BAB II

LANDASAN TEORI

2.1 Histogram Equalization

Histogram Equalization adalah teknik yang digunakan untuk mengatur intensitas gambar dengan cara meningkatkan kontras ("Histogram Equalization", n.d.). Proses *equalization* adalah proses nonlinear dan tidak dapat dikembalikan (*irreversible*) (Sundararajan, 2017).

Representasi dari probabilitas sebuah *gray level* senilai u adalah (Sundararajan, 2017):

$$p(u) = \frac{n_u}{N \times N} \quad u = 0, 1, 2, \dots, L-1 \qquad ..(2.1)$$

Dimana:

$N \times N$: jumlah dari piksel di gambar

n_u : jumlah dari piksel yang memiliki *gray level* senilai u

L : jumlah dari *gray level* yang ada di gambar

Histogram Equalization dihitung dengan Persamaan (2) yang dimana v adalah nilai *gray level* yang berkorespondensi di gambar yang telah diaplikasikan *Histogram Equalization* (Sundararajan, 2017).

$$v = (L-1) \sum_{n=0}^{u} p(n), u = 0, 1, \dots, L-1 \qquad ..(2.2)$$

Dimana:

$p(n)$: probabilitas dari sebuah *gray level.*

2.2 *Discrete Cosine Transform* (DCT)

Discrete Cosine Transform (DCT) dikenalkan oleh Ahmed, Natarajan, & Kao (1974). DCT sendiri telah menjadi alat penting untuk banyak aplikasi seperti pemrosesan sinyal digital, pengolahan citra, dan penyembunyian informasi (Zhou & Chen, 2009). DCT dapat digunakan untuk pemisahan sinyal gambar menjadi komponen frekuensi (Uehara, Safavi-Naini, & Agunbona, 2006). *Inverse* DCT sendiri digunakan untuk mengembalikan gambar dari sinyal frekuensi (*frequency domain*) menjadi gambar (*spatial domain*) kembali (Agarwal & Khan, 2014).

DCT dapat didefinisikan sebagai (Agarwal & Khan, 2014):

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)cos\left[\frac{\pi(2x+1)u}{2N}\right]cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad ..(2.3)$$

Bentuk *inverse* dari Persamaan (3) adalah (Agarwal & Khan, 2014):

$$f(x,y) = \sum_{u-0}^{N-1}\sum_{v-0}^{N-1} \alpha(u)\alpha(v)c(u,v)cos\left[\frac{\pi(2x+1)u}{2N}\right]cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad ..(2.4)$$

Dimana:

$C(u,v)$: Nilai frekuensi untuk u,v

$f(x,y)$: Nilai warna dari *piksel* di posisi (x,y)

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & , u = 0 \\ \sqrt{\frac{2}{N}} & , u \neq 0 \end{cases} \quad ..(2.5)$$

$$\alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} & , v = 0 \\ \sqrt{\frac{2}{N}} & , v \neq 0 \end{cases} \quad ..(2.6)$$

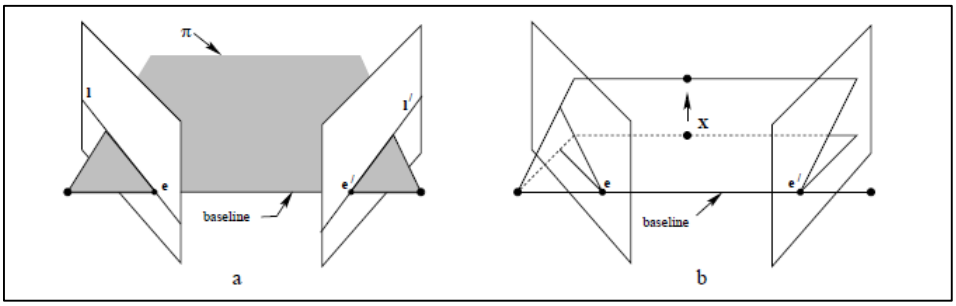
DCT sendiri dapat digunakan untuk mendapatkan komponen berfrekuensi rendah berisi informasi untuk pengenalan wajah dan sensitif terhadap pencahayaan sementara komponen berfrekuensi tinggi berisi informasi mengenai sudut dan sensitif terhadap pose dan variasi ekspresi (Thamizharasi & J.S, 2016). Pada bagian pencahayaan sendiri, koefisien pertama dari DCT (komponen DC) menentukan kualitas pencahayaan secara keseluruhan dari gambar (Chen, Er, & Wu, 2006) dan merupakan nilai rata-rata dari blok gambar serta menyimpan energi yang paling besar di gambar sementara komponen AC membawa energi sesuai dengan banyaknya jumlah detail di blok gambar (Uehara, Safavi-Naini, & Agunbona, 2006). Untuk mengatur nilai dari komponen DC sendiri dapat menggunakan Persamaan (2.7) berikut (Chen, Er, & Wu, 2006):

