

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Sebagai referensi untuk penelitian ini, terdapat beberapa penelitian serupa yang telah dilakukan sebelumnya :

2.1.1 Aplikasi Deteksi Tepi pada *Realtime Video* menggunakan Algoritma *Canny Detection*

Penelitian yang berjudul “Aplikasi Deteksi Tepi pada Realtime Video menggunakan Algoritma *Canny Detection*” (E. Winarno, 2011), membahas tentang pembuatan aplikasi deteksi tepi yang diterapkan pada video secara real time dan keluaran berupa deteksi tepi secara real time.

Dalam aplikasi yang menggunakan gambar video real time, algoritma Deteksi Tepi Canny dapat digunakan dalam aplikasi ini. Namun, karena perekaman secara real-time, memori komputer yang besar diperlukan untuk memproses semua data gambar video yang direkam sehingga tidak terjadi kesalahan pemrosesan.

Dari hasil penelitian tersebut, akan dilakukan Rancang Bangun di mana aplikasi tersebut akan diterapkan pada Sistem Pencatatan Keluar Masuk Kendaraan Dengan Memanfaatkan Deteksi Tepi Plat Nomor Kendaraan (Studi Kasus : Perumahan Asabri, Kota Malang) yang mana pada sistem tersebut akan dilakukan pendeteksian tepi terhadap objek yang sudah ditentukan yaitu plat nomor kendaraan, dan pengenalan karakter pada plat nomor kendaraan.

2.1.2 Pembuatan Prototipe Sistem Deteksi Plat Kendaraan Bermotor di Indonesia

Penelitian yang berjudul “Pembuatan Prototipe Sistem Deteksi Plat Kendaraan Bermotor di Indonesia” (Wihartasih & Wibawanto, 2018), bertujuan untuk mengembangkan prototipe yang mampu mengenali dan mengenali nomor polisi pada plat nomor kendaraan bermotor, yang telah disesuaikan dengan status pendaftaran kendaraan bermotor di Indonesia. Pengenalan nomor polisi pada gambar plat dilakukan dengan melengkapi prototipe dengan sistem optical character recognition (OCR) menggunakan metode perbandingan template, dengan database template gambar dalam aplikasi yang digunakan sebagai Referensi parameter pengenalan karakter pada plat nomor.

Dalam penelitian tersebut diperoleh hasil di mana prototipe tersebut layak untuk dikembangkan ke model perangkat lunak. Kesalahan pendeteksian nomor kendaraan disebabkan oleh kondisi plat yang tidak sesuai dengan standar Tanda Nomor Kendaraan Bermotor (TNKB) seperti : plat tidak memiliki tepi (*border*); plat dalam kondisi cacat, bengkok, atau rusak; huruf pada plat tidak sesuai dengan jenis huruf plat di Indonesia. Penyebab kesalahan pendeteksian nomor kendaraan yang lain adalah ketika plat kendaraan dimodifikasi dengan ditutupi oleh kaca atau plastik transparan, sehingga citra yang diperoleh kurang jelas karena terpantul oleh cahaya. Pengambilan citra plat yang miring dan/atau dalam kondisi *blur*, juga menyebabkan kesalahan dalam pendeteksian dan pengenalan karakter.

Dari hasil penelitian tersebut, akan dilakukan Rancang Bangun di mana konsep dasar pada penelitian tersebut akan diterapkan pada Sistem Pencatatan Keluar Masuk Kendaraan Dengan Memanfaatkan Deteksi Tepi Plat Nomor

Kendaraan (Studi Kasus : Perumahan Asabri, Kota Malang) yang mana pada sistem tersebut akan dilakukan pendeteksian tepi terhadap objek yang sudah ditentukan yaitu plat nomor kendaraan, dan pengenalan karakter pada plat nomor kendaraan menggunakan algoritma *Canny Edge Detection* secara *realtime*. Pada sistem yang akan dibuat juga akan diberikan fitur di mana hasil pendeteksian plat nomor kendaraan akan disimpan dalam sebuah *database*, yang nantinya *database* tersebut akan ditampilkan dalam sebuah web yang menyajikan hasil pencatatan pendeteksian plat nomor.

