Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Perpustakaan Univeristas Multimedia Nusantara

BAB II

LANDASAN TEORI

2.1 Kualitas Pengulangan *(Repeatability)*

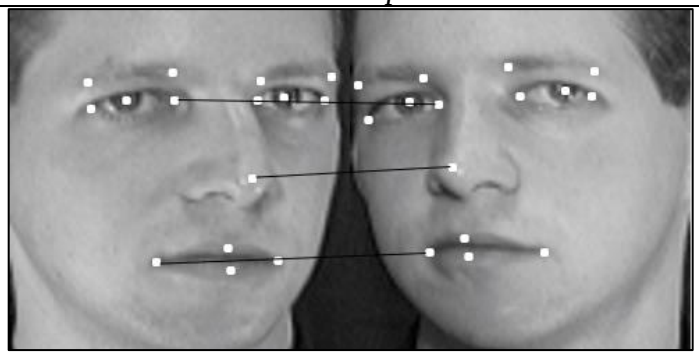
Kualitas pengulangan (*repeatability*) adalah sebuah nilai yang direpresentasikan berdasarkan *keypoints* yang saling berkorespondensi pada sepasang citra yang memiliki kondisi geometris yang bervariasi. Nilai tersebut digunakan untuk mengevaluasi performa dari *detector* dalam mendeteksi relasi yang terdapat antara dua bagian yang berbeda. Semakin tinggi tingkat korespondensi *keypoints* yang terjadi pada sepasang citra, maka semakin tinggi pula tingkat kualitas pengulangan atau *repeatability* yang terjadi. (Mikolajczyk & Schmid, 2005).

2.2 *Keypoints*

Keypoints merupakan sebuah titik yang merepresentasikan fitur dari sebuah objek yang sedang diamati sehingga objek tersebut dapat dikenali (Rudinac, Lenseigne, & Jonker, 2019). Dalam *face recognition*, *keypoints* digunakan untuk menandakan bagian dari fitur wajah manusia dan *keypoints* ini dikenal dengan *facial keypoints* (Wang, dkk, 2018). Berdasarkan Wang dan Song (2014), terdapat 15 *facial keypoints* pada manusia yang dapat digunakan untuk mengenali identitas seseorang seperti yang di representasikan pada Tabel 2.1 dan Gambar 2.1.

Tabel 2.1 Tabel 15 *Facial Keypoints*

Left eye center	*Right eye center*
Left eye inner corner	*Right eye inner corner*
Left eye outer corner	*Right eye outer corner*
Left eyebrow inner end	*Right eyebrow inner end*
Left eyebrow outer end	*Right eyebrow outer end*
Mouth left corner	*Mouth right corner*
Mouth center bottom lip	*Mouth center top lip*
Nose tip	

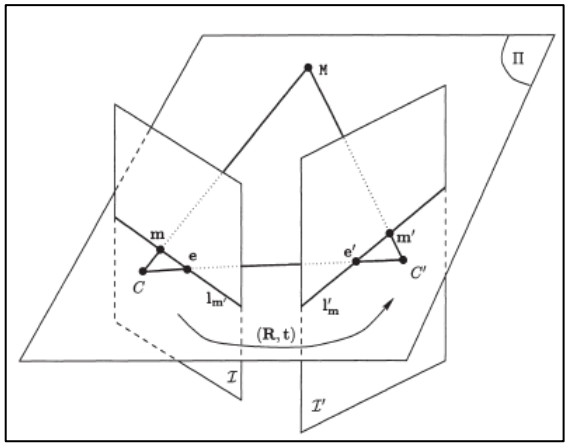


Gambar 2.1 Posisi *Facial Keypoints*

2.3 Geometri Epipolar

Geometri epipolar merupakan sebuah konsep yang digunakan untuk mengatasi permasalahan dalam memvisualisasi sebuah objek atau *scene* dari berbagai sudut pandang yang berbeda (*multiple view*) (Orozco, dkk. 2017). Dalam penerapannya, konsep geometri epipolar digunakan pada proses pencocokan secara stereo (*stereo matching*) antara dua citra yang diambil dengan sudut pandang yang berbeda untuk mengetahui bentuk 3 dimensi (3D) dari objek dalam citra tersebut (Orozco, dkk. 2017). Hal ini dilakukan dengan cara mencocokkan titik dari kedua bidang citra tersebut, yang terdapat pada sebuah garis epipolar (*epipolar lines*), kemudian dilakukan tahap triangulasi untuk mengetahui dimana letak dari bentuk 3D objek tersebut (Xu and Zhang, 1996). Untuk mendapatkan proyeksi 2D dari objek 3D yang diamati, digunakan dua buah kamera yang diletakkan secara sejajar dari dua sisi yang berbeda untuk menangkap objek yang akan diambil citranya. Untuk menyejajarkan kedua kamera tersebut, maka diperlukan sebuah garis yang