$$C(0,0) = \log \mu \cdot \sqrt{MN} \qquad ..(2.7)$$

Nilai dari μ adalah nilai rata-rata dari *gray level* gambar yang digunakan (Chen, Er, & Wu, 2006).

2.3 Geometri Epipolar

Dalam rekonstruksi gambar 3D, dapat digunakan gambar 2D untuk mendapatkan informasinya dan terdapat dua pendekatan untuk dapat melakukan hal tersebut, yaitu: *laser scanning based* dan *computer vision based*. Salah satu metode dengan pendekatan *computer vision* adalah geometri epipolar. Geometri epipolar bekerja dengan menerima masukan berupa sepasang gambar tanpa diperlukan informasi tambahan sebelumnya (Takimoto dkk., 2012). Menurut Brandt (2008), geometri epipolar membantu untuk menemukan titik-titik yang cocok diantara dua gambar karena korespondensi dari titik di gambar pertama harus berada di garis epipolar yang berkorespondensi di gambar kedua.



Gambar 2.1 Geometri Epipolar
(Hartley & Zisserman, 2004)

Gambar 2.1 (Hartley & Zisserman, 2004) menjelaskan mengenai geometri epipolar yang dimana bagian a menjelaskan bahwa *baseline* dari kamera memotong dari dua planar tiap gambar di *epipole* (titik pertemuan dari garis yang menggabungkan titik tengah dari kamera (*baseline*) dengan planar gambar) *e* dan e'. Setiap planar π yang mengandung *baseline* adalah planar epipolar (planar yang mengandung *baseline*) dan memotong planar gambar dari garis epipolar (pertemuan antara planar epipolar dengan planar gambar) l dan l' yang berkorespondensi. Gambar 2.1 (Hartley & Zisserman, 2004) bagian b menjelaskan bahwa titik X di tiga dimensi dapat bervariasi dan planar epipolar berputar mengitari *baseline*. Semua garis epipolar berpotongan di *epipole*.

2.4 Pengulangan

Pengulangan (*repeatability*) merujuk pada rata-rata jumlah daerah yang berkorespondensi dan terdeteksi pada gambar dibawah transformasi geometris yang berbeda baik secara absolut maupun secara relatif (Mikolajczyk dkk., 2006). Menurut Mikolajczyk & Schmid (2005), semakin tinggi nilai pengulangannya, semakin banyak titik yang berpotensi dapat dihubungkan serta lebih baik hasil serta pengenalannya.

2.5 *Keypoint*

Keypoint adalah salah satu konsep kritis dalam deteksi pada rekonstruksi tiga dimensi (Kusnadi dkk., 2018). Tabel 2.1 menampilkan daftar dari 15 *keypoint* di wajah (Wang & Song, 2014).

Tabel 2.1 15 *Keypoint* di Wajah (Wang & Song, 2014)

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 top lip	*Mouth center bottom lip*
Nose tip	

2.6 *Feature Detectors*

Feature extraction adalah proses mentransformasikan sebuah data masukan menjadi sebuah set fitur yang dimana berisi informasi yang relevan dari data masukan tersebut. *Feature extraction* ini sendiri adalah bentuk dari sebuah pengurangan dimensi (*dimensionality reduction*) dan bertujuan untuk mendapatkan informasi yang paling relevan dari data masukan dan merepresentasikan informasi tersebut dalam dimensi yang lebih rendah. Fitur haruslah bersifat seunik mungkin

pada tiap objek sehingga dapat membedakan tiap objeknya namun memiliki pola yang mirip pada kelas yang serupa. Fitur terdiri dari dua kategori, yaitu: lokal dan global (Kumar & Bhatia, 2014).