2.1.3 Deteksi Tanda Nomor Kendaraan Bermotor Pada Media Streaming Dengan Algoritma Convolutional Neural Network Menggunakan Tensorflow

Penelitian yang berjudul “Deteksi Tanda Nomor Kendaraan Bermotor Pada Media Streaming Dengan Algoritma Convolutional Neural Network Menggunakan Tensorflow” (Novyantika, 2018) menjelaskan mengenai sistem yang digunakan untuk mendeteksi plat nomor kendaraan pada *streaming* gambar dengan menggunakan algoritma *Convolutional Neural Network*. Penelitian ini menggunakan bahasa pemrograman Python dan *framework* TensorFlow sebagai alat bantu dalam penelitian.

Data yang digunakan dalam penelitian ini adalah data primer berupa gambar dan video yang diambil dengan kamera ponsel, kamera DSLR Canon 1200D, pengunduhan gambar melalui internet, dan *webcam* laptop. Di sisi lain, atribut yang digunakan untuk mendeteksi TNKB pada media *streaming* dikhususkan untuk gambar, video, atau *webcam* yang bersifat interaktif pada kendaraan bermotor.

Pengumpulan data dilakukan di wilayah Kabupaten Indramayu dan di wilayah Provinsi Yogyakarta. Gambar yang diambil meliputi gambar TNKB dengan warna plat hitam, merah, kuning, dan putih. Dataset yang digunakan adalah 502 citra, yang terdiri dari 472 citra untuk data latih, serta 30 citra dan 10 video untuk data uji.

Tujuan dari penelitian ini adalah untuk mengetahui nilai akurasi pengenalan plat nomor untuk citra *streaming*. Perancangan sistem yang digunakan dalam penelitian ini terbagi menjadi tiga bagian yaitu input, processing, dan output. Input yang digunakan dalam perancangan sistem ini adalah 472 gambar yang ditangkap oleh kamera ponsel untuk proses pelatihan, serta 30 gambar dan 10 video yang ditangkap oleh kamera Canon DSLR 1200D untuk proses pengujian citra melalui media *streaming*.

Proses yang digunakan dalam penelitian ini dibagi menjadi beberapa tahapan yaitu proses *input* citra, *sliding window*, ekstraksi region dan perhitungan fitur intra-regional menggunakan algoritma *convolutional neural network*. Hasil dari perancangan sistem pada penelitian ini adalah klasifikasi citra apakah area citra pada media *streaming* adalah plat nomor kendaraan.

Dari hasil analisis yang dilakukan, akurasi plat nomor kendaraan pada media *streaming* menggunakan algoritma *Convolutional Neural Network* berada pada kisaran 70-99%, dan akurasi hasil pengujian dapat dipengaruhi oleh banyaknya *dataset* dan perspektif yang berbeda dalam pada proses pelatihan.

Dari hasil penelitian ini dilakukan perancangan dengan menggunakan algoritma Canny Edge Detection untuk menambahkan pengenalan karakter pada plat nomor kendaraan dan menyimpan hasil pengenalan plat tersebut ke dalam

database. Data tersebut akan ditampilkan di web yang berisi pencatatan pendeteksian plat nomor.

2.2 Teori Terkait

2.2.1 Kendaraan Bermotor

Menurut Undang-Undang Republik Indonesia Nomor 28 Tahun 2009, Kendaraan Bermotor adalah semua kendaraan beroda dan gandengan yang digunakan pada semua jenis jalan dan digerakkan oleh peralatan teknis berupa mesin atau alat lain yang bekerja untuk mengubah suatu sumber energi tertentu menjadi energi kendaraan bermotor yang bersangkutan, termasuk alat berat dan alat besar yang dalam pengoperasiannya menggunakan roda dan mesin dan tidak tetap serta kendaraan bermotor dalam negeri. Pengertian kendaraan bermotor adalah kendaraan bermotor dengan roda dan gandengannya, yang dioperasikan pada semua jenis jalan dan kendaraan bermotor yang beroperasi di atas air dengan massa total dari GT 5 (lima ton) sampai dengan GT 7 (tujuh ton).