menghubungkan antara kedua kamera tersebut yang dikenal dengan *epipole* (Hartley, 2004).



Gambar 2.2 Geometri epipolar (Xu and Zhang, 1996)

Citra yang ditangkap melalui masing-masing kamera akan diproyeksikan sebuah titik citra pada sebuah bidang citra yang terbentuk dari tiap kamera. Titik citra dari gambar pertama akan memiliki pasangan titik citra pada garis epipolar yang terdapat pada di gambar kedua. Garis epipolar merupakan garis perpotongan antara tiap bidang citra dengan bidang π seperti pada Gambar 2.2. Pada garis epipolar tersebut terdapat dua buah titik penting untuk melakukan rekonstruksi wajah, yaitu titik citra dan titik yang berfungsi sebagai penghubung antara kedua posisi kamera sehingga dapat membentuk sebuah *epipole*. Ketika kedua posisi kamera sudah berada pada *epipole* yang sejajar maka dapat dilakukan triangulasi untuk melakukan rekonstruksi dari objek 3D yang diamati (Xu and Zhang, 1996).

2.4 Discrete Cosine Transform

Discrete Cosine Transform adalah sebuah metode untuk mentransformasikan citra dari domain *spatial* ke domain frekuensi dengan membentuk sekumpulan blok berukuran NxN dari *pixel* matriks suatu citra (Agarwal and Khan, 2014). Menurut Ustubioglu, dkk (2015), domain *spatial* merupakan penjumlahan dari serangkaian

gelombang kosinus (*cosine*) yang bergerak di frekuensi yang berbeda-beda. Dalam mentransformasikan citra dari domain *spatial* ke domain frekuensi digunakan rumus sebagai berikut (Agarwal and Khan, 2014):

$$C(u,v) = \alpha(u) * \alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) * cos\left[\frac{\pi(2x+1)u}{2N}\right] * \cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad (2.1)$$

Citra yang telah ditransformasikan dengan DCT akan dibagi menjadi 3 komponen frekuensi, yaitu *low-frequency*, *mid-frequency*, dan *high-frequency*. Pada citra wajah manusia *low-frequency* terdapat informasi berupa tingkat variasi pencahayaan dan permukaan halus pada citra seperti dahi, pipi dan sejenisnya. Pada *mid-frequency* memiliki informasi berupa struktur dasar dari wajah pada citra. Sedangkan *high-frequency* memiliki informasi berupa *noise* dan tepian *(edge*) (Maheshkar, dkk. 2012).

Kemudian citra yang sudah diproses di domain frekuensi akan dikembalikan ke domain *spatial* melalui *inverse* DCT dengan menggunakan rumus berikut (Agarwal and Khan, 2014):

$$f(x,y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a(u) * a(v) * C(u,v) * cos\left[\frac{\pi(2x+1)u}{2N}\right] * \cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad (2.2)$$

Dengan syarat:

$$\alpha\ (\mathrm{u}) = \begin{cases} \sqrt{\frac{1}{N}}, \text{untuk } u = 0 \\ \sqrt{\frac{2}{N}},\ untuk\ \mathrm{u} \neq 0 \end{cases}$$

$$\alpha\ (\mathrm{v}) = \begin{cases} \sqrt{\frac{1}{N}}, untuk\ v = 0 \\ \sqrt{\frac{2}{N}},\ untuk\ \mathrm{v} \neq 0 \end{cases}$$

