

## SISTEM OTOMATISASI BLUR CITRA WAJAH MENGGUNAKAN GAUSSIAN FILTER DAN DLIB CNN

Audrey Nauffal Juniar<sup>1</sup>, Sultan Mahdi<sup>2</sup>, Hendra<sup>3</sup>, Galuh Larasati Wardhani<sup>4</sup>, Raudah<sup>5</sup>,

<sup>1,2</sup>PT Phoenix Resources International, Tarakan, Kalimantan Utara, Indonesia

<sup>3,4,5</sup>Universitas Borneo Tarakan, Tarakan, Kalimantan Utara, Indonesia

\*<sup>1</sup>audreynauffal1206@gmail.com

<sup>2</sup>sultanmahdi43@gmail.com

<sup>3</sup>abdillahhendra794@gmail.com

**Abstract**—The purpose of writing this paper is to test an image processing-based facial blur automation system using OpenCV DLib CNN against several conditions such as the large number of faces in the image, the position facing the face, and when the face is blocked by a mask. The system process starts with inputting the image, resizing the image, turning the image gray and equalizing the image. Then face detection will be carried out with CNN Dlib where the detected face in the image will proceed to the blur process on the detected face. As a result of this writing, the system is able to detect faces both in the large number of faces in the image, the face position is 90 degrees from the front face, or when the face is blocked by a mask. And also there is an error when there are many faces in the image and there is an obstacle on one of the faces that causes the face not to be blurred.

**Keywords**— OpenCV, Face Detection, CNN.

**Intisari**—Penulisan ini bertujuan untuk menguji sistem otomasi blur wajah berbasis pengolahan citra menggunakan OpenCV DLib CNN terhadap beberapa kondisi seperti banyaknya jumlah wajah pada citra, posisi hadap dari wajah, dan juga ketika wajah terhalang oleh masker. Proses sistem ini diawali dengan menginput citra, *resize* citra, mengubah citra menjadi *graycale* dan ekualisasi citra. Kemudian akan dilakukan pendeteksian wajah dengan Dlib CNN dimana wajah yang terdeteksi pada citra akan dilanjutkan ke proses blur pada bagian wajah yang terdeteksi. Hasil dari penulisan ini, sistem mampu mendeteksi wajah baik pada jumlah wajah pada citra yang banyak, posisi wajah 90 derajat dari wajah depan, maupun saat wajah terhalang oleh masker. Dan juga terdapat error ketika terdapat banyak wajah pada citra dan terdapat halangan pada salah satu wajah yang menyebabkan wajah tersebut tidak ter-blur.

**Kata Kunci**—OpenCV, Deteksi Wajah, CNN

### I. PENDAHULUAN

*Blurring* citra sering kita lihat di televisi untuk mengaburkan atau men-sensor konten-konten negative atau untuk melindungi identitas sesuatu. Salah satunya adalah *blurring* pada wajah. Dalam Pedoman Prilaku Penyiaran (P3), anak-anak atau remaja dalam peristiwa hukum baik sebagai pelaku maupun korban wajib disamarkan identitasnya. Selain anak-anak, seseorang yang masih berstatus tersangka juga harus dijaga identitasnya. Selain dari dunia jurnalistik, *blurring* wajah juga kita temukan di bidang lain, contohnya *street view* pada Google

*Maps*. Wajah semua orang yang tertangkap kamera di-blur wajahnya untuk menjaga identeitas mereka.

Proses blur ini biasanya dilakukan secara manual sehingga pada bidang seperti jurnalistik suatu kejadian tidak dapat ditampilkan secara langsung karena harus melalu proses *blurring* secara manual. Untuk memudahkan pekerjaan tersebut kita dapat membuat suatu program yang dapat mendeteksi wajah dari segala sudut yang kemudian akan di blur sehingga suatu kejadian dapat langsung disiarkan secara langsung dengan menghilangkan proses *blurring* secara manual. OpenCV adalah salah satu *library* yang dapat membantu karena dapat mengolah citra. Dengan memberi beberapa sample wajah, program tersebut dapat mendeteksi wajah dengan otomatis. Kemudian untuk mengaburkan citra wajah tersebut, terdapat *library* khusus untuk melakukan pekerjaan itu.

