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.

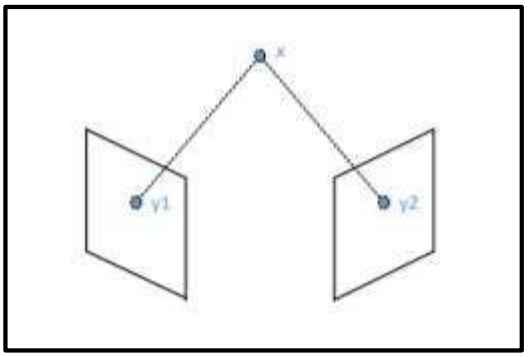
Perpustakaan Univeristas Multimedia Nusantara

BAB II

TINJAUAN PUSTAKA

2.1 Epipolar Geometry

Titik x diproyeksikan diambil dari sisi kiri dan kanan, menghasilkan dua gambar yang sesuai poin y1 dan y2. Gambar 2.1 adalah kasus geometri epipolar yang ideal. Diperlukan letak titik yang sama bila sebuah gambar diambil dari sisi kiri dan kanan untuk dapat merekonstruksi model 3D (Kusnadi et al., 2018).



Gambar 2.1 Epipolar Geometry

Poin gambar yang sesuai dapat digunakan oleh teori epipolar geometri untuk menemukan koordinat titik wajah dalam proses rekonstruksi model 3D wajah. Gambar 2.1 menggambarkan epipolar geometri dari sepasang model kamera karena untuk mengimplementasikan teori epipolar geometri membutuhkan minimal 2 buah kamera. Bila y1 adalah kamera 1 dan y2 adalah kamera 2, maka titik x yang diproyeksikan dari sudut pandang y1 dan y2 akan memiliki koordinat titik yang sama (Kusnadi et al., 2018).

2.2 Iluminasi

Proses pengenalan wajah sangat dipengaruhi oleh perubahan pada penampilan wajah yang disebabkan oleh variasi dalam iluminasi. Variasi iluminasi berarti suatu gambar wajah terdiri dari berbagai area wajah dengan pencahayaan yang berbeda-beda. Sedangkan untuk mendapatkan hasil deteksi wajah yang baik, iluminasi pada gambar wajah harus sama rata di seluruh permukaan gambar. Iluminasi juga dapat dianggap sebagai permasalahan kompleks baik pada gambar wajah yang diambil di dalam ataupun di luar ruangan. Pengenalan wajah lebih sulit dilakukan untuk gambar dengan variasi iluminasi yang buruk. Iluminasi yang buruk berpengaruh pada hasil akurasi pendeteksian wajah sehingga harus ditangani dengan menyamaratakan iluminasi pada gambar wajah (Shah, Sharif, Raza, Murtaza, & Saeed-Ur-rehman, 2015).

Gambar 2.2 Contoh Gambar yang Sama Terlihat dari Iluminasi yang Berbeda

2.3 Konversi ColorSpace Dataset

Setiap *database* akan dipilih 3 pasang *dataset* gambar wajah 2D untuk dilakukan pengujian *keypoints*. Setiap pasangan *dataset* akan dikonversi dari *RGB channel* ke *YCbCr channel*. Proses konversi dilakukan untuk mendapatkan *file* gambar yang terkompresi namun masih dapat menyimpan sebagian besar informasi

dari gambar asli. Setelah proses konversi, *channel* yang akan digunakan dalam penelitian hanya Y *channel* saja karena sebagian besar informasi yang diterima mata manusia adalah komponen *luminance* pada Y *channel* bukan *chrominance* pada *Cb* dan *Cr Channel*. Gambar terdiri dari *pixel* digital yang direpresentasikan dalam format *RGB*, dimana 0 dan 255 masing-masing mewakili warna hitam dan putih sehingga komponen Y *channel* dapat diperoleh sesuai dengan persamaan berikut (Campos, 2016).