2.5 Wiener Filtering

Wiener *filtering* merupakan salah satu metode restorasi citra yang cukup umum digunakan untuk melakukan *denoising* dan *deblurring* secara bersamaan. Wiener Filtering bekerja dalam domain frekuensi untuk membentuk sebuah filter restorasi berdasarkan nilai dari *degradation filter* yang digunakan seperti pada rumus berikut (Mitchell dan Gilmore, 1992) (Afiyat, 2017):

$$W_{(u,v)} = \frac{H^*_{(u,v)}}{|H_{(u,v)}|^2 + K} \tag{2.3}$$

$H_{(u,v)}$ merupakan *degradation filter* yang digunakan dan K merupakan nilai konstanta yang ditentukan oleh pengguna dalam melakukan penelitiannya (Afiyat, 2017) (Zisserman, 2014). Kemudian $H^*_{(u,v)}$ merupakan hasil kompleks konjugasi dari $H_{(u,v)}$. Dikarenakan tidak ada metode yang pasti untuk menentukan nilai K tersebut, maka penentuan nilai K tersebut dilakukan secara eksperimental (Mitchell dan Gilmore, 1992). Setelah perhitungan untuk filter Wiener selesai maka didapatkan filter berupa $\mathrm{W}_{(u,v)}$ yang akan diaplikasikan pada $\mathrm{G}_{(u,v)}$, yang merepresentasikan citra telah didegradasi, dengan rumus berikut (Zisserman, 2014) (Afiyat, 2017):

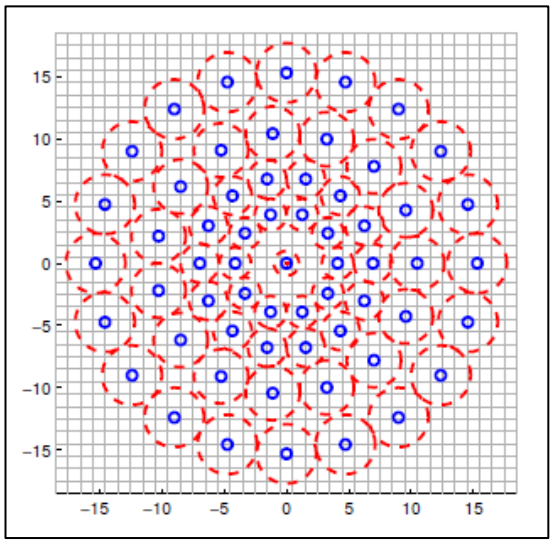
$$\hat{\mathrm{F}}_{(u,v)} = W_{(u,v)} * G_{(u,v)} \tag{2.4}$$

2.6 Feature Detectors

2.6.1 Binary Robust Invariant Scalable Keypoints (BRISK)

BRISK merupakan sebuah metode untuk mendeteksi, mendeskripsikan, dan mencocokkan *keypoints* dengan cara membandingkan tingkat kecerahan (*brightness*) yang terdapat dalam bagian citra (Leutenegger dkk., 2011). Cara kerja dari metode BRISK didasari oleh metode AGAST yang dilakukan oleh Mair,dkk

(2010). *Descriptor* yang didapatkan melalui metode ini terbentuk dari *binary string* yang merupakan hasil dari penggabungan hasil perbandingan tingkat kecerahan dari *pixel* yang ada. Agar hasil deteksi *keypoint* dapat menjadi lebih baik, maka hasil perbandingan tingkat kecerahan dilakukan dengan teliti (Leutenegger dkk., 2011).



Gambar 2.3 BRISK Sampling Pattern (Leutenegger dkk., 2011)

Kemudian, dilakukan *sampling* terhadap *keypoint* yang berdekatan dengan menggunakan sebuah *pattern* seperti pada Gambar 2.3. Dari *pattern* tersebut dilakukan *sampling* terhadap titik-titik yang berada di sekitar sebuah *keypoint* yang berada di pusat *pattern* tersebut. Dalam prosesnya, algoritma BRISK akan menerapkan *Gaussian smoothing* dengan standar deviasi σ_i yang sebanding dengan jarak antar *points* dari masing-masing daerah lingkaran yang ada (Leutenegger dkk., 2011). Tujuan dari diterapkannya *Gaussian smoothing* adalah untuk menghindari efek *aliasing*. Ketika menggunakan salah satu *keypoint* dari $N.(N-1)/2$ maka akan dilakukan pencocokan *sampling point* yang dipasangkan dengan sebuah titik di (p_i, p_j) (Leutenegger dkk., 2011). Nilai intensitas *smoothing* dari titik-titik tersebut adalah $I(p_i, \sigma_i)$ dan $I(p_j, \sigma_j)$, yang kemudian akan digunakan

untuk mengestimasi *local gradient* $g(p_i, p_j)$ melalui rumus berikut (Leutenegger dkk., 2011):

$$(p_i, p_j) = (p_j - p_i) \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2} \quad (2.5)$$