Feature extraction yang bersifat lokal terdiri dari *feature detector* dan *feature descriptor* (Madbouly, Wafy, & Mostafa, 2015). *Feature detection* adalah proses ekstraksi informasi dari gambar dengan cara mencari dan menemukan apakah terdapat sebuah fitur yang mirip dengan tipe fitur yang sudah tersedia sebelumnya, seperti titik, garis, sudut, atau *blob*. Metode deteksi fitur dan algoritmanya memiliki kontribusi yang banyak dalam implementasi pada aplikasi di *computer vision* seperti penyamaan fitur (*feature matching*), rekonstruksi tiga dimensi, pencarian gambar, maupun kategorisasi objek. Sementara itu, *feature description* adalah proses untuk mendeskripsikan sebuah titik yang telah dipilih (*interest point*) dan membuat sebuah vektor yang berisi semua informasi mengenai ukuran dan arahnya (Ali & Salman, 2018).

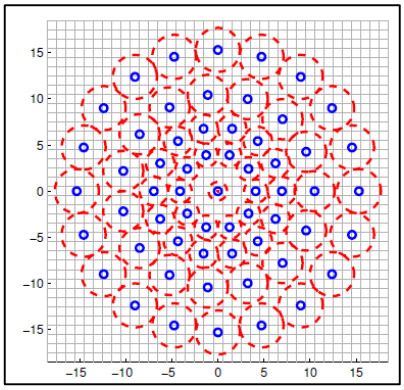
Berikut adalah lima *feature detectors* yang akan digunakan untuk mencari *keypoint* pada gambar di penelitian ini:

2.6.1. *Binary Robust Invariant Scalable Keypoints* (BRISK)

Binary Robust Invariant Scalable Keypoints (BRISK) diperkenalkan oleh Leutenegger, Chli, & Siegwart (2011) yang dimana memiliki karakteristik untuk dapat bekerja pada waktu komputasi yang lebih singkat dan bekerja dengan langkah *scale-space keypoint detection* dan *keypoint description.*

Langkah *scale-space keypoint detection* adalah mengidentifikasi titik-titik yang dikehendaki (*interest point*) diantara gambar dan dimensi skala dengan menggunakan *saliency criterion.* Untuk meningkatkan efisiensi dari komputasi,

keypoint dideteksi diantara setiap lapisan dari *image pyramid*. Langkah *keypoint description* bekerja dengan mengaplikasikan sebuah pola *sampling* pada titik-titik yang bertetanggaan dengan *keypoint* untuk mendapatkan *gray values*. Pola *sampling* dari BRISK dapat dilihat pada Gambar 2.2. Langkah ini akan memproses gradien dari intensitas lokal yang kemudian dapat digunakan untuk menentukan karakteristik arah dari *keypoint*. Langkah ini akan menghasilkan *descriptor* untuk BRISK (Leutenegger, Chli, & Siegwart, 2011).



Gambar 2.2 Pola *Sampling* dari BRISK
(Leutenegger, Chli, & Siegwart, 2011)

Untuk menghindari efek *aliasing* saat melakukan *sampling* intensitas gambar dari sebuah poin p_i di pola, Gaussian *smoothing* diaplikasikan dengan standar deviasi σ_i yang proporsional dengan jarak antara titik dan lingkaran yang berada dari pola. Peletakan dan pengaturan skala pola dilakukan untuk *keypoint k* di gambar dan dianggap bahwa salah satu dari $N \cdot (N-1)/2$ yang merupakan titik *sampling* berpasangan dengan (p_i, p_j). Nilai intensitas yang telah diaplikasikan *smoothing* dari setiap titik yang adalah $I(p_i, \sigma_i)$ dan $I(p_j, \sigma_j)$ yang kemudian digunakan untuk melakukan estimasi dari gradien lokal $g(p_i, p_j)$ dengan Persamaan (2.8) (Leutenegger, Chli, & Siegwart, 2011):

$$g(p_i, p_j) = (p_j - p_i) \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2} \quad ..(2.8)$$

Dengan mempertimbangkan set dari $\mathcal{A}$ dari semua pasangan *sampling-point* (Leutenegger, Chli, & Siegwart, 2011):

$$\mathcal{A} = \{(p_i, p_j \in \mathbb{R}^2 \times \mathbb{R}^2 \mid i < N \wedge j < i \wedge i, j \in \mathbb{N})\} \quad ..(2.9)$$

