

BAB 2

DASAR TEORI

2.1 KAJIAN PUSTAKA

N. A. Zulfa dan M. Mujazi pada tahun 2021 melakukan penelitian tentang “Hubungan Pola Tidur Terhadap Konsentrasi Belajar Peserta Didik di SDN Kembangan Utara” menggunakan metode penelitian kuantitatif disertai dengan observasi analitik. Dari hasil studi tersebut, dapat disimpulkan bahwa masalah tidur atau pola tidur yang tidak baik dapat mengganggu kemampuan seseorang untuk fokus dalam pekerjaan atau belajar, dan juga dapat menyebabkan perasaan sedih dan lelah. Tidur memiliki manfaat yang sama pentingnya dengan olahraga dan nutrisi yang seimbang untuk kesehatan tubuh, karena dapat membantu tubuh memproses informasi dan menjaga metabolisme. Ketika seseorang cukup tidur, mereka akan merasa segar, bertenaga, dan menjaga kesehatan dengan baik [14].

G. T. Tjahjono dan D. Maulina pada tahun 2018 melakukan penelitian yang berjudul “Kelalaian Manusia (*Human Error*) Dalam Kecelakaan Lalu Lintas : Analisis Berdasarkan Pemrosesan Informasi” menggunakan metode analisis berdasarkan pemrosesan informasi. Hasil penelitian ini mengidentifikasi beberapa faktor penyebab tingginya angka kematian akibat kecelakaan lalu lintas di Indonesia. Faktor-faktor tersebut meliputi faktor fisik lingkungan jalan, kendaraan, dan manusia. Beberapa contoh kesalahan manusia yang dapat menyebabkan kecelakaan meliputi *recognition error* (kesalahan dalam mengidentifikasi objek), *decision error* (kesalahan dalam mengambil keputusan), *executive error* (kesalahan dalam tindakan), serta kesalahan lainnya seperti kelelahan atau mengantuk. Salah satu contoh *recognition error* adalah ketidakmampuan pengemudi untuk berkonsentrasi karena gangguan internal maupun eksternal. Sementara itu, kesalahan dalam pengambilan keputusan bisa terjadi jika pengemudi mengemudi terlalu cepat atau salah menentukan jarak dan kecepatan kendaraan sendiri dengan kendaraan lain [6]. Kesalahan ini bisa sangat merusak atau mengancam jiwa saat mengemudi. Untuk menghindari hal tersebut, sangat penting bagi pengemudi atau penumpang untuk tetap fokus dan waspada setiap saat.

T. A.-S. Zulkhaidi, E. Maria, dan Yulianto pada tahun 2019 melakukan penelitian yang berjudul “Pengenalan Pola Bentuk Wajah dengan *OpenCV*” menggunakan modul *OpenCV* dengan menggunakan bahasa pemrograman *Python*. Penelitian ini menggabungkan pengenalan pola wajah dengan modul *OpenCV* untuk mengenali dan mengklasifikasikan pola. Fitur persegi, fitur tiga persegi, dan fitur empat persegi digunakan untuk mengklasifikasikan citra berdasarkan nilai fitur tunggal. Pengujian dilakukan pada faktor-faktor seperti cahaya, jarak, kemiringan, dan beberapa wajah pada satu citra. Hasil pengujian menunjukkan bahwa sistem dapat mendeteksi wajah manusia secara akurat 100% dalam waktu kurang dari 0,5 detik dalam kondisi garis lurus. Sistem juga dapat mendeteksi wajah manusia dengan gradien $\pm 70\%$ dan dapat mendeteksi beberapa wajah dalam satu citra. Selain itu, sistem juga dapat mendeteksi objek yang menyerupai wajah manusia seperti boneka atau topeng jika objek tersebut memiliki garis luar yang mirip dengan wajah manusia [9].

G. A. Anarki, K. Auliasari, dan M. Orisa pada tahun 2021 melakukan penelitian yang berjudul “Penerapan Metode *Haar Cascade* Pada Aplikasi Deteksi Masker”. Penelitian ini menggunakan metode *Haar Cascade*. Metode ini didasarkan pada *cascade classifier*, sebuah metode deteksi objek yang dikembangkan oleh Paul Viola dan Michael Jones dalam makalah "*Fast Object Detection Using Simple Improvements*" pada tahun 2001. *Haar Cascade* terdiri dari kumpulan fitur *Haar-Like* yang dikombinasikan untuk membentuk sebuah *classifier* (klasifikasi). Fitur ini bekerja dengan cara menghitung jumlah piksel putih dikurangi jumlah piksel hitam pada area persegi panjang yang terdefinisi. Pengklasifikasi *Haar Cascade* didasarkan pada pola piksel hitam dan putih yang membentuk sebuah jaring. Dalam penelitian ini, dilakukan beberapa tes terhadap metode *Haar Cascade*, seperti pada kondisi pencahayaan, jarak dan sudut, data gambar, aplikasi, dan lain-lain. Hasil pengujian menunjukkan bahwa metode *Haar Cascade* dapat diaplikasikan pada deteksi objek dengan melakukan *training* data untuk pertama kali. Intensitas cahaya dan jarak objek memainkan peran penting dalam mendeteksi objek, di mana cahaya yang cukup dan jarak yang tepat akan memberikan hasil deteksi yang optimal [11].