2.2.1.1 Tanda Nomor Kendaraan Bermotor (TNKB)

Menurut Peraturan Kepala Kepolisian Negara Republik Indonesia Nomor 5 Tahun 2012, Tanda Nomor Kendaraan Bermotor yang selanjutnya disingkat TNKB adalah tanda registrasi dan tanda pengenal kendaraan bermotor sebagai bukti izin pengoperasian kendaraan bermotor berupa pelat nomor atau bahan lain dengan spesifikasi tertentu yang dikeluarkan Polri dan memiliki kode wilayah, nomor registrasi, serta masa berlaku dan dipasang pada kendaraan bermotor.

Menurut Undang-Undang Republik Indonesia No. 22 Tahun 2009, Pasal 68, mengatur bahwa setiap kendaraan bermotor yang digunakan di jalan harus dilengkapi dengan Surat Tanda Nomor Kendaraan (STNK) dan Tanda Nomor

Kendaraan Bermotor (TNKB). STNK berisi data tentang kendaraan bermotor, identitas pemilik, nomor registrasi kendaraan bermotor, masa berlaku dan kode wilayah. TNKB harus memenuhi syarat bentuk, ukuran, bahan dan cara pemasangan.

2.2.2 Sistem

Suatu sistem pada dasarnya adalah sekelompok elemen yang berhubungan erat dan bekerja sama untuk mencapai suatu tujuan tertentu. Setiap sistem harus terdiri dari struktur dan proses. Struktur sistem adalah gabungan dari elemen-elemen yang membentuk sistem. Proses sistem menjelaskan bagaimana setiap elemen sistem bekerja untuk mencapai tujuan sistem. Setiap sistem terdiri dari beberapa sistem yang lebih kecil yang disebut subsistem.

Ada dua pendekatan untuk mendefinisikan sistem, yang pertama adalah untuk menyoroti prosedur dan menyoroti komponen atau elemen. Pendekatan sistem yang lebih prosedural mendefinisikan sistem sebagai jaringan prosedur yang saling berhubungan untuk melakukan suatu kegiatan atau mencapai tujuan tertentu. Pendekatan sistem, yang merupakan jaringan prosedur, menekankan urutan aktivitas dalam sistem. Pendekatan yang lebih sederhana atau berbasis komponen mendefinisikan sistem sebagai sekumpulan elemen yang berinteraksi untuk mencapai suatu tujuan. Suatu sistem harus memiliki tujuan tertentu. Sebagai aturan umum, tujuan ditetapkan ke perimeter yang lebih luas dan tujuan ke perimeter yang lebih terbatas. (*Konsep Sistem – Program Studi Teknologi Pendidikan Universitas Pendidikan Indonesia, n.d.*)

2.2.2.1 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu metode pemodelan visual yang digunakan dalam dunia industri untuk merancang suatu sistem seperti: mendefinisikan kebutuhan pengguna, menganalisis kinerja sistem, dan pemodelan arsitektur dalam pemrograman. Menurut (Defni, 2019), UML sendiri juga menyediakan standar penulisan sebuah sistem *blueprint*, meliputi konsep dari proses bisnis, penulisan dari kelas dalam bahasa pemrograman tertentu, diagram basis data, dan komponen yang dibutuhkan dalam sistem perangkat lunak. Diagram UML mendefinisikan diagram sebagai diagram use case, diagram kelas, diagram keadaan, diagram aktivitas, diagram urutan, diagram kolaborasi, diagram komponen, dan diagram implementasi. (Nugroho, 2010).