Dapat didefinisikan *subset* dari pasangan yang memiliki jarak pemisah pendek $\mathcal{S}$ dan *subset* pasangan yang memiliki jarak pemisah panjang $\mathcal{L}$ (Leutenegger, Chli, & Siegwart, 2011):

$$\mathcal{S} = \{(p_i, p_j) \in \mathcal{A} \mid \|p_j - p_i\| < \delta_{max}\} \subseteq \mathcal{A} \quad ..(2.10)$$

$$\mathcal{L} = \{(p_i, p_j) \in \mathcal{A} \mid \|p_j - p_i\| > \delta_{min}\} \subseteq \mathcal{A} \quad ..(2.11)$$

Ambang batas dari jarak dipasang pada $\delta_{max} = 9.75t$ dan $\delta_{min} = 13.67t$ (t adalah skala dari k). Dengan iterasi melalui pasangan titik di $\mathcal{L}$, dilakukan estimasi keseluruhan karakteristik pola arah dari *keypoint k* dengan (Leutenegger, Chli, & Siegwart, 2011):

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \cdot \sum_{(p_i, p_j) \in \mathcal{L}} g(p_i, p_j) \quad ..(2.12)$$

Untuk membentuk formasi dari *descriptor* yang telah dinormalisasi perputaran dan skalanya, BRISK mengaplikasikan pola *sampling* yang diputar dengan g di sekitar *keypoint* k. *Bit-vector descriptor* d_k dihimpun di sekitar semua pembanding intensitas yang memiliki jarak pendek dari pasangan titik $(p_i^\alpha, p_j^\alpha) \in \mathcal{S}$ agar setiap bit b berkorespondensi dengan (Leutenegger, Chli, & Siegwart, 2011):

$$b = \begin{cases} 1, & I(p_j^{\alpha}, \sigma_j) > I(p_i^{\alpha}, \sigma_i) \\ 0, & otherwise \end{cases} \quad ..(2.13)$$

Dalam membentuk *descriptor*, BRISK menggunakan pola *sampling* deterministik yang menghasilkan konsentrasi titik *sampling* yang serupa di sekitar *keypoint* dan penggunaan Gaussian *smoothing* sendiri tidak akan menghilangkan informasi dari perbandingan kontras yang dimana terjadi dengan melakukan *blurring* pada dua titik *sampling* yang berdekatan untuk perbandingan. Lalu, BRISK juga menggunakan titik *sampling* yang lebih sedikit untuk membatasi kompleksitas dalam membandingkan nilai intensitas dan pada akhirnya, perbandingan yang dilakukan dibatasi secara spasial yang dimana hanya dibutuhkan variasi iluminasi yang konsisten secara lokal (Leutenegger, Chli, & Siegwart, 2011).

2.6.2. *Features from Accelerated Segment Test* (FAST)

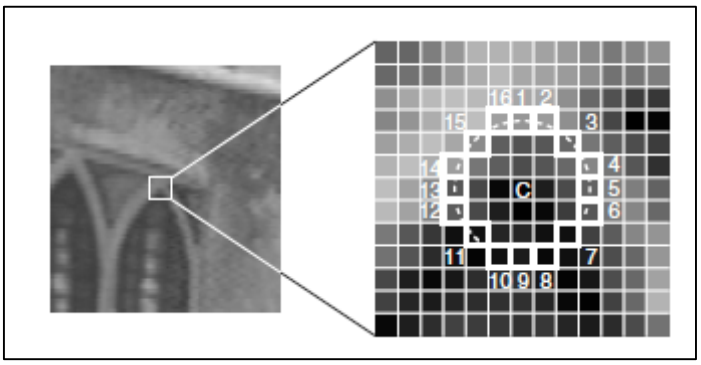
Features from Accelerated Segment Test (FAST) adalah algoritma yang dikenalkan pertama kali oleh Rosten dan Drummond yang ditujukan untuk mengidentifikasi titik fitur dari sebuah gambar. Titik fitur dari sebuah gambar adalah sebuah piksel yang memiliki posisi yang terdefinisi secara baik dan dapat secara kuat dideteksi, memiliki konten informasi lokal yang tinggi dan secara ideal, harus dapat diulang diantara gambar yang berbeda (Viswanathan, 2011).