$$[Y] = [16] + [0.257 \quad 0.504 \quad 0.098] \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad ...(2.1)$$

2.4 Keypoints

Keypoints juga dikenal sebagai fitur lokal. *Keypoints* wajah sangat penting untuk merekonstruksi model 3D dari gambar wajah 2D. Hal tersebut telah dipelajari secara ekstensif dalam beberapa tahun terakhir. Dalam penelitian ini, terdapat 15 *keypoints* wajah yang akan dideteksi. 15 *keypoints* wajah tersebut merujuk pada suatu platform bernama Kaggle dengan data yang valid dan telah diteliti oleh banyak orang yang berkecimpung di dunia data *science* (Kaggle, 2016). Gambar wajah 2D yang digunakan diambil dari 2 sisi yaitu sisi kiri dan sisi kanan.

Tabel 15 facial keypoints dapat dilihat pada tabel 2.1. Tabel tersebut yang akan menjadi patokan dalam perhitungan jumlah *keypoints* pada gambar wajah yang diuji pada penelitain ini. Dikarenakan gambar diambil dari 2 sisi, maka perhitungan *keypoints* juga akan dilakukan pada sisi kiri dan kanan gambar wajah. Jumlah *keypoints* yang didapat dari kedua sisi wajah juga turut menjadi penentu utama dalam mendapatkan hasil dari akurasi pendeteksian *keypoints* wajah. Semakin banyak pasangan keypoints yang *match*, maka hasil akurasi pendeteksian

akan semakin tinggi pula. Sebaliknya, semakin banyak titik selain *facial keypoints* yang terdeteksi, maka hasil akurasi pendeteksian akan semakin menurun nilainya.

Tabel 2.1 15 Facial *Keypoints* (Kaggle, 2016)

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	

Keypoints menjadi titik kunci yang ideal jika memenuhi kriteria pengulangan, kekhasan/keinformatifan, variasi, kuantitas dan akurasi. Hal yang paling penting dari kriteria ini adalah pengulangan *keypoints* yang terdeteksi dari kedua sisi gambar wajah. Jika ada dua gambar dari objek yang sama maka jumlah *keypoints* yang sama dihitung pada kedua gambar (Kusnadi et al., 2018). Kualitas pengulangan *keypoints* terlihat dari nilai akurasi pendeteksian yang diperoleh dari jumlah pasangan *facial keypoints* dengan letak pengulangan yang sama dari gambar wajah kiri dan kanan. Contoh implementasi 15 *facial keypoints* dapat dilihat pada gambar 2.3.

Gambar 2.3 Implementasi 15 Facial *Keypoints*

2.5 Algoritma DCT

Pendeteksian gambar wajah dapat didukung dengan mengaplikasikan algoritma *DCT* (A.Thamizharasi & Jayasudha, 2016). Dengan merepresentasikan sebuah gambar ke dalam frekuensi domain, dapat dilakukan suatu teknik pengolahan untuk meningkatkan kualitas gambar dengan mengaplikasikan algoritma *DCT*. Frekuensi domain sebuah gambar dapat direpresentasikan ke dalam 3 pita frekuensi yaitu frekuensi rendah, sedang, dan tinggi.

Xiao-Yuan Jing (Jing & Zhang, 2004) mengusulkan algoritma *DCT* untuk mengatasi permasalahan variasi iluminasi dalam domain logaritma. Dengan fungsi *DCT* yang dapat mengatasi variasi iluminasi pada gambar wajah, *DCT* tentunya menjadi algoritma *preprocessing* yang baik sebelum pendeteksian *keypoints* dilakukan. Melalui frekuensi domain dari sebuah gambar wajah, *DCT* merepresentasikan frekuensi domain sebagai penambahan dari beberapa gelombang kosinus. Pengaturan komponen *DCT* frekuensi rendah menjadi nol sama dengan melakukan proses kompresi pada gambar wajah yang menyebabkan ukuran data gambar akan lebih kecil daripada gambar asli, sehingga dapat mempercepat jalannya *image processing* pada suatu gambar.