Dalam rancang bangun sistem pencatatan keluar masuk kendaraan dengan memanfaatkan deteksi tepi plat nomor kendaraan, akan digunakan empat diagram dari UML yaitu : *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class diagram*.

2.2.2.1.1 Use Case Diagram

Menurut (Sukamto & Shalahuddin, 2018), diagram use case adalah model perilaku (behavior) dari suatu sistem informasi. Secara umum diagram use case memungkinkan untuk mengetahui fungsi-fungsi yang termasuk dalam suatu sistem informasi dan siapa yang berhak menggunakan fungsi-fungsi tersebut. Ada tiga elemen dalam diagram use case, yaitu aktor, objek use case, dan hubungan antar objek. Aktor adalah manusia atau entitas mesin yang berinteraksi dengan sistem untuk melakukan tugas tertentu. Objek use case adalah unit aksi atau fungsionalitas sistem. Sebuah use case dapat berisi fungsi dari use case lain, sehingga duplikasi

fungsi dapat dihindari dengan mengekstraksi fungsionalitas umum (Zuhdi & Safarudin, 2019).

2.2.2.1.2 Activity Diagram

Menurut (Sukanto & Shalahuddin, 2018), Activity diagram menjelaskan alur kerja suatu sistem, atau proses bisnis, atau menu yang disertakan dalam perangkat lunak.

2.2.2.1.3 Sequence Diagram

Menurut (Defni, 2019), *Sequence Diagram* menjelaskan perilaku objek dalam use case dengan menggambarkan pesan yang dikirim dan diterima antar objek.

2.2.2.1.4 Class Diagram

Diagram kelas adalah diagram yang menggambarkan struktur sistem dan hubungan kelas yang akan digunakan saat merancang sistem. Diagram kelas ini juga menunjukkan interaksi dan/atau metode yang diatur dalam kelas-kelas, atribut yang disimpan, jenis objek dalam sistem, batasan hubungan antar objek, dll. Sedangkan diagram objek adalah diagram yang berisi gambaran umum struktur model suatu sistem (Zuhdi & Safarudin, 2019).

2.2.2.2 Database

Database adalah kumpulan dari banyak file yang saling terkait. Hubungan tersebut biasanya diwakili oleh kunci dari setiap file yang ada. Database berfungsi untuk menunjukkan kumpulan data yang digunakan oleh perusahaan.

File yang berisi record dengan tipe, ukuran, dan bentuk yang sama, merupakan kumpulan entitas yang disatukan. Record terdiri dari banyak field yang

ditautkan bersama untuk menunjukkan bahwa field tersebut dicatat dalam satu record.

Sebuah atribut digunakan oleh entitas tertentu, misalnya atribut alamat menunjukkan entitas alamat pengguna. Entitas adalah objek yang nyata dan akan disimpan datanya dalam sebuah atribut. (Kristanto, 1994).

2.2.3 Citra Digital

2.2.3.1 Definisi Citra

Citra adalah suatu cahaya dalam bidang dua dimensi. Dari sudut pandang matematika, gambar adalah fungsi kontinu dari intensitas cahaya dalam bidang dua dimensi. Ketika sumber cahaya mengenai suatu benda, benda tersebut memantulkan sebagian dari berkas cahaya. Cahaya yang dipantulkan ini ditangkap oleh alat optik seperti mata manusia, kamera, pemindai, dll., untuk dapat merekam gambar objek. Citra sebagai keluaran dari suatu sistem perekaman data dapat berupa :

- a. Optik berupa foto
- b. Analog berupa sinyal video seperti citra pada monitor televisi
- c. Digital yang dapat langsung disimpan pada suatu pita magnetik

Citra juga dapat didefinisikan sebagai fungsi dua dimensi $f(x, y)$, di mana x dan y adalah koordinat spasial dan amplitudo f pada setiap pasangan (x, y) disebut intensitas citra pada titik ini. (Iriyanto & Zaini, 2014).