Algoritma dari FAST adalah sebagai berikut (Viswanathan, 2011):

1. Pilih sebuah piksel *p* di dalam gambar. Asumsikan intensitas dari piksel adalah I_p. Piksel ini adalah piksel yang akan diidentifikasikan sebagai titik fitur.

2. Atur sebuah ambang batas intensitas bernilai T.

3. Pertimbangkan sebuah lingkaran yang terdiri dari 16 piksel yang mengelilingi piksel *p.*



Gambar 2.3 Deteksi Piksel dengan FAST
(Rosten & Drummond, 2005)

4. Sejumlah *N* piksel bernilai T yang berdekatan dari 16 piksel harus berada di salah satu atas atau bawah dari I_p jika piksel ingin dideteksi sebagai fitur.

5. Untuk membuat algoritma berjalan cepat, pertama bandingkan intensitas dari piksel 1, 5, 9, dan 13 dari lingkaran dengan I_p. Setidaknya harus ada tiga dari empat piksel yang memenuhi kriteria ambang batas agar titik fitur dianggap ada.

6. Jika tiga dari empat nilai piksel (I_1, I_5, I_9, I_{13}) tidak di atas atau di bawah $I_p + T$, maka P bukan titik fitur. Dalam kasus ini, tolak piksel *p* sebagai kemungkinan titik fitur. Tetapi, jika setidaknya tiga dari piksel berada di atas atau di bawah $I_p + T$, maka periksa semua 16 piksel dan periksa apakah 12 piksel yang berdekatan masuk dalam kriteria.

7. Ulang prosedur tersebut untuk semua piksel di gambar.

2.6.3. Harris-Stephens

Harris-Stephens *feature detector* dikenalkan oleh Chris Harris dan Mike Stephens yang dimana kerja utamanya berasal dari kombinasi deteksi pada tepi dan sudut yang didasarkan dari fungsi *auto-correlation* secara lokal (Harris & Stephens, 1988).

Anggap gambar adalah sebuah fungsi skalar $I : \Omega \rightarrow \mathcal{R}$ dan h adalah sebuah penambahan (*increment*) yang kecil di sekitar posisi manapun di daerah $x \in \Omega$. Titik sudut didefinisikan sebagai titik x yang membesarkan nilai dari Persamaan (2.14) untuk setiap perpindahan kecil sebesar h (Sánchez, Monzón, & Salgado, 2018),

$$E(h) = \sum w(x)(I(x+h) - I(x))^2 \qquad ..(2.14)$$

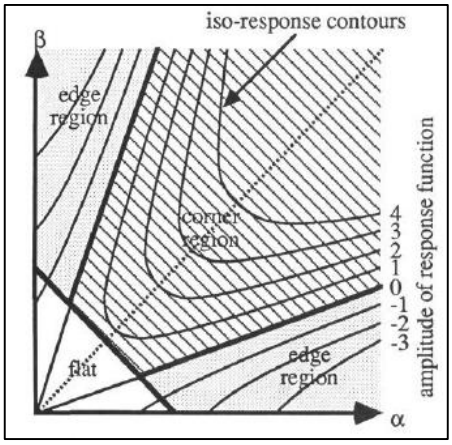
Fungsi $w(x)$ adalah *support region* yang biasanya didefinisikan sebagai persegi atau sebuah fungsi Gaussian. Ekspansi Taylor dapat digunakan untuk melinearisasi $I(x+h)$ menjadi $I(x+h) \simeq I(x) + \nabla I(x)^T h$ yang dimana bagian kanan dari Persamaan (2.14) dapat menjadi (Sánchez, Monzón, & Salgado, 2018):

$$E(h) \simeq \sum w(x)(\nabla I(x)h)^2 dx = \sum w(x)(h^T \nabla I(x) \nabla I(x)^T h) \qquad ..(2.15)$$

Persamaan (2.16) bergantung pada gradien dari gambar yang diproses dengan matriks *auto-correlation* yang dimana memiliki bentuk sebagai berikut (Sánchez, Monzón, & Salgado, 2018):

$$M = \sum w(x)(\nabla I(x) \nabla I(x)^T) = \begin{pmatrix} \sum w(x) I_x^2 & \sum w(x) I_x I_y \\ \sum w(x) I_x I_y & \sum w(x) I_y^2 \end{pmatrix} \qquad ..(2.16)$$

Nilai tertinggi dari Persamaan (2.14) ditemukan melalui analisis dari matriks ini. Eigenvalue terbesar dari *M* berkorespondensi pada arah dari variasi intensitas terbesar sementara nilai terbesar kedua berkorespondensi pada variasi intensitas di area orthogonal (Sánchez, Monzón, & Salgado, 2018).



