihan dikombinasikan ke bentuk *committee* menggunakan koefisien dengan bobot berbeda. Gambar 2 berikut menunjukkan skema metode *boosting*.



$$Y_M(x) = \text{sign}\ [\textstyle\sum_m^M \propto_m y_m(x)]$$

Gambar 2. Ilustrasi skema metode Boosting

Setiap pengklasifikasi dasar $y_m(x)$ dilatih menggunakan bobot $w_n^{(m)}$ tergantung pada performa pengklasifikasian dasar sebelumnya $y_{m-1}(x)$. Hasil pelatihan semua pengklasifikasi dasar dikombinasikan ke pengklasifikasi penentu $Y_m(x)$. Berikut ini adalah sebagai berikut:

1. Inisialisasi bobot data $\{w_n\}$ dengan $w_n^{(m)} = 1 / N$ untuk n = 1, 2, …, N.
2. Untuk m = 1 sampai dengan M, lakukan langkah berikut:
   a. Latih $y_m(x)$ dengan meminimalkan fungsi kesalahan sebagai berikut:

      $J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)$ . . . (2.6.1)

      Dimana $I(y_m(x_n) \neq t_n)$ adalah fungsi indikator sama dengan satu jika $y_m(x_n) \neq t_n$ dan bernilai nol untuk lainnya.

   b. Evaluasi kesalahan dengan fungsi:

$$\varepsilon_m = \frac{\sum_{n=1}^{N} w_n{}^{(m)} I(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n{}^{(m)}} \quad . . . (2.6.2)$$

Selanjutnya ε_m digunakan untuk mengevaluasi persamaan berikut:

$$\propto_m = ln \left\{ \frac{1 - \varepsilon_m}{\varepsilon_m} \right\} . . . (2.6.3)$$

c. Perbaharui bobot data dengan menggunakan persamaan berikut:

$$w_n{}^{m+1} = w_n{}^{m} \exp \{ \propto_m I(y_m(x_n) \neq t_n) \} . . . (2.6.4)$$

3. Membuat prediksi menggunakan model terakhir sebagai berikut:

$$Y_M(x) = sign \left(\sum_{m=1}^{M} \propto_m y_m(x) \right) . . . (2.6.5)$$

Pengklasifikasi dasar $y_1(x)$ dilatih menggunakan koefisien bobot $w_n^{(1)}$ yang nilainya semua sama. Pada persamaan (2.6.4), koefisien bobot $w_n{}^{m}$ dinaikan untuk data yang salah klasifikasi dan diturunkan untuk data yang telah diklasifikasikan dengan benar.