Penelitian ini akan mengimplementasikan DCT 2 dimensi. Definisi DCT 2 dimensi untuk gambar input A (data gambar asli) dan gambar output B (hasil DCT 2 dimensi) dapat dilihat pada persamaan 2.2 (Jain, 1989).

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\left[\frac{\pi(2m+1)p}{2M}\right] \cos\left[\frac{\pi(2n+1)q}{2N}\right], \begin{matrix} 0 \le p \le M-1 \\ 0 \le q \le N-1 \end{matrix} \quad \ldots(2.2)$$

Dengan p = 0,1,2, …, M-1, dan q = 0,1,2, …, N-1

B_{pq} merepresentasikan koefisien dalam pixel gambar setelah DCT 2 dimensi diterapkan.

Untuk perhitungan inverse DCT pada DCT 2 dimensi (*idct2*) yang digunakan pada input B (hasil setelah DCT) dan output A (hasil setelah inverse DCT) adalah

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q B_{pq} \cos\left[\frac{\pi(2m+1)p}{2M}\right] \cos\left[\frac{\pi(2n+1)q}{2N}\right], \begin{matrix} 0 \le m \le M-1 \\ 0 \le n \le N-1 \end{matrix} \quad \ldots(2.3)$$

Dengan m = 0,1,2, …, M-1, dan n = 0,1,2, …, N-1

A_{mn} merepresentasikan koefisien dalam pixel gambar yang asli. A_{mn} pada persamaan 2.3 merupakan hasil dari proses inverse DCT 2 dimensi dengan mengakses B_{pq} yang merupakan hasil dari proses DCT 2 dimensi.

Kedua persamaan diatas memiliki persyaratan variabel pq terhadap koefisien α yang digunakan.

dimana

$$\alpha_p = \left\{ \begin{matrix} \frac{1}{\sqrt{M}}, p = 0 \\ \sqrt{\frac{2}{M}}, 1 \le p \le M-1 \end{matrix} \right\}$$

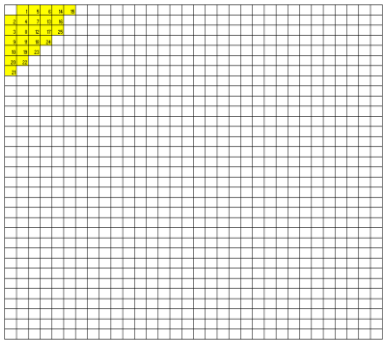
dan …(2.4)

$$\alpha_q = \begin{Bmatrix} \frac{1}{\sqrt{N}}, q = 0 \\ \sqrt{\frac{2}{N}}, 1 \ \leq p \leq N - 1 \end{Bmatrix}$$

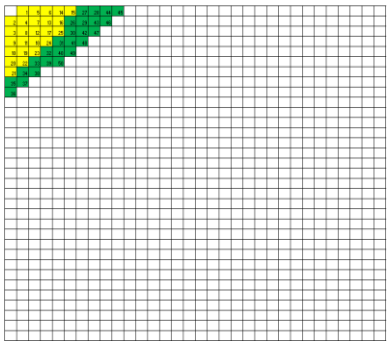
Dengan M dan N adalah ukuran matrix *DCT* 2D MxN.

Pada langkah sebelumnya komponen *DCT* 2D dengan ukuran MxN sudah dilakukan pengaturan komponen menjadi nol. Dalam kaitannya dengan variasi iluminasi, sejumlah koefisien *DCT* pada frekuensi rendah akan dihilangkan untuk meminimalkan variasi iluminasi yang ada. Meminimalkan variasi iluminasi berarti menyamaratakan iluminasi pada seluruh permukaan gambar. Hal tersebut dapat dilakukan hanya dengan mengatur komponen *DCT* yang berada pada frekuensi rendah menjadi nol. Sistem yang dihasilkan tentu akan bekerja seperti filter *high-pass* yang dapat memblokir komponen frekuensi rendah yang tidak diinginkan.