Gambar 2.4 Klasifikasi Daerah Datar/Tepi/Sudut
(Harris & Stephens, 1988)

Dari hal ini, dapat ditemukan tiga kemungkinan (Harris & Stephens, 1988):

1. Jika nilai kedua eigenvalue kecil maka fungsi *autocorrelation* lokal adalah datar.
2. Jika salah satu nilai eigenvalue besar dan nilai lainnya adalah kecil, maka fungsi *autocorrelation* lokal berbentuk seperti punggung bukit. Hal ini menunjukkan daerah tepi.
3. Jika kedua eigenvalue adalah besar, maka fungsi *autocorrelation* lokal memiliki bentuk puncak tajam. Hal ini menunjukkan sudut.

2.6.4. Minimum Eigenvalue (Shi-Tomasi)

Minimum Eigenvalue atau dikenal juga sebagai Shi-Tomasi diperkenalkan oleh Jianbo Shi dan Carlo Tomasi (Shi & Tomasi, 1994). *Feature detector* ini didasarkan dari metode yang diperkenalkan oleh Harris-Stephens (Gauglitz,

Höllerer, & Turk, 2011) yang dimana menambahkan langkah tambahan yaitu membandingkan nilai minimum dari dua eigenvalue dan hanya membandingkan *window* yang berisi variasi intensitas yang lebih besar dari nilai ambang batas (*threshold*) sebesar λ yang telah ditentukan. Persamaan (2.17) menyatakan syarat penerimaan daerah gambar (Shi & Tomasi, 1994):

$$min(\lambda_1, \lambda_2) > \lambda \quad ..(2.17)$$

2.6.5. *Speeded-Up Robust Features* (SURF)

Speeded-Up Robust Features (SURF) adalah skema detektor-deskriptor yang dikenalkan oleh Bay, Tuytelaars, & Gool (2006) yang dimana memiliki detektor yang berdasarkan matriks Hessian. *Feature detector* ini terinspirasi oleh SIFT dan memiliki karakteristik *scale invariant* dan *rotation invariant* (Işık & Özkan, 2014).

Detektor pada SURF didasarkan pada determinan matriks Hessian yang ditujukan untuk memilih lokasi dan skala. Pada titik $x = (x, y)$ di gambar I, matriks Hessian $\mathcal{H}(x, \sigma)$ di x dengan skala σ didefinisikan sebagai (Bay, Tuytelaars, & Gool, 2006):

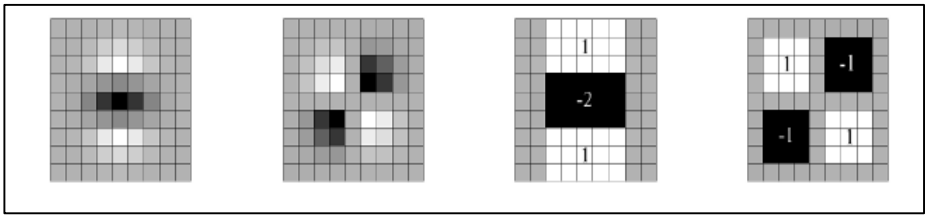
$$\mathcal{H}(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad ..(2.18)$$