### II. LANDASAN TEORI

#### A. Citra Digital

Berdasarkan kamus Webster, citra merupakan representasi, kemiripan, atau imitasi dari suatu objek atau benda. Sebuah citra digital dapat mewakili matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel (*pixel = picture element*). Piksel merupakan elemen terkecil dari suatu citra. Sebuah piksel terdiri dari dua parameter, yaitu koordinat dan intensitas atau warna [1].

Secara matematis fungsi intensitas cahaya pada bidang dua dimensi didefinisikan sebagai  $f(x, y)$  dan koordinat bidang dua dimensi didefinisikan  $(x, y)$ . Citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit agar dapat diolah komputer. Pada citra yang berukuran  $N \times M$  dapat dinyatakan dengan matriks sebagai berikut : [2]

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (1)$$

Suatu koordinat titik citra dinyatakan dengan indeks baris ( $i$ ) dan indeks kolom ( $j$ ), sehingga intensitas derajat keabuan pada titik ( $i,j$ ) dinyatakan dengan  $f(i, j)$ . Bentuk matriks dari persamaan (1) dapat dituliskan sebagai berikut:

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,M-1} \\ a_{1,0} & a_{1,1} & \dots & a_{0,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1,0} & a_{N-1,1} & \dots & a_{N-1,M-1} \end{bmatrix} \quad (2)$$

Dengan  $a_{i,j} = f(i,j)$  sehingga matriks pada persamaan (1) sama dengan matriks persamaan (2) [3].

#### B. Citra *Resize*

Citra *resize* merupakan proses untuk mengubah ukuran dari lebar dan tinggi sebuah citra. Proses ini berfungsi untuk mengubah citra yang terlalu besar atau terlalu kecil dari yang dibutuhkan. Dengan melakukan *resize*, ukuran beberapa citra dapat diatur sama dan kinerja sistem untuk proses lebih lanjut tidak terganggu [4]. Proses ini dibutuhkan pada sistem ini untuk mendeteksi wajah yang memiliki piksel kecil.

#### C. Citra *Grayscale*

*Grayscale* adalah citra digital yang hanya memiliki satu nilai *channel* pada setiap pikselnya, dimana nilai bagian warna merah, hijau, dan biru sama. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dihasilkan adalah warna dari hitam, keabuan, hingga putih. Tingkat keabuan merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih dan memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan) [5]. Tujuan dari *Grayscale* pada sistem ini adalah untuk mempercepat pemrosesan data untuk mendeteksi wajah tiap piksel.

#### D. Ekualisasi Citra

Ekualisasi digunakan untuk memperoleh penyebaran tingkat intensitas yang merata sehingga tiap tingkat intensitas memiliki jumlah piksel yang relatif sama. Untuk mengubah intensitas suatu piksel menjadi tingkat intensitas yang baru, menggunakan fungsi transformasi [6].

#### E. *Convolutinal Neural Network* (CNN)

CNN merupakan salah satu metode dalam *deep learning* yang menggunakan lapisan konvolusi untuk mengonvolusi suatu input dengan *filter*. CNN mempunyai dua tahapan utama yaitu *feature learning* yang terdiri dari *convolution layer*, ReLU (fungsi aktivasi) dan *pooling* dan *classification* yang terdiri dari *flatten*, *fully-connected layer*, dan prediksi. Pada setiap bagian CNN terdapat dua proses utama, yaitu *feed-forward* dan *backpropagation* [7].