Dengan mempertimbangkan set dari $\mathcal{A}$ dari seluruh pasangan *sampling point* (Leutenegger dkk., 2011):

$$\mathcal{A} = \{(p_i, p_j \in \mathbb{R}^2 \times \mathbb{R}^2 \mid i < N \wedge j < i \wedge i, j \in \mathbb{N})\} \quad (2.6)$$

Dari pasangan yang didapatkan, terdapat dua jenis *subset* pasangan yang dibedakan berdasarkan jarak. Subset yang didapatkan adalah *subset* pasangan jarak jauh dan *subset* pasangan jarak dekat. Adapun nilai ambang batas (*threshold*) untuk menentukan jenis jarak pasangan, yaitu $\delta_{max} = 9.75t$ dan $\delta_{min} = 13.67t$ (t adalah skala dari k) (Leutenegger dkk., 2011). Jarak antar pasangan dianggap jauh apabila nilai jaraknya lebih besar dari δ_{min}, sedangkan untuk jarak dekat harus lebih kecil dari ambang batas δ_{max}. Pasangan jarak jauh akan digunakan untuk membuat *descriptor* menjadi *invariant* terhadap rotasi dengan menerapkan rumus (2.7) di sekitar *descriptor* tersebut (Leutenegger dkk., 2011).

$$\alpha = arctan2(g_y, g_x) \quad (2.7)$$

Kemudian, pasangan jarak pendek akan digunakan untuk menentukan *descriptor* dengan syarat seperti berikut (Leutenegger dkk., 2011):

$$b = \begin{cases} 1, & I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i) \\ 0, & otherwise \end{cases} \quad (2.8)$$

2.6.2 Features from Accelerated Segment Test (FAST)

FAST merupakan sebuah algoritma untuk mendeteksi *interest points* yang diciptakan oleh Rosten dan Drummond dari sebuah citra (Viswanathan, 2011).

Interest points adalah sebuah *pixel* yang posisinya dapat digambarkan dan dideteksi dengan jelas.

Dalam mendeteksi *interest points*, algoritma FAST memiliki tahapan kerja sebagai berikut (Kadhim and Araheemah, 2019) (Viswanathan, 2011):

1. Memilih sebuah *pixel* "p" dari sebuah citra dengan mengasumsikan intensitas dari *pixel* tersebut I_p. *Pixel* ini kemudian akan dideteksi sebagai sebuah *interest point* atau bukan.
2. Menetapkan nilai *threshold* atau ambang batas "T".
3. Membuat sebuah lingkaran yang terdiri dari 16 *pixels* yang mengelilingi *pixel* p.
4. Jika *pixel* tersebut perlu dinyatakan sebagai sebuah *interest point*, maka setidaknya sebanyak N *pixel* dari 16 *pixels* berada di atas ataupun di bawah nilai I_p dengan nilai T.
5. Kemudian akan dilakukan proses perbandingan tingkat intensitas dari *pixel* ke- 1, 5, 9, dan 13 yang berada dalam lingkaran dengan nilai intensitas *pixel*, I_p. Jika setidaknya terdapat 3 dari 4 *pixels* tersebut yang nilainya memenuhi syarat *threshold*, maka *interest point* akan terdeteksi.
6. Jika terdapat, setidaknya, tiga dari empat *pixel* yang digunakan bernilai tidak di atas maupun di bawah $I_p + T$, maka *pixel* P bukan merupakan *interest points*.
7. Ulangi semua langkah sebelumnya untuk *pixel* lainnya yang terdapat pada citra yang diteliti.