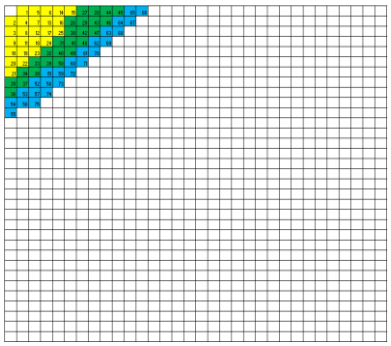
Sifat dari algoritma DCT adalah mengubah informasi gambar yang difokuskan hanya pada beberapa koefisien DCT saja. Perubahan informasi gambar tersebut dikenal sebagai proses normalisasi iluminasi yang bertujuan untuk menyamaratakan iluminasi pada gambar (Chen et al., 2006). Proses normalisasi diimplementasikan melalui koefisien *DCT* yang bernilai 25, 50 dan 75. Ketiga koefisien tersebut menetapkan jumlah koefisien *DCT* yang akan diubah menjadi nol (penghapusan komponen *DCT*). Bila difokuskan hanya untuk menangani variasi iluminasi, maka proses normalisasi iluminasi tersebut hanya dilakukan pada area frekuensi rendah saja (terdapat pada bagian kiri atas gambar). Proses normalisasi iluminasi pada frekuensi rendah dari setiap koefisien dapat dilihat pada gambar 2.4, 2.5 dan 2,6. Sedangkan contoh hasil penghapusan komponen *DCT* dapat dilihat pada gambar 2.7.



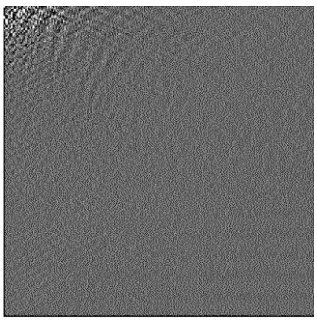
Gambar 2.4 Koefisien 25



Gambar 2.5 Koefisien 50



Gambar 2.6 Koefisien 75



Gambar 2.7 Hasil Penghapusan Komponen *DCT*

Normalisasi iluminasi terjadi pada domain logaritma dan domain frekuensi sambil menjaga fitur wajah utama agar tidak terganggu karena hanya gambar wajah yang dinormalisasi yang diinginkan dalam domain logaritma. Variasi iluminasi

dapat dikurangi secara signifikan dengan memotong koefisien frekuensi rendah *DCT* dalam domain logaritma yang setara dengan kompensasi terhadap variasi iluminasi. Hal tersebut merupakan alasan mengapa algoritma *DCT* harus diimplementasikan pada domain logaritma (Chen et al., 2006).

Koefisien *DCT* pertama pada *DCT* 2-D (yaitu komponen DC) menentukan keseluruhan iluminasi yang ada pada gambar wajah. Oleh karena itu, iluminasi yang seragam dapat diperoleh dengan mengatur koefisien DC dalam nilai yang sama seperti yang ditulis pada rumus 2.5.

$$C(0,0) = log\, \mu * \sqrt{M\ N} \qquad ...(2.5)$$

Dengan C(0,0) adalah koefisien DC dari gambar wajah dan nilai μ dipilih dari nilai yang μ asli.

2.6 Algoritma CLAHE

Metode *CLAHE* digunakan untuk menangani masalah kontras yang tidak selaras pada gambar. *CLAHE* bekerja dengan membatasi peningkatan kontras yang biasanya dilakukan oleh *Histogram Equalization*. Jadi, peningkatan kontras pada gambar wajah dapat dibatasi sesuai kebutuhan untuk menangani variasi iluminasi (Sonali, Sahu, Singh, Ghrera, & Elhoseny, 2019).

