



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.

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian ini merupakan bagian dari penelitian untuk membangun aplikasi pintar yang mampu mengenali bentuk wajah seseorang. Penelitian terdahulu yang berkaitan dengan penelitian akan dijelaskan pada Table 2.1.

Tabel 2.1. Penelitian Terdahulu

Judul Jurnal	Pembahasan	Kesimpulan
<p><i>Convolutional Neural Network Untuk Pengenalan Wajah Secara Real-Time</i></p> <p>Peneliti Muhammad Zufar, Budi Setiyono</p> <p>Lokasi Jurusan Matematika, Fakultas MIPA, Institut Teknologi Sepuluh Nopember (ITS)</p> <p>Nomor & Tahun 2337-3520 (2301-928X Print), 2016</p> <p>Nama Jurnal Jurnal Sains dan Seni ITS Vol. 5 No. 2</p>	<p>Dalam penelitian ini, menggunakan metode <i>Convolutional Neural Network</i> untuk membangun aplikasi pengenalan wajah. Pengenalan wajah bekerja secara <i>real-time</i> dalam mendeteksi wajah. <i>Dataset</i> pada penelitian dikumpulkan dari relawan sebanyak 126 gambar dengan dua kondisi pencahayaan yaitu pencahayaan terang dan redup. Hasil akurasi aplikasi dalam mendeteksi wajah mencapai 87,48% dengan pembangunan infrastruktur dari <i>Convolutional Neural Network</i> sebanyak tujuh <i>layer</i>.</p>	<p>Penelitian ini mencapai 87.48% akurasi dalam mendeteksi wajah. Hasil tersebut dipengaruhi oleh <i>dataset</i> yang diperoleh dan diolah serta permodelan infrastruktur yang dirancang. Sehingga pada penelitian ini mengambil konsep dan prinsip dari <i>Convolutional Neural Network</i>.</p>

<p>Perancangan Aplikasi Deteksi Wajah Menggunakan Algoritma Viola-Jones</p> <p>Peneliti Sandy Permata Shulur</p> <p>Lokasi Universitas Pasundan Bandung</p> <p>Nomor & Tahun 09.304.0021, 2015</p>	<p>Dalam penelitian ini, merancang sebuah aplikasi deteksi wajah menggunakan metode Viola Jones. Hasil akhir dari penelitian ini adalah aplikasi yang dapat digunakan untuk mendeteksi wajah berdasarkan <i>inputan</i> citra dari kamera ataupun dari komputer</p>	<p>Mengambil teori dan prinsip dari metode Viola Jones untuk mendeteksi wajah.</p>
<p>Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101</p> <p>Peneliti Wayan Suartika E. P, Arya Yudhi Wijaya, Rully Soelaiman</p> <p>Lokasi Institut Teknologi Sepuluh Nopember Surabaya</p> <p>Nomor & Tahun ISSN: 2337-3539, 2016</p> <p>Nama Jurnal Jurnal Teknik ITS Vol. 5, No. 1</p>	<p>Dalam penelitian ini, dilakukan klasifikasi citra sebanyak 390 citra yang terdiri dari 150 citra unggas dan sisanya adalah buaya, <i>cougar</i> dan wajah dengan masing – masing 80 citra. Dengan menggunakan metode <i>Convolutional Neural Network</i> persentase keberhasilan sebesar 20% untuk kategori unggas dan 50% untuk kategori buaya, <i>cougar</i> dan wajah</p>	<p>Pada penelitian ini mengambil konsep dari <i>Convolutional Neural Network</i>.</p>

2.2. Landasan Teori

2.2.1. Convolutional Neural Network (CNN)

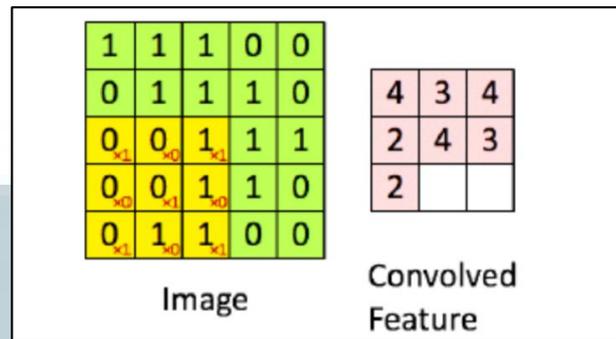
CNN merupakan pengembangan dari *Multilayer Perceptron* yang didesain untuk mengolah data dua dimensi seperti citra. *Multilayer Perceptron* (MLP) merupakan jenis model jaringan syaraf tiruan yang terdiri dari sejumlah neuron yang dihubungkan

oleh bobot – bobot penghubung. Konsep dari CNN memiliki tidak jauh berbebeda dengan MLP, letak perbedaannya adalah setiap neuron dalam CNN dipresentasikan kedalam bentuk dua dimensi, sedangkan MLP setiap neuron hanya berukuran satu dimensi. Pada kasus klasifikasi citra, MLP kurang sesuai digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik (P, Wijaya, & Soelaiman, 2016).