2.6.3 Harris-Stephens

Harris-Stephens *feature detector* adalah sebuah algoritma yang dikembangkan oleh Harris dan Stephens (1988) dalam mendeteksi *feature points* yang terdapat dalam citra berdasarkan unsur pendeteksian tepi (*edge*) dan sudut (*corner*) dengan menggunakan fungsi *auto-correlation* lokal. *Auto-correlation detector* merupakan sebuah fungsi yang dikembangkan oleh Harris dan Stephens (1998) untuk menanggulangi permasalahan pada penelitian sebelumnya yang menggunakan menerapkan *corner detection* dengan metode Moravec (Harris & Stephens, 1988).

Perubahan intensitas dari dua titik yang berbeda direpresentasikan melalui nilai E yang didapatkan menggunakan rumus berikut (Harris & Stephens, 1988):

$$E_{x,y} = \sum_{u,v} w_{u,v} \left[I_{x+u,y+v} - I_{u,v} \right]^2 \quad (2.9)$$

Pada nilai E (x,y) dapat dinyatakan secara singkat seperti berikut (Harris & Stephens, 1988):

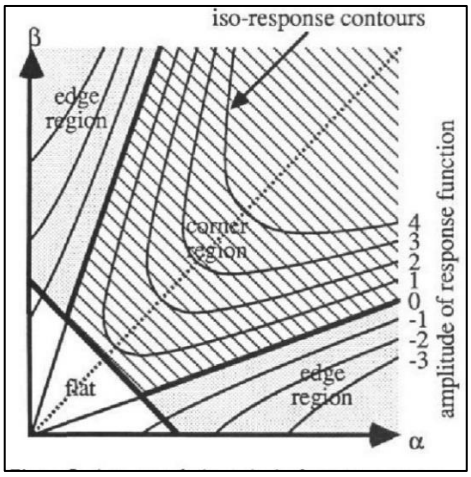
$$E(x,y) = (x,y)\, M\, (x,y)^T \quad (2.10)$$

Dengan matriks simetris 2x2 seperti berikut (Harris & Stephens, 1988):

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

Beberapa hal yang perlu diketahui adalah E merupakan nilai yang paling berpengaruh dalam fungsi *local autocorrelation*, dengan M yang merepresentasikan bentuknya (Harris and Stephens, 1988). Asumsikan α, β adalah eigenvalue dari M, α dan β akan proporsional terhadap prinsip kelengkungan (*curvatures*) dari fungsi *local auto-correlation,* kemudian membentuk M yang konstan di tiap rotasinya. Dari pernyataan tersebut, dapat disimpulkan menjadi tiga kondisi yang perlu dipertimbangkan, yaitu (Harris and Stephens, 1988):

1. Jika kedua *curvatures* yang didapat kecil sehingga membentuk fungsi *local auto-correlation* yang mendatar (*flat*), maka daerah citra yang didapatkan terdiri dari intensitas yang konstan. (contohnya: segala perubahan yang terjadi pada citra tidak memiliki pengaruh yang besar terhadap nilai E).
2. Jika satu *curvature* tinggi dan satu lainnya memiliki *curvature* yang rendah sehingga membentuk fungsi *local auto-correlation* yang membumbung (*ridge*), maka perubahan yang terjadi sepanjang bumbungan tersebut hanya berpengaruh kecil terhadap nilai E dan hal ini yang menandakan sebuah tepian (*edge*).
3. Jika kedua *curvatures* tinggi sehingga membentuk fungsi *local auto-correlation* yang berbentuk sebuah puncak tajam (*sharply peaked*), maka segala perubahan yang terjadi di tiap arah akan meningkatkan nilai E dan hal ini yang menadakan sebuah sudut (*corner*).



Gambar 2.4 Penentuan Suatu Daerah Pada Harris-Stephens (Harris and Stephens, 1988)

Setelah melakukan klasifikasi tepi dan sudut, perlu dilakukan pengukuran terhadap kualitas dari respon yang didapatkan juga. Besar dari respon yang didapatkan akan digunakan untuk memilih *pixel* dari sudut yang terpencil dan untuk

menipiskan *pixel* dari tepian. Asumsikan ukuran dari respon sudut (*corner response*) berupa R, yang diperlukan untuk menjadi fungsi dari α dan β (Harris and Stephens, 1988). Untuk menghindari nilai *eigenvalue* dari M yang didekomposisi secara eksplisit, maka digunakan $Tr(M)$ dan $Det(M)$ dalam rumus berikut (Harris and Stephens, 1988):

$$Tr(M) = \alpha + \beta = A + B \tag{2.11}$$

$$Det(M) = \alpha\beta = AB - C^2 \tag{2.12}$$

Untuk menghitung respon sudut yang didapatkan digunakan rumus berikut (Harris and Stephens, 1988):

$$R = Det - k\,Tr^2 \tag{2.13}$$

Dengan k yang merupakan konstanta dengan *range* nilai dari 0,04 hingga 0,06.