Penelitian lain turut mengusulkan *CLAHE* untuk digunakan dalam proses pendeteksian wajah (Thamizharasi & J., 2016) Hasil yang diperoleh setelah mengaplikasikan metode *CLAHE* menghasilkan nilai akurasi sebesar 0.985. Hasil akurasi yang tinggi menunjukkan bahwa gambar yang diproses dengan *CLAHE* dapat dengan baik menangani variasi iluminasi.

Adapun metode *CLAHE* dalam penelitian ini berjalan sesuai dengan tahapan berikut (Ma, Fan, Yang, Zhang, & Zhu, 2018).

1. Membaca data gambar yang akan diimplementasi algoritma *CLAHE*
2. *Clip Limit* didefinisikan sebagai N_{CL} yang memiliki range $[0,1]$.

   Kemudian pixel akan di- 'clip' sesuai dengan nilai *Clip Limit*. Jumlah total dari clipped pixel didefinisikan sebagai $N_{\sum clip}$, dan rata - rata dari pixel untuk didistribusikan ke setiap *gray levels* adalah:

   $$N_{avggray} = \frac{N_{\sum clip}}{N_{gray}} \quad \ldots(2.6)$$

   Aturan clipping histogram disertakan dalam rumus berikut

   $$If\ H_{region}(i) > N_{CL}\ then$$

   $$H_{region_clip}(i) = N_{CL} \quad \ldots(2.7)$$

   $$Else\ if\ \left(H_{region}(i) + N_{avggray}\right) > N_{CL}\ then$$

   $$H_{region_clip}(i) = N_{CL} \quad \ldots(2.8)$$

   $$Else\ H_{region_{clip}}(i) = H_{region}(i) + N_{CL} \quad \ldots(2.9)$$

   Dengan $H_{region}(i)$ dan $H_{region_clip}(i)$ adalah histogram original dan *clipped histogram* dari setiap wilayah pada gray level ke- i.

   Clip limit adalah faktor kontras yang mencegah kelebihan saturasi pada gambar terutama pada area homogen. Area homogen ditandai dengan tingginya jumlah pixel yang memiliki kisaran tingkat abu-abu yang sama. Tanpa adanya clip limit, adaptive histogram equalization dapat memberi hasil yang dalam beberapa kasus lebih buruk dari gambar asli.
3. Mendistribusikan ulang *pixel* yang tersisa hingga semua *pixel* telah berhasil didistribusikan. Langkah redistribusi *pixel* adalah sebagai berikut

$$Step = \frac{N_{gray}}{N_{remain}} \quad ...(2.10)$$

Dengan N_{remain} adalah sisa clipped *pixel*. $Step$ adalah bilangan bulat positif mulai dari angka 1. Program memulai pencarian level minimum hingga level maksimum dari *gray* level. Jika jumlah pixel pada *gray* level kurang dari N_{CL} , program akan mendistribusikan satu *pixel* ke *gray* level. Jika *pixel* tidak terdistribusikan semua sampai pencarian level selesai, program akan menghitung $Step$ baru sesuai dengan persamaan (2.10) dan memulai putaran pencarian baru hingga seluruh *pixel* tersisa didistribusikan.

4. Meningkatkan intensitas di setiap wilayah dengan distribusi *Rayleigh*. Distrubusi *Rayleigh* mengakses nilai α yang adalah parameter distribusi *Rayleigh* yang ditentukan tergantung pada setiap input gambar. Nilai α yang lebih tinggi akan menghasilkan peningkatan kontras yang lebih signifikan pada gambar.

2.7 Detektor Fitur

Penjelasan mengenai 5 detektor fitur yang akan digunakan dalam penelitian ini adalah sebagai berikut.