2.2.3.2 Definisi Citra Digital

Menurut (E. Winarno, 2011), citra digital adalah citra yang dibuat oleh pemrosesan komputer di mana citra direpresentasikan secara digital dengan nilai diskrit. Citra digital biasanya berbentuk persegi panjang dan dimensi tinggi sama

dengan lebar, atau panjang sama dengan lebar. Citra digital dengan tinggi N , lebar M , dan nilai skala abu-abu L dapat dilihat berdasarkan fungsi:

$$f(x, y) = \begin{matrix} 0 \leq x \leq M \\ 0 \leq y \leq N \\ 0 \leq f \leq L \end{matrix}$$

Citra digital berukuran $N \times M$ biasanya dipresentasikan sebagai matriks berukuran N baris dan M kolom sebagai berikut:

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

2.2.3.3 Grayscale

Grayscale merupakan proses awal yang paling sering dilakukan dalam pengolahan citra dengan tujuan untuk menyederhanakan model citra. Secara umum citra RGB terdiri dari 3 lapisan matriks yaitu R-layer, G-layer, dan B-layer. Jika setiap proses perhitungan dilakukan dengan tiga lapisan, berarti akan dilakukan tiga perhitungan yang sama. Dengan mengubah 3 level tersebut menjadi matriks 1 level akan memiliki hasil citra grayscale. Citra ini tidak memiliki unsur warna, tetapi memiliki derajat keabuan. (Pemayun et al., 2015).

2.2.3.4 Deteksi Tepi (Edge Detection)

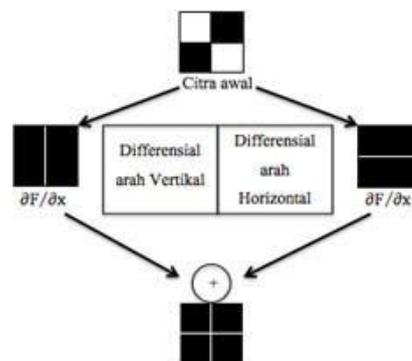
Deteksi tepi merupakan suatu metode untuk mengidentifikasi perubahan intensitas yang berbeda secara nyata pada sebuah citra (Sutoyo, 2009). Pada penelitian lain ditemukan operator edge detection sebagai operasi lingkungan, yaitu operasi yang mengubah nilai keabuan suatu titik sesuai dengan nilai keabuan titik-titik sekitarnya, masing-masing operator memiliki nilai keabuan sendiri-sendiri. (Wibowo et al., 2013).

Tujuan deteksi tepi suatu citra antara lain (Yodha & Kurniawan, 2014):

- a. Menandai bagian detail citra.
- b. Memperbaiki detail citra yang kabur.
- c. Adanya efek proses akuisisi citra.
- d. Mengubah citra 2D menjadi kurva

Gambar berikut ini menunjukkan bagaimana tepi dalam gambar diperoleh.

Semua tepi suatu citra dapat diperoleh dengan menggabungkan tepi yang diperoleh dari garis horizontal dan garis vertikal.



Gambar 2.1 Proses Deteksi Tepi

2.2.3.5 Improved Canny Detection Algorithm

Algoritma Canny tradisional menggunakan Gaussian filtering dan persamaan perbedaan orde pertama untuk menghitung amplitudo gradien, yang menghasilkan anti- noise yang buruk dan rentan terhadap pseudo-edge. Juga, nilai ambang batas (thresholds) yang digunakan tidak kondusif untuk deteksi tepi dalam situasi yang kompleks. (Feng et al., 2017).

Improved Canny Detection Algorithm meningkatkan operator penghitungan gradien gambar yang lebih kuat terhadap *noise* dan mempertahankan tepian detail yang berguna. Terdapat 4 langkah dalam algoritma Canny tradisional yaitu : *Image*

Filtering, Image Gradient Calculation, Non-Maximum Suppression, dan Double Thresholding.