#### F. *Gaussian Blur*

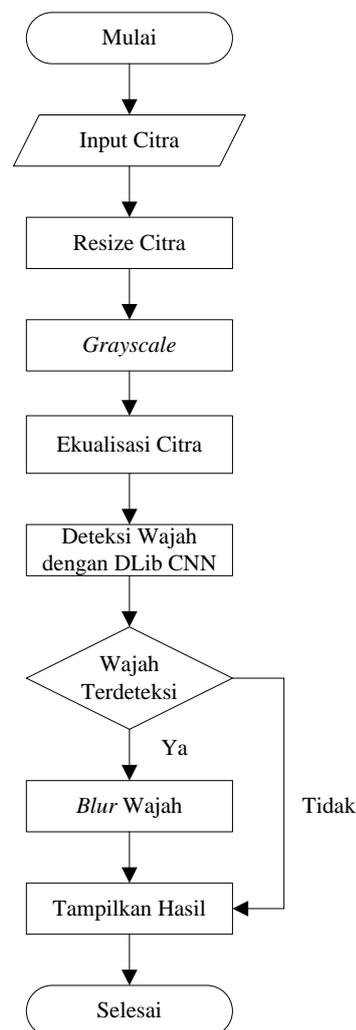
*Gaussian blur* adalah model *blur* yang memiliki fungsi kerapatan probabilitas (*probability density function / PDF*) yang diberikan oleh kurva *Gaussian*. PDF yang mewakili sifat paling acak dalam bentuk satu dimensi seperti berikut:

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (3)$$

Dimana  $\mu$  adalah nilai rerata dan  $\sigma$  adalah deviasi standar (atau akar varians) variabel random [8].

### III. METODE PENELITIAN

Tahapan perancangan algoritma yang digunakan pada penelitian ini ditunjukkan oleh gambar diagram alir di bawah.



Gambar 1. Diagram Alir Perancangan Sistem

Input citra berupa wajah manusia dapat diambil secara langsung menggunakan kamera atau dapat menggunakan citra yang tersimpan pada komputer. Setelah itu citra akan di-*resize* agar citra wajah yang pikselnya kecil akan di-perbesar sehingga dapat terdeteksi dengan mudah.



428 640

Gambar 1. Citra input yang belum di-*resize*



550 750

Gambar 2. Hasil Citra yang telah di-resize

Gambar 2 menunjukkan hasil citra yang telah melalui proses *resize*. Sebelum dilakukan *resize* citra memiliki dimensi piksel sebesar 428 x 640, setelah dilakukan *resize* menjadi 550 x 750. Proses ini dibutuhkan pada sistem untuk mendeteksi wajah yang memiliki piksel kecil.

Kemudian citra warna diubah menjadi *grayscale* yang akan menampilkan variasi warna hitam dan putih. Warna RGB pada setiap piksel dijumlahkan kemudian dibagi tiga sehingga diperoleh nilai rata-rata *grayscale*.



550 750

Gambar 3. Hasil Citra *Grayscale*

Tujuan dilakukannya *grayscale* adalah untuk mempercepat proses pendeteksian wajah pada citra. Karena dengan dilakukan *grayscale*, setiap piksel pada citra hanya memiliki satu *channel* warna dan citra yang memiliki kombinasi warna RGB memiliki 3 *channel* warna.

Setelah itu, dilakukan proses ekualisasi pada citra untuk menyamaratakan tingkat intensitas setiap piksel. Gambar di bawah menunjukkan hasil citra yang telah melewati proses ekualisasi. Tujuan dilakukannya ekualisasi pada sistem ini adalah untuk mengatur penyebaran intensitas keabuan yang rata. Sehingga citra wajah yang ada pada sisi gelap dapat terdeteksi dengan mudah.



Gambar 4. Citra Hasil Ekualisasi

Kemudian dilakukan deteksi wajah dengan menggunakan DLib CNN yaitu salah satu metode *deep learning* untuk mendeteksi wajah secara otomatis. Deteksi wajah menggunakan DLib CNN memiliki indikator penting yaitu *pre-trained*. Pada DLib merepresentasikan kotak pembatas melalui objek persegi panjang dengan properti kiri, atas, kanan, dan bawah. DLib CNN dianggap bisa mendeteksi dari segala sudut pandang. Namun jika citra yang di-*input* tidak terdapat wajah atau tidak terdeteksi wajah maka sistem hanya akan menampilkan hasil citra hingga proses *resize*.