2.7.1 SURF

SURF menggunakan pendekatan matriks *Hessian* yang beroperasi pada gambar integral untuk menemukan letak *keypoints* pada gambar sehingga dapat meningkatkan efisiensi waktu dalam proses perhitungan yang perlu dilakukan. Diberikan sebuah titik $X = (x, y)$ dalam sebuah gambar *I*, *Hessian* matriks $H(x, \sigma)$ dalam X pada skala σ didefinisikan seperti berikut.

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X,\sigma) & L_{xy}(X,\sigma) \\ L_{xy}(X,\sigma) & L_{yy}(X,\sigma) \end{bmatrix} \quad ...(2.11)$$

Dimana $L_{xx}(X,\sigma)$, $L_{xy}(X,\sigma)$, dan $L_{yy}(X,\sigma)$ merupakan konvolusi orde kedua dari Derivatif Gaussian pada gambar di titik $X = (x, y)$.

SURF terdiri dari detektor *keypoints* beserta deskriptornya. Detektor tersebut menempatkan *keypoints* pada gambar, sedangkan deskriptor menggambarkan fitur dari *keypoints* serta menciptakan vektor fitur dari *keypoints* tersebut. Secara konseptual, deskriptor pada detektor SURF dengan ukuran 64 dimensi turut fokus pada distribusi spasial dari informasi gradien dalam *keypoints* di sekitar area gambar.

Deskriptor yang dimiliki oleh SURF adalah untuk mengatasi rotasi, kecerahan, dan setelahnya direduksi menjadi satuan panjangnya nilai kontras. Dikarenakan deskriptor SURF sebagian besar didasarkan pada perbedaan intensitas, kinerja dalam proses perhitungan menjadi lebih cepat. Hal ini membuat SURF lebih unggul dalam hal pengulangan *keypoints* karena memiliki kinerja yang lebih cepat bila dibandingkan dengan detektor fitur lainnya (Carro, Larios, Huerta, Caporal, & Cruz, 2015).

2.7.2 FAST

Untuk mendeteksi fitur dari sebuah gambar, detektor *FAST* beroperasi di atas semua elemen gambar. Hal ini bertujuan untuk mengidentifikasi apakah pixel *p* dalam gambar tersebut dapat disebut sebagai fitur *keypoints* atau tidak.

Cara kerja detektor *FAST* adalah seperti yang dijelaskan berikut ini.

1. Menyebut sebuah gambar sebagai *I*
2. Intensitas pada kandidat pixel adalah *p* pada gambar *I* dan menjadi I_p
3. Tentukan sebuah nilai intensitas *Threshold* dalam nilai *T*

4. Tandai jari-jari lingkaran *Brensenham* yang terdiri dari 16 pixel dan mengelilingi pixel *p*.
5. Tandai setiap pixel pada lingkaran dari angka 1 sampai 16 dalam urutan gerakan searah jarum jam
6. Bandingkan intensitas dari pixel 1, 5, 9 dan 13 dari lingkaran dengan *Ip* dan cek apakah setidaknya tiga dari empat nilai pixel I_1, I_5, I_9 dan I_{13} berada di atas atau di bawah I_p dengan nilai *T*. Jika tidak, maka lanjut ke step a atau lanjut ke step b bila memenuhi persyaratan sebelumnya.
    a. Tolak kandidat pixel *p* dan tidak disebut sebagai fitur, kembali ke step 5
    b. Bandingkan intensitas dari semua pixel yang tersisa dan klasifikasikan pixel sebagai fitur jika sebuah set dari N (<12) pixel yang berdekatan diluar 16 pixel dari lingkaran Brensenham adalah di atas atau di bawah I_p dengan nilai *T*.
7. Ulangi prosedur untuk semua pixel yang ada pada gambar.

   Fitur yang terdeteksi disimpan dalam vektor dua dimensi, dengan jumlah baris menentukan jumlah fitur dan kolom yang mewakili koordinat pixel (Bhat, 2017).