Perbedaan antara algoritma Canny tradisional dengan *Improved Canny Detection Algorithm* terdapat pada tahap *image filtering*, perbaikan gradient amplitude dan perhitungan arah, perbaikan nilai *Threshold*. Pada algoritma Canny tradisional tahap *image filtering* dilakukan dengan menggunakan Gaussian Filtering yang mana cara tersebut membuat detail tepi terlihat kabur. Untuk mengatasi masalah tersebut metode Median Filter dapat dilakukan untuk mempertahankan detail gambar dan menghilangkan *noise*. Median Filter dibuat menggunakan kuadrat jarak Euclidean untuk meningkatkan tepi gambar. Filter tersebut dihitung dengan menggunakan rumus sebagai berikut :

$$f(x, y) = \sqrt{f_{xy}^2(x, y) + f_{45^\circ 135^\circ}^2(x, y)}$$

$$f_{xy} = \begin{bmatrix} 0 & 2 & 0 \\ 2 & -1 & 2 \\ 0 & 2 & 0 \end{bmatrix} \quad f_{45^\circ 135^\circ} = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 2 \end{bmatrix}$$

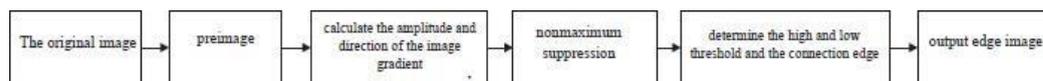
Sedangkan untuk perbaikan perhitungan arah gradien dihitung dengan rumus sebagai berikut :

$$D_A(x, y) = \arctan[A_x / A_y]$$

Untuk perbaikan nilai *Threshold*, *Improved Canny Detection Algorithm* menggunakan rumus sebagai berikut :

$$T_h = \frac{1}{2 \times n} \sum_{i=1}^n t_i$$

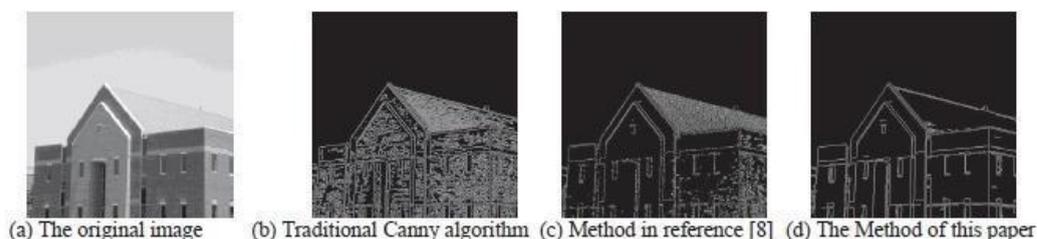
Secara singkat *Improved Canny Detection Algorithm* memiliki alur proses seperti pada gambar berikut.



Gambar 2.2.2 *Improved Canny Detection Algorithm*

Perbedaan antara algoritma Canny tradisional dengan *Improved Canny Detection Algorithm* dapat dilihat pada gambar berikut. Pada gambar tersebut terlihat bahwa algoritma Canny tradisional memang dapat mendeteksi hampir seluruh tepi yang terdapat pada citra asli, termasuk tepi tekstur pada objek citra tersebut. Hasil deteksi tepi yang terlalu akurat hingga mampu mendeteksi tepian tekstur objek citra membuat hasil deteksi tepi kurang dapat dibedakan antara tepian bentuk utama objek dengan tepian tekstur objek pada citra.

Namun pada *Improved Canny Detection Algorithm*, deteksi tepi yang dihasilkan hanyalah tepian utama objek pada citra, yang mana pada hasil deteksi tepi ini memiliki jauh lebih sedikit *noise* yang dihasilkan dari pada deteksi tepi tradisional.