Gambar 5. Citra Hasil Deteksi Wajah

Setelah wajah terdeteksi maka setiap wajah akan di-*blur* menggunakan metode *Gaussian blur*. Untuk melakukan pengaburan penuh pada wajah, maka kernel pada *Gaussian blur* diatur menjadi 85 x 85.

#### IV. HASIL DAN PEMBAHASAN

Dalam perancangan sistem *blur* wajah secara otomatis ini digunakan bahasa pemrograman *Python*. Adapun hasil yang ditampilkan oleh sistem ketika citra telah diproses sebagai berikut :

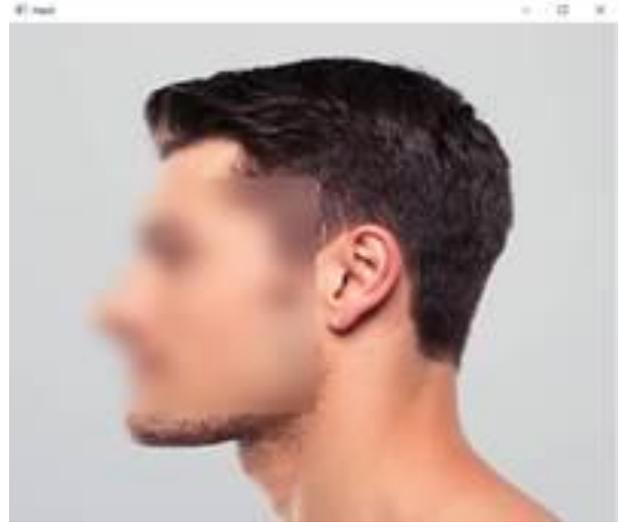


Gambar 7. Hasil Citra dengan Banyak Wajah dalam Satu Foto

Hasil deteksi wajah dengan kondisi banyak citra wajah dalam satu gambar menunjukkan metode DLib CNN dapat bekerja dengan baik. Secara keseluruhan wajah dapat dideteksi dan secara otomatis diburamkan walaupun masih terdapat kekurangan dari sistem dimana sistem tidak dapat membaca salah satu wajah dimana salah satu wajah tersebut terhalang dan terdapat *noise* pada citra wajah tersebut.