2.7.3 Harris-Stephans

Detektor *Harris-Stephans* mendeteksi titik berdasarkan variasi intensitas dari pixel gambar yang ada. Daerah kecil dari fitur harus menunjukkan sebuah perbedaan intensitas yang besar ketika dibandingkan dengan pixel lainnya melalui berbagai arah. Hal ini dikarenakan detektor *Harris* pada dasarnya adalah menemukan perbedaan intensitas untuk perpindahan titik (u, v) di semua arah.

Tentukan *I* sebagai sebuah gambar. Temukan turunan x dan y dari gambar pada persamaan (2.12) dan diikuti oleh masing-masing produk turunan pada persamaan (2.13).

$$I_x \leftrightarrow \frac{\delta I}{\delta x} I_y \leftrightarrow \frac{\delta I}{\delta y} I_x I_y \leftrightarrow \frac{\delta I}{\delta x} \frac{\delta I}{\delta y} \quad ...(2.12)$$

$$I_x^2 = I_x * I_x I_y^2 = I_y * I_y I_{xy} = I_x * I_y \quad ...(2.13)$$

Hitung kovarians matrix, C

$$C = \sum_{x,y} w(x,y) \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \quad ...(2.14)$$

Lakukan perhitungan dengan menggunakan window function: w (x, y)

$$S(u,v) = \sum_{xy} w(x,y)[I(x+u, y+v) - I(x,y)] \quad ...(2.15)$$

Persamaan (2.15) dapat disederhanakan menjadi:

$$S(u,v) \approx [u \quad v] C \begin{bmatrix} u \\ v \end{bmatrix} \quad ...(2.16)$$

Dari persamaan (2.16), hitung C yang merupakan *second moment* matrix.

Hitung *Eigen Value* : jika λ cenderung bernilai 0, tidak bisa ditentukan sebagai *keypoints*. Oleh karena itu, kita membutuhkan nilai λ yang besar.

Hitung fungsi respon R:

$$R = det(C) - k\, trace(C)^2 = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad ...(2.17)$$

Jika nilai R lebih besar dari karakteristik nilai threshold, maka ditentukan sebagai berikut (Vinay, Aklecha, Meghana, Murthy, & Natarajan, 2018):

R	Conclusion
|R| is small	No *keypoints*
<0	Edge
>0	Corner

2.7.4 BRISK

BRISK mencapai akurasi yang baik sehubungan dengan pencocokan *keypoints* dan juga memiliki persyaratan komputasi yang lebih rendah bila dibandingkan dengan detektor lain. Detektor ini mendeteksi *keypoints* yang kuat untuk semua gambar yang dapat ditransformasi. Titik-titik tersebut diidentifikasi menggunakan kriteria penting. Untuk mencapai hasil komputasi yang lebih baik, *keypoints* terdeteksi pada lapisan oktaf piramida gambar serta di lapisan di antaranya. Menggunakan fungsi kuadrat yang sesuai dengan lokasi dan skala, *keypoints* diperoleh dalam domain kontinu.

Pola pengambilan sampel yang terdiri dari titik-titik yang terletak pada lingkaran konsentris diterapkan pada area sekitar titik untuk mengambil nilai abu-abu (*gray values*). Pola pengambilan sampel *BRISK* yang berorientasi digunakan untuk mendapatkan hasil yang dikumpulkan di deskriptor. Deskriptor *BRISK* bersifat biner sehingga pencocokan letak *keypoints* dilakukan dengan sangat efisien (Vinay, Deshpande, et al., 2018).