Untuk menghitung determinan dari matriks Hessian, diaplikasikanlah filter Gaussian dan kemudian dilakukan penurunan kedua. Alasan mengapa SURF dapat berjalan dengan cepat adalah karena penggunaan perhitungan turunan kedua dari Gaussian dan dapat dievaluasi dengan biaya komputasi yang sangat rendah menggunakan *integral images* dan terbebas dari pengaruh ukuran. Representasi dari

perhitungan determinan matriks Hessian ditunjukkan oleh Persamaan (2.19) (Bay, Tuytelaars, & Gool, 2006):

$$\det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad ..(2.19)$$

Scale spaces biasanya diimplementasikan sebagai *image pyramids*. Gambar-gambar dihaluskan secara berulang dengan Gaussian dan kemudian dilakukan *sub-sampling* dengan tujuan untuk mendapatkan level piramida yang lebih tinggi. Karena penggunaan *box filter* seperti yang ditunjukkan Gambar 2.5 dan *integral image*, SURF tidak perlu berkali-kali mengaplikasikan filter yang sama ke hasil *layer* yang telah difilter sebelumnya. Hal ini dapat disimpulkan bahwa analisis *scale spaces* dilakukan dengan cara melakukan pembesaran (*up-scaling*) filter, bukan mengurangi ukuran gambar secara berulang (Bay, Tuytelaars, & Gool, 2006).



Gambar 2.5 *Box Filter*
(Bay, Tuytelaars, & Gool, 2006)

Selanjutnya, untuk pembuatan *descriptor* terdiri dari dua langkah yaitu memastikan sebuah orientasi yang dapat dibuat direproduksi dari informasi yang didasarkan dari daerah berbentuk lingkaran yang mengelilingi titik fitur, lalu membuat daerah berbentuk persegi yang tegak lurus dari orientasi yang dipilih dan mengekstrak SURF *descriptor* dari daerah tersebut. SURF juga memiliki bentuk *descriptor* yang invariant pada rotasi gambar yang bernama U-SURF (Upright

SURF) yang dapat dikomputasi lebih cepat dan cocok untuk diaplikasikan saat posisi kamera konsisten horizontal (Bay, Tuytelaars, & Gool, 2006).

2.7 *F-Measure*

F-Measure (*F-score*) dalam penelitian ini digunakan sebagai metode evaluasi. *F-Measure* memiliki rumus sebagai berikut (Kusnadi dkk., 2019):

$$F\ measure = 2\ \times \frac{(Recall\ \times Precision)}{(Recall\ + Precision)} \quad ..(2.20)$$

Di dalam *F-Measure*, terdapat perhitungan atas *Precision* dan *Recall*. *Recall* mengukur dalam kelas (*output*) yang bernilai positif, berapa banyak yang berhasil diprediksi secara benar (Shung, 2018). Sementara, *Precision* mengukur dalam kelas yang telah diprediksi bernilai positif, berapa banyak yang sebenarnya bernilai positif (Shung, 2018). *Precision* dan *Recall* memiliki rumus dasar sebagai berikut (Shung, 2018):

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad ..(2.21)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad ..(2.22)$$

Bagian-bagian dari *Precision* dan *Recall* ini dapat diletakkan dalam sebuah *Confusion Matrix* yang merupakan sebuah pengukuran performa untuk sebuah masalah klasifikasi yang dimana hasil keluarannya dapat terdiri dari dua kelas atau lebih (Narkhede, 2018). Dalam *Confusion Matrix* ini, bagian baris dari matriks berkorespondensi pada kelas prediksi dan bagian kolom berkorespondensi pada kelas aktual (Brownlee, 2016).

Tabel 2.2 *Confusion Matrix* (Narkhede, 2018)

		Actual Values	
		Positive(1)	**Negative(0)**
Predicted Values	**Positive(1)**	True Positive	False Positive
	Negative(0)	False Negative	True Negative

Dalam *image processing*, *Recall* dan *Precision* didasarkan pada jumlah dari pasangan (*matches*) yang benar dan salah di antara dua gambar. *Recall* adalah jumlah dari bagian yang dipasangkan (*matched*) secara benar dengan memperhatikan jumlah dari daerah yang berkorespondensi diantara dua gambar (Tao dkk., 2010). Sementara, *Precision* adalah jumlah bagian yang dipasangkan (*matched*) secara benar yang relatif dengan jumlah semua yang dipasangkan. *Precision* dan *Recall* dalam *image processing* memiliki rumus sebagai berikut (Kusnadi dkk., 2019):

$$precision = \frac{correct\ matches}{Total\ No.\ All\ matches} \quad ..(2.23)$$

$$recall = \frac{correct\ matches}{Total\ No.\ Correspondences} \quad ..(2.24)$$