Nilai R akan digunakan untuk mengidentifikasi sebuah *corner*, yaitu jika nilai R yang positif maka titik tersebut merupakan sebuah *corner*. Jika R bernilai negatif, maka itu menunjukan daerah *edge* (Kadhim and Araheemah, 2019).

2.6.4 Minimum Eigenvalue

Minimum Eigenvalue merupakan sebuah *feature detector* yang dikembangkan oleh Shi dan Tomasi (1994) berdasarkan *feature detector* Harris-Stephens. Maka dari itu, cara kerja dari *feature detector* ini dalam mendeteksi *feature points* sama dengan cara kerja Harris-Stephens, yang membedakannya hanyalah cara untuk menentukan sebuah *feature points* (Kadhim and Araheemah, 2019). Dalam menentukan sebuah *corner,* Minimum Eigenvalue menggunakan rumus seperti berikut (Kadhim and Araheemah, 2019):

$$R = Minimum\ (\lambda_1, \lambda_2) \tag{2.14}$$

2.6.5 Speeded-Up Robust Features (SURF)

SURF merupakan sebuah algoritma *feature detector* yang terdiri dari 2 tahap, yaitu tahap ekstraksi dan tahap deskripsi. Dalam melakukan tahap ekstraksi, SURF melakukan perhitungan yang berbasis pada matriks *Hessian* yang dikenal dengan metode *Fast Hessian Matrix* untuk mendeteksi *interest points* (Bay dkk., 2006). Sedangkan untuk tahap deskripsi digunakan nilai total dari respon yang dihasilkan dari *Haar-wavelet* yang terdapat di sekitar *interest point* pada sebuah arah *x* dan *y* (Bay dkk., 2006).

SURF menggunakan *integral image* untuk mempercepat proses konvolusi dari Gaussian Filtering. Sebuah citra integral yang terletak $I_{\Sigma}(x)$ pada lokasi $x = (x, y)$ merepresentasikan jumlah semua *pixel* dari citra yang diinputkan I yang memiliki daerah berbentuk segi empat yang terbentuk dari titik *x* dan didefinisikan sebagai berikut (Bay dkk., 2006):

$$I_{\Sigma}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,j) \tag{2.15}$$

SURF kemudian menggunakan metode Fast Hessian Detector untuk mendeteksi *interest points*. Jika didapatkan sebuah titik $x = (x, y)$ dalam sebuah citra *I*, maka matriks *Hessian* $\mathcal{H}(x, \sigma)$ pada titik *x* dengan skala σ dapat didefinisikan sebagai rumus berikut (Bay dkk., 2006):

$$\mathcal{H}(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \tag{2.16}$$

Dengan $L_{xx}(x, \sigma)$ merupakan hasil *convolution* dari nilai turunan kedua Gaussian. Kemudian untuk mendapatkan nilai determinan dari matriks *Hessian*, digunakan filter Gaussian dan dilakukan *sub-sampling* dari hasil tersebut agar dapat mencapai tingkat piramida yang lebih tinggi (Bay dkk., 2006). Piramida yang dimaksud adalah piramida citra (*image pyramids*) yang merupakan hasil

implementasi dari skala ruangan (*scale space*). Untuk tahap *descriptor*, pertama SURF mencari arah paling dominan dengan membuat daerah lingkaran di sekitar *interest points* dan mengaplikasikan *haar wavelet* pada arah x dan y. Kedua, dibuat sebuah daerah persegi sesuai dengan arah dari langkah pertama untuk mendeskripsikan *keypoints* (Bay dkk., 2006).