Penempatan pola yang dilakukan oleh detektor *BRISK* dapat dinyatakan dalam persamaan berikut.

$$\frac{N(N-1)}{2} \quad \text{...(2.18)}$$

Nilai intensitas yang digunakan untuk menghitung gradien lokal $g(p_i,p_j)$ adalah $I(p_i,\sigma_i)$ dan $I(p_j,\sigma_j)$ dengan persamaan:

$$g(p_i.p_j) = (p_i - p_j).\frac{I(p_j,\sigma_j) - I(p_i,\sigma_i)}{\|p_i - p_j\|} \quad ...(2.19)$$

Untuk mengaplikasikan pola rotasi, maka *keypoints* akan dirotasi sebesar α

$$\alpha = atan2(g_y, g_x) \quad ...(2.20)$$

Setelah melalui tahapan diatas, deskriptor *BRISK* akan melakukan pencocokan letak *keypoints* dari jarak yang terdekat terlebih dahulu (Leutenegger Stefan, Margarita Chli, 2011).

2.7.5 Minimum Eigen Value

Detektor *Minimum Eigen Value* dikembangkan oleh Jianbo Shi dan Carlo Tomasi dan karena itu disebut juga detektor titik sudut Shi-Tomasi. Metode pendeteksian fitur ini menonjol dari yang lain karena menggunakan perbedaan fitur untuk memantau kualitas fitur gambar. Residu *RMS* dari frame pertama dan saat ini mengkuantifikasi perubahan dalam penampilan fitur. Ini menekankan pada fakta bahwa fitur ini dikategorikan sebagai fitur yang bagus jika pelacak *keypoints*nya berfungsi dengan baik.

Berbeda dengan *Harris-Stephans*, *Minimum Eigen Value* secara langsung menghitung *R* dengan mengakses dua nilai *eigen minimum*, karena untuk proses pelacakan *keypoints*, sudut-sudut gambar lebih stabil dengan mempertimbangkan asumsi tertentu (Vinay, Aklecha, et al., 2018).

$$R = min(\lambda_1, \lambda_2) \quad ...(2.21)$$

2.8 Evaluasi Kriteria

Untuk mengevaluasi kinerja fitur detektor, nilai pengulangan pendeteksian *keypoints* akan dihitung. Hal tersebut didasarkan pada jumlah *correct match* dan *correspondent* yang diperoleh dari kedua sisi gambar wajah. Untuk perhitungan akurasi atau yang disebut dengan nilai *F-score* menggunakan *recall* dan *precision*.

Pada persamaan *recall* dan *precision*, terdapat *true positive* (*tp*), *false positive* (*fp*) dan *false negative* (*fn*). *Correct match* dinyatakan sebagai *true positive* yang menandakan jumlah *keypoints* yang berada pada posisi yang sama antara gambar wajah kiri dan kanan. Sedangkan *correspondent* dinyatakan sebagai *false positive* yang menandakan titik selain keypoint yang letaknya sama antara gambar wajah kiri dan kanan. Untuk nilai *false negative*, dilihat dari berapa jumlah pasangan titik tersisa di gambar wajah kiri dan kanan selain *correct match* yang memiliki letak berbeda atau dapat direpresentasikan juga sebagai jumlah total pasang titik yang dideteksi dari gambar kiri dan kanan wajah (pada kasus penelitian ini *correct match* + *fn* adalah 15 pasang titik).

Recall dihitung dari jumlah *correct match* dibagi dengan jumlah total pasangan titik yang dideteksi (Manning et al., 2002).

$$recall = \frac{tp}{tp+fn} = \frac{Correct\ Match}{Total\ Number\ of\ All\ Matches\ Detected} \quad \ldots(2.22)$$

Precision dihitung dari jumlah *correct match* dibagi dengan jumlah semua *correct match* dan *correspondent* yang terdeteksi (Manning et al., 2002).

$$precision = \frac{tp}{tp+fp} = \frac{Correct\ Match}{Correct\ Match+Correspondent} \quad \ldots(2.23)$$

F-score dihitung dari nilai *recall* dan *precision* yang telah didapat (Kusnadi et al., 2018). Nilai ini juga yang akan menjadi nilai hasil akurasi pendeteksian *keypoints* pada akhir penelitian.

$$F\ Score\ = 2x\frac{(RecallxPrecision)}{(Recall+Precision)} \quad \ldots(2.24)$$