Terdapat tiga macam *layer* utama pada konsep CNN yaitu, *Convolution Layer*, *Subsampling Layer (Pooling Layer)* dan *Fully Connected Layer* (P, Wijaya, & Soelaiman, 2016).

1. *Convolution Layer*

Layer ini merupakan proses utama yang pertama kali menerima *input* gambar. *Convolutional Layer* terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah *filter* dengan panjang dan tinggi (*pixels*). Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada citra *inputan* disemua *offset* yang memungkinkan seperti yang ditunjukkan pada Gambar 2.1 (P, Wijaya, & Soelaiman, 2016).



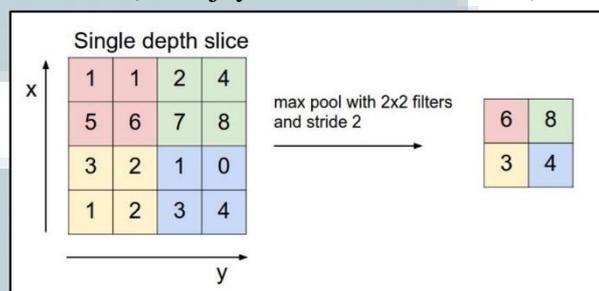
Gambar 2.1. Proses Konvolusi
Sumber: (P, Wijaya, & Soelaiman, 2016)

Kotak hijau secara keseluruhan adalah citra yang akan dikonvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah sehingga menghasilkan sebuah output atau biasa disebut sebagai *Activation Map* atau *Feature Map* (P, Wijaya, & Soelaiman, 2016).

2. *Subsampling Layer (Pooling Layer)*

Subsampling atau *pooling* adalah proses mereduksi resolusi *feature maps* yang diperoleh dari tahap konvolusi untuk mempercepat proses komputasi. Dalam sebagian besar CNN, metode *subsampling* atau *pooling* yang biasa digunakan adalah *Max Pooling*. *Max Pooling* membagi *output* dari *Feature Maps* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid*. *Grid*

yang berwarna merah, hijau, kuning dan biru merupakan kelompok *grid* yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan *grid* disebelah kanannya. Ilustrasi Maxpooling dapat dilihat pada Gambar 2.2 (P, Wijaya, & Soelaiman, 2016).



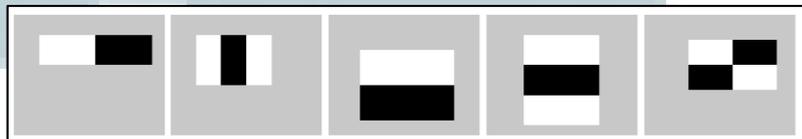
Gambar 2.2. Max Pooling
Sumber: (P, Wijaya, & Soelaiman, 2016)

3. Fully Connected Layer

Layer ini bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara *linear*. *Feature Maps* yang diperoleh perlu ditransformasi menjadi data satu dimensi terlebih dahulu agar dapat digunakan sebagai *input* pada *Fully Connected Layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak *reversibel*, *Fully Connected Layer* hanya dapat diimplementasikan di akhir jaringan (P, Wijaya, & Soelaiman, 2016).

2.2.2. Viola-Jones *Algorithm*

Viola-Jones pendeteksi objek wajah adalah pendeteksi objek pertama yang memberikan tingkat deteksi objek yang kompetitif secara *real-time* yang diusulkan pada tahun 2001 oleh Paul Viola dan Michael Jones. (Gupta & Tiwari, 2015)



Gambar 2.3. Pola Fitur Haar
Sumber: (Wang, 2014)

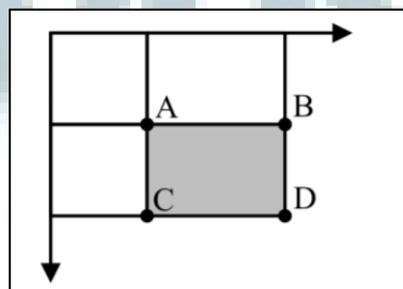
Terdapat empat konsep utama pada metode ini, yaitu *Haar-like Feature*, *Integral Images*, *Adaboost*, *Cascade Classifier*. Fitur Haar memproses setiap skala pada suatu gambar dengan mendeteksi bagian gelap dan bagian terang. Perhitungan ini membentuk satu persegi panjang di sekitar wajah yang terdeteksi. Pada Gambar 2.3. merupakan beberapa fitur Haar yang biasa digunakan adalah fitur dua persegi panjang, fitur tiga persegi panjang dan empat fitur persegi panjang (Chaudhari, sondur, & Vanjare, 2015).