2.7 F-Score

F-Score merupakan sebuah nilai rata-rata harmonis antara nilai *precision* dan *recall* dengan menggunakan rumus (Sasaki, 2007):

$$F - Score = \frac{2 \times precision \times recall}{precision + recall} \tag{2.17}$$

Recall digunakan untuk merepresentasikan tingkat relevansi suatu kejadian berdasarkan aturan yang diberlakukan atau untuk merepresentasikan tingkat kejadian *true positives* (Powers, 2011). *Recall* memiliki rumus seperti berikut (Powers, 2011):

$$recall = \frac{true\ positives}{(true\ positives + false\ negatives)} \tag{2.18}$$

Dalam penelitian ini, nilai *recall* merupakan sebuah nilai yang merepresentasikan perbandingan jumlah *facial keypoints* yang berkorespondensi dengan total pasangan *facial keypoints* yang seharusnya terjadi (Kusnadi, dkk, 2018) (Kusnadi, dkk. 2019). Maka dari itu, rumus untuk menentukan nilai *recall* dalam penelitian kali ini adalah seperti pada rumus berikut (Kusnadi, dkk. 2018) (Kusnadi, dkk. 2019):

$$recall = \frac{No.correct\ matches}{Total\ No.Correspondences} \tag{2.19}$$

Precision merepresentasikan tingkat akurasi dari kejadian yang merupakan *True Positive* karena nilai tersebut mengukur seberapa akurat tingkat kejadian *Predicted Posisitives* berdasarkan total kejadian *Real Positives* yang ditemukan. Maka dari itu, *precision* memiliki rumus seperti berikut (Powers, 2011):

$$precision = \frac{true\ positives}{(true\ positives + false\ positives)} \tag{2.20}$$

Dalam penelitian kali ini, nilai *precision* didapatkan dari jumlah pasangan yang benar terhadap jumlah titik yang berpasangan (Kusnadi, dkk, 2018) (Kusnadi, dkk, 2019). Dengan kata lain, nilai *precision* yang digunakan dalam penelitian kali ini didapatkan melalui rumus berikut (Kusnadi, dkk. 2018) (Kusnadi, dkk. 2019):

$$precision = \frac{correct\ matches}{Total\ No.All\ Matches} \tag{2.21}$$

2.8 Deskripsi *Dataset*

Dataset yang digunakan dalam penelitian kali ini diperoleh dari ORL (Olivetti Research Lab) dan Head Pose *Image Database*. Dari masing-masing *dataset* diambil subjek sebanyak 5 orang dan digunakan dua citra yang diambil dari sisi kiri dan kanan untuk tiap subjek.

2.8.1 ORL Image Database

ORL (Olivetti Research Lab), yang sekarang dikenal dengan nama AT&T, *The Database of Faces*, merupakan sebuah *database* citra wajah dari 40 subjek yang berbeda dan diambil dari bulan April 1992 hingga April 1994 di sebuah laboratorium untuk penelitian. Setiap subjek memiliki 10 citra yang berbeda dan tiap citra memiliki latar yang gelap serta memiliki tingkat kecerahan (*brightness*) dan ekspresi wajah yang berbeda-beda ("The Database of Faces," 2001). Berikut

merupakan kelima subjek dari *dataset* ORL yang digunakan dalam penelitian kali ini.

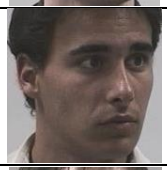
Tabel 2.2 Subjek yang Digunakan Dari *Dataset* ORL

No.	Subjek	Kiri	Kanan
1.	*Subject 1*		
2.	*Subject 10*		
3.	*Subject 23*		
4	*Subject 39*		
5.	*Subject 40*		

2.8.2 Head Pose Image Database

Head Pose merupakan *database* berupa citra wajah dari 15 orang yang berbeda dengan variasi sudut pengambilan citra yang berkisar dari sudut -90° hingga sudut +90° dan pose yang berbeda-beda. Beberapa diantara orang yang menjadi subjek dalam *database* ini mengenakan kacamata dan memiliki warna kulit yang berbeda (Gourier dkk., 2004). Berikut merupakan kelima subjek yang digunakan dari *dataset* Head Pose dalam penelitian kali ini.

Tabel 2.3 Subjek yang Digunakan Dari *Dataset* Head Pose

No.	Subjek	Kiri	Kanan
1.	*Person 3*		
2.	*Person 6*		
3.	*Person 10*		
4	*Person 11*		
5.	*Person 13*		