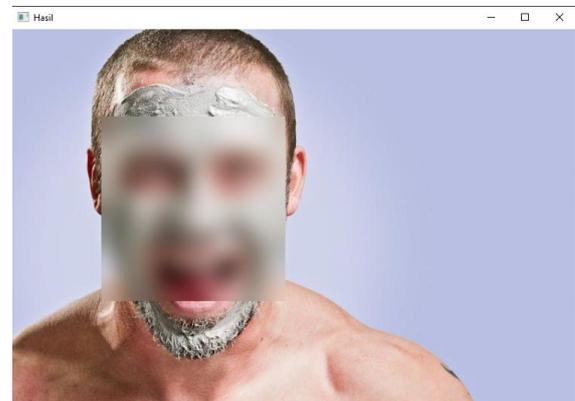


Gambar 8. Hasil Citra dengan perbedaan posisi kepala

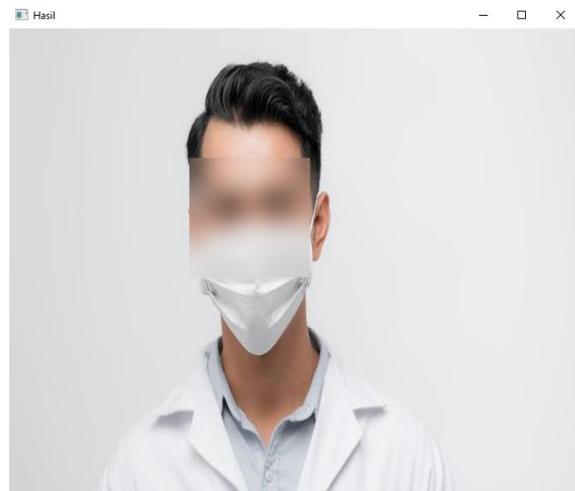


Gambar 9. Citra Hasil dengan Posisi Kepala 90 derajat

Hasil deteksi wajah dengan perbedaan posisi kepala menunjukkan bahwa DLib CNN dapat bekerja dengan baik walaupun posisi wajah berubah-ubah bahkan dengan posisi kepala 90 derajat dari arah wajah depan. Sehingga sistem dapat membuat wajah pada citra tersebut *blur*.



Gambar 10. Hasil Citra dengan Masker Wajah



Gambar 11. Citra Hasil dengan Masker Medis

Gambar di atas menunjukkan bahwa citra wajah yang menggunakan masker wajah maupun masker medis dapat mendeteksi citra wajah dan di-*blur* oleh sistem.

## V. KESIMPULAN

Sistem pendeteksi wajah ini dapat bekerja dengan baik dimana sistem ini mampu membaca atau mendeteksi wajah pada sebuah citra pada kondisi jumlah wajah yang banyak, posisi wajah 90 derajat dari wajah depan dan citra wajah yang sedang mengenakan masker wajah maupun masker medis. Namun sistem ini belum mampu mendeteksi citra wajah ketika terdapat banyak wajah dalam citra tersebut dan terdapat wajah yang terhalang seperti pada Gambar 2, dimana salah satu wajah tidak terdeteksi karena sedikit terhalang oleh kepala dari orang yang terdapat didepannya.

## UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada Tuhan yang maha esa yang memberikan kami kesempatan untuk dapat menyelesaikan penelitian ini, Mahasiswa Universitas Borneo Tarakan dan semua pihak yang telah memberikan dukungan selama pelaksanaan penelitian.

## REFERENSI

- [1] E Ardhiyanto, W Hadikurniawati, dan Z Budiarmo, "Implementasi Metode Image Subtracting dan Metode Regionprops untuk Mendeteksi Jumlah Objek Berwarna RGB pada File Video," *Jurnal Teknologi Informasi DINAMIK*, vol. 18, pp. 91-100, 2013.
- [2] D Bee, W Weku, dan A Rindengan, "Aplikasi Penentuan Tingkat Kesegaran Ikan Selar Berbasis Citra Digital," *d'CARTESIAN*, pp. 121-130, 2016.
- [3] Gonzalez R.C., R.E. Woods., dan S.L.Eddins, *Digital Image Processing 3rd Edition*, New Jersey: Pearson, 2008.
- [4] N Nurdin, D Hamdhana, dan MJ Setiawan, "Sistem Pendeteksi Pola Menggunakan Metode Peirce," *Teknik Informatika Universitas Malikussaleh*, vol. 9, 2017.
- [5] C. T. Utari, "Implementasi Algoritma *Run Length Encoding* Untuk Perancangan Aplikasi Kompresi dan Dekompresi File Citra," *Jurnal TIMES*, vol. V, pp. 24-31, 2016.
- [6] MA Bustomi, AZ Dzulfikar, "Analisis Distribusi Intensitas RGB Citra Digital untuk Klasifikasi Kualitas Biji Jagung menggunakan Jaringan Syaraf Tiruan," *JURNAL FISIKA DAN APLIKASINYA*, vol. 10, 2014.
- [7] F. I. Noviana Dewi, "Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Sistem Pengenalan Wajah," *Faktor Exacta*, vol. 14, pp. 34-43, 2021.
- [8] A. Gunadi, "Pengurangan Noise Pada Citra Digital Menggunakan Filter Aritmatik Mean, Harmonik Mean, Gaussian, Max, Min, Dan Median Dengan Membandingkan PSNR," *Jurnal Ilmu Komputer Indonesia (JIK)*, vol. 5, pp. 2615-2711, 2020.