Cara menghitung nilai fitur ini adalah dengan mengurangi jumlah nilai piksel pada area putih dengan jumlah nilai piksel pada area hitam (Putro, Adji, & Winduratna, 2012). Untuk mempercepat proses penghitungan nilai fitur, digunakan sebuah teknik yang disebut *Integral Image*. Nilai integral setiap piksel didapat dengan menjumlahkan seluruh nilai sebelah atasnya dan sebelah kirinya. Contoh perhitungan *Integral Image* dapat dilihat pada Gambar 2.4 (Shulur, 2015).

1	2	3	1	3	6
4	5	6	5	12	21
7	8	9	12	27	45
Citra Masukan			Citra Integral		

Gambar 2.4. Perhitungan Citra Integral
Sumber: (Shulur, 2015)

Dengan mendapatkan nilai dari citra integral maka jumlah dari seluruh piksel yang ada dalam setiap persegi panjang dapat dihitung dengan empat nilai. Nilai-nilai ini merupakan nilai piksel pada setiap sudut – sudut persegi panjang pada citra integra. Ilustrasi penentuan nilai piksel dapat dilihat pada Gambar 2.5 (Shulur, 2015).



Gambar 2.5. Penentuan Nilai Piksel
Sumber: (Shulur, 2015)

Rumus untuk menghitung jumlah nilai piksel tersebut adalah
(Shulur, 2015)

$$ABDC = D - (B + C) + A$$

Rumus 2.1. Rumus Perhitungan Nilai Piksel

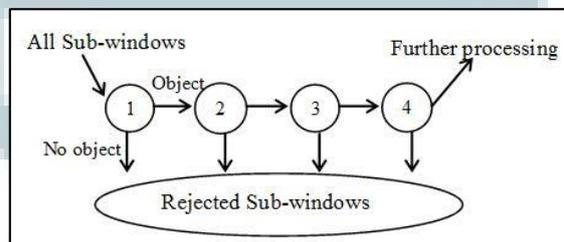
Sumber: (Shulur, 2015)

Untuk membantu memilih fitur yang terbaik atau relevan dari keseluruhan fitur, digunakan sebuah metode *machine learning* yang disebut *Adaboost*. *Adaboost* menggabungkan banyak *classifier* lemah untuk membuat sebuah *classifier* kuat. *Classifier* lemah merupakan fitur – fitur yang tergolong sebagai fitur yang baik untuk mengidentifikasi sebuah wajah dibandingkan dengan fitur – fitur lainnya. (Shulur, 2015).

Karakteristik dari metode Viola Jones adalah adanya klasifikasi bertingkat (*Cascade Classifier*). Klasifikasi pada algoritma ini terdiri dari beberapa tingkatan dan tiap tingkatan mengeluarkan subcitra yang diyakini bukan wajah. Hal ini dilakukan karena lebih mudah untuk menilai subcitra yang bukan wajah daripada menilai apakah subcitra tersebut berisi wajah (Shulur, 2015).

Pada klasifikasi tingkat pertama, tiap subcitra akan diklasifikasi menggunakan satu fitur. Hasil dari klasifikasi pertama

ini berupa T (*True*) untuk gambar yang memenuhi fitur Haar tertentu dan F (*False*) bila tidak. Klasifikasi ini kira-kira akan menyisakan 50% subcitra untuk diklasifikasi di tahap kedua. Hasil dari klasifikasi kedua berupa T (*True*) untuk gambar yang memenuhi proses *Integral Image* dan F (*False*) bila tidak. Seiring dengan bertambahnya tingkatan klasifikasi, maka diperlukan syarat yang lebih spesifik sehingga fitur yang digunakan menjadi lebih banyak. Jumlah subcitra yang lolos klasifikasi pun akan berkurang hingga mencapai jumlah sekitar 2%. Hasil dari klasifikasi terakhir berupa T (*True*) untuk gambar yang memenuhi proses *Adaboost* dan F (*False*) bila tidak. Alir *cascade classifier* dapat dilihat pada Gambar 2.6 (Shulur, 2015).