Gambar 2.3 Perbandingan Traditional Canny Algorithm dengan Improved Canny Detection Algorithm

2.2.4 Media Streaming

Media Streaming adalah sebuah *file* media yang diterima dan disajikan kepada pengguna saat sedang streaming oleh penyedia. Kata *stream* mengacu pada proses penyajian media dengan cara seperti ini. Istilah media *streaming* menjadi semakin sering dipakai di jaringan internet. *Media streaming* makin marak

digunakan sebab sejak akhir tahun 1990-an dan 2000-an terjadi peningkatan teknologi internet, seperti *bandwidth* jaringan lebih baik dan akses internet yang lebih merata di wilayah-wilayah terpencil. (Edy Winarno, 2015).

Media *streaming* memiliki karakteristik sebagai berikut (Tim Digital Studio, 2005) :

1. Teknologi media *streaming* memungkinkan akses *real-time* ke konten audio, video, dan media melalui internet atau intranet.
2. Konten media *streaming* yang ditransmisikan oleh aplikasi server media, diproses, dan diputar ulang oleh aplikasi klien saat diterima, tanpa menyalin konten ke perangkat penerima.
3. File *streaming* diterima, diproses, dan/atau diputar langsung secara bersamaan, tanpa salinan konten yang tersisa di perangkat penerima.

2.2.5 Python

Dilansir oleh Wikipedia (*Python (Programming Language) - Wikipedia*, n.d.) Python adalah bahasa pemrograman yang ditafsirkan, tingkat tinggi, dan tujuan umum. Dikembangkan oleh Guido van Rossum dan pertama kali diterbitkan pada tahun 1991, filosofi desain Python menekankan keterbacaan kode dengan penggunaan substansial spasi. Konstruksi bahasa dan pendekatan berorientasi objek dirancang untuk membantu pemrogram menulis kode yang jelas dan logis untuk proyek besar dan kecil.

Python ditulis dan dikompilasi secara dinamis. Python mendukung beberapa paradigma pemrograman termasuk pemrograman terstruktur, pemrograman berorientasi objek, dan fungsional. Python sering digambarkan sebagai bahasa "*batteries included*" karena *library* standarnya yang komprehensif.

Python translator tersedia untuk banyak sistem operasi. Komunitas programmer di seluruh dunia mengembangkan dan memelihara CPython, implementasi referensi yang gratis dan open source. Sebuah organisasi nirlaba, Python Software Foundation, mengelola dan mengarahkan sumber daya untuk pengembangan Python dan CPython.

2.2.6 API

API atau (*Application Programming Interface*) adalah sebuah aplikasi perangkat lunak yang berfungsi agar pengembang untuk menggabungkan dan menghubungkan dua aplikasi yang berbeda secara bersamaan untuk saling terhubung satu sama lain. Tujuan dari penggunaan API adalah mempercepat proses pengembangan aplikasi dengan cara menyediakan sebuah metode yang terpisah sehingga *developer* tidak perlu lagi membuat fungsi yang serupa (Intermedia, 2020).

2.2.7 Open CV

OpenCV (Open Source Computer Vision Library), merupakan sebuah *library open source* yang dikembangkan oleh Intel yang berfokus pada penyederhanaan pemrograman yang terkait dengan citra digital (Sidharta, 2017). OpenCV memiliki banyak fitur, antara lain: pengenalan wajah, pelacakan wajah, pengenalan wajah, Kalman filtering, dan berbagai jenis metode AI (Artificial Intelligence). OpenCV menyediakan beberapa algoritma sederhana yang terkait dengan computer vision sederhana untuk API tingkat rendah. OpenCV adalah library computer vision yang bersifat open source untuk bahasa pemrograman C/C++ dan dikembangkan dengan Python, Java, Matlab.

Berikut ini merupakan fitur utama dari OpenCV :

1. Input-output gambar dan video
2. *Computer Vision* secara umum dan pemrosesan citra digital (untuk level API rendah dan menengah)
3. Modul *computer vision* tingkat lanjut
4. AI dan metode pembelajaran mesin
5. Pengambilan sampel dan konversi citra
6. Metode untuk membuat dan menganalisis gambar biner
7. Metode untuk mempertimbangkan model 3D