Gambar 2.6. Cascade Classifier

Sumber: (Khatta, Brunet, Dubois, & Miteran, 2011)

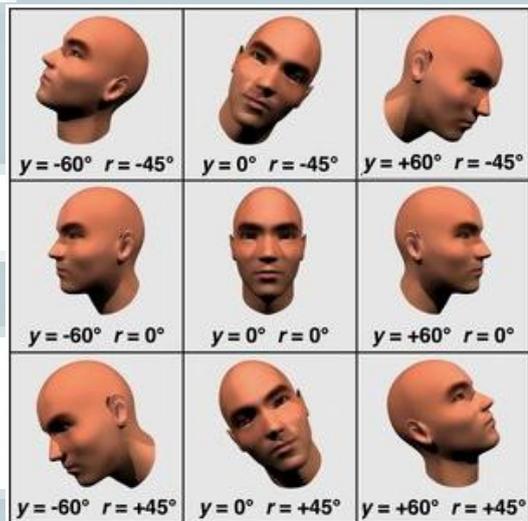
2.2.3. Face API

Sebuah wajah yang terdeteksi dapat juga ditelusuri bagian – bagian wajahnya seperti mata, hidung dan mulut. Dengan menggunakan *Face API* yang diprakarsai oleh Google, dapat secara

otomatis mencari wajah manusia dalam suatu gambar atau video (Moroney, 2015).

Dalam mendeteksi wajah, API mengembalikan hasil yaitu sebuah objek wajah. Objek wajah menyediakan data spasial untuk wajah sehingga kita dapat menggambar persegi pada sekitar wajah serta menemukan bagian – bagian wajah (Moroney, 2015).

Face API dapat dengan mudah mendeteksi wajah dalam berbagai sudut yang berbeda. Gambar 2.7. merupakan contoh ilustrasinya (Moroney, 2015).



Gambar 2.7. Face Orientation
Sumber: (Moroney, 2015)

Terdapat dua metode pemanggilan yaitu EulerY dan EulerZ. EulerY digunakan untuk mengembalikan nilai rotasi wajah disekitar sumbu vertical dan sedang EulerZ untuk mengembalikan nilai rotasi

wajah disekitar sumbu Z contohnya jika *user* memiringkan leher untuk memiringkan kepala ke samping (Moroney, 2015).

Landmark merupakan bagian - bagian yang ada pada wajah. API menyediakan metode “getLandmarks()” yang mengembalikan *list*, dimana objek – objek *landmark* mengembalikan nilai koordinat *landmark* yang terdiri dari, mulut, pipi kiri, telinga kiri, ujung telinga kiri, mata kiri, mulut kiri, pangkal hidung, pipi kanan, telinga kanan, ujung telinga kanan, mata kanan atau mulut kanan (Moroney, 2015).

2.2.4. Python

Python merupakan bahasa pemrograman yang dirancang agar mudah dipahami dan dengan fungsionalitas yang tinggi. Python dapat dijalankan dan dapat menampilkan hasil. Bahasa Python mudah dimengerti dan tidak perlu *compiling*, maka dari itu dapat menjalankan *script* Python begitu kita menyimpannya tanpa memerlukan langkah lanjutan (Joyner, 2010).

2.2.5. Android

Android merupakan *software* yang digunakan oleh perangkat *mobile* yang meliputi *middleware*, sistem operasi, dan aplikasi inti. (Murphy, 2009)

2.2.6. *Rapid Application Development*

Rapid Application Development (RAD) merupakan metode pengembangan sistem informasi dengan waktu yang singkat. RAD menggunakan metode *iterative* (berulang) dalam mengembangkan sistem dimana sistem *working* model dikonstruksikan di awal tahap pengembangan (Arora & Arora, 2016). Metode RAD memiliki tiga tahapan yaitu:



Gambar 2.8. *Rapid Application Development*

Sumber: (Kendall, 2013)

Pada tahap *Requirement Planning*, *User* dan *analyst* mengidentifikasi tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan. Pada tahap ini merupakan hal terpenting yaitu adanya keterlibatan dari kedua belah pihak.

Pada tahap *Design System*, melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *User* dan *Analyst*. Seorang *user* dapat langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain, merancang sistem dengan mengacu pada dokumentasi kebutuhan *user* yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahapan ini

adalah spesifikasi *software* yang meliputi organisasi sistem secara umum, struktur data dan yang lain.

Pada tahap *Implementation*, mengembangkan desain suatu program yang telah disetujui oleh *user* dan *analyst*. Sebelum diaplikasikan pada suatu organisasi terlebih dahulu dilakukan proses pengujian terhadap program tersebut apakah ada kesalahan atau tidak. Pada tahap ini *user* biasa memberikan tanggapan akan sistem yang sudah dibuat serta mendapat persetujuan mengenai sistem tersebut.

The logo of Universitas Muhammadiyah Negeri (UMMN) is displayed in a light blue, semi-transparent font. It features a circular emblem with a stylized face and the acronym 'UMMN' in large, bold, block letters below it.