H. Suraya, Ibnu, Zaid, dan Suroso pada tahun 2018 melakukan penelitian yang berjudul “Rancang Bangun Alat Pendeteksi Kantuk Pada Mobil Berbasis IoT Menggunakan *Raspberry Pi* dan Kamera” dengan menggunakan

metode *Haar Cascade*. Penelitian ini berhasil menciptakan sebuah alat pendeteksi tidur berbasis IoT dengan menggunakan *Raspberry Pi* dan kamera, serta metode *Haar Cascade Classifier*. Metode ini adalah *classifier multi-level* yang dapat mengidentifikasi fitur wajah dengan akurat. Dalam penelitian ini, dilakukan pengujian untuk menentukan akurasi aplikasi dalam kondisi jarak dan pencahayaan yang berbeda. Hasil pengujian menunjukkan bahwa pada jarak sekitar 30-55cm, program dapat mengidentifikasi mata dengan akurasi sebesar 80% [7]. Dengan tingkat akurasi tersebut, program pendeteksi kantuk tersebut secara signifikan dapat mengurangi tingginya angka kecelakaan lalu lintas. Perbandingan penelitian ini dengan penelitian sebelumnya dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan dengan penelitian sebelumnya

| No. | Judul dan Tahun | Penulis | Kelebihan | Kekurangan |
|-----|---|-------------------------------|--|---|
| 1. | Hubungan Pola Tidur Terhadap Konsentrasi Belajar Peserta Didik di SDN Kembangan Utara (2021) | N. A. Zulfa dan M. Mujazi | Dari hasil studi tersebut, dapat disimpulkan bahwa masalah tidur atau pola tidur yang tidak baik dapat mengganggu kemampuan seseorang untuk fokus dalam pekerjaan atau belajar, dan juga dapat menyebabkan perasaan sedih dan lelah. | Hanya menjelaskan hubungan antara pola tidur terhadap konsentrasi dalam proses pembelajaran di dalam kelas. |
| 2. | Kelalaian Manusia (Human Error) Dalam Kecelakaan Lalu Lintas : Analisis Berdasarkan Pemrosesan Informasi (2018) | G. T. Tjahjono dan D. Maulina | Hasil penelitian ini mengidentifikasi beberapa faktor penyebab tingginya angka kematian akibat kecelakaan lalu lintas di Indonesia. Faktor-faktor tersebut meliputi faktor fisik jalan, kendaraan, dan manusia. | Hanya menjelaskan beberapa faktor penyebab kecelakaan lalu lintas berdasarkan hasil wawancara kepada pengemudi atau pengendara. |

| No. | Judul dan Tahun | Penulis | Kelebihan | Kekurangan |
|-----|--|---|--|---|
| 3. | Pengenalan Pola Bentuk Wajah dengan OpenCV (2019) | T. A.-S. Zulkhaidi, E. Maria dan Yulianto | Penelitian ini menggabungkan pengenalan pola wajah dengan modul OpenCV. Selain itu, sistem juga dapat mendeteksi objek yang menyerupai wajah manusia seperti boneka atau topeng jika objek tersebut memiliki garis luar yang mirip dengan wajah manusia. | Sistem hanya dapat mendeteksi objek yang menyerupai wajah manusia seperti topeng dan boneka. |
| 4. | Penerapan Metode Haar Cascade Pada Aplikasi Deteksi Masker (2021) | G. A. Anarki, K. Auliasari, dan M. Orisa | Penelitian ini menggunakan metode <i>Haar Cascade</i> . Metode ini didasarkan pada <i>cascade classifier</i> . | Sistem ini hanya dapat mendeteksi masker dengan menggunakan Metode <i>Haar Cascade</i> . Intensitas cahaya dan jarak objek memainkan peran penting dalam mendeteksi objek, di mana cahaya yang cukup dan jarak yang tepat akan memberikan hasil deteksi yang optimal. |
| 5. | Rancang Bangun Alat Pendeteksi Kantuk Pada Mobil Berbasis IoT Menggunakan Raspberry Pi dan Kamera (2018) | H. Suraya, Ibnu, Zaid, dan Suroso | Penelitian ini berhasil menciptakan sebuah alat pendeteksi tidur berbasis IoT dengan menggunakan <i>Raspberry Pi</i> dan kamera, serta metode <i>Haar Cascade Classifier</i> . | Sistem dapat dijalankan namun belum menggunakan peringatan suara dalam mendeteksi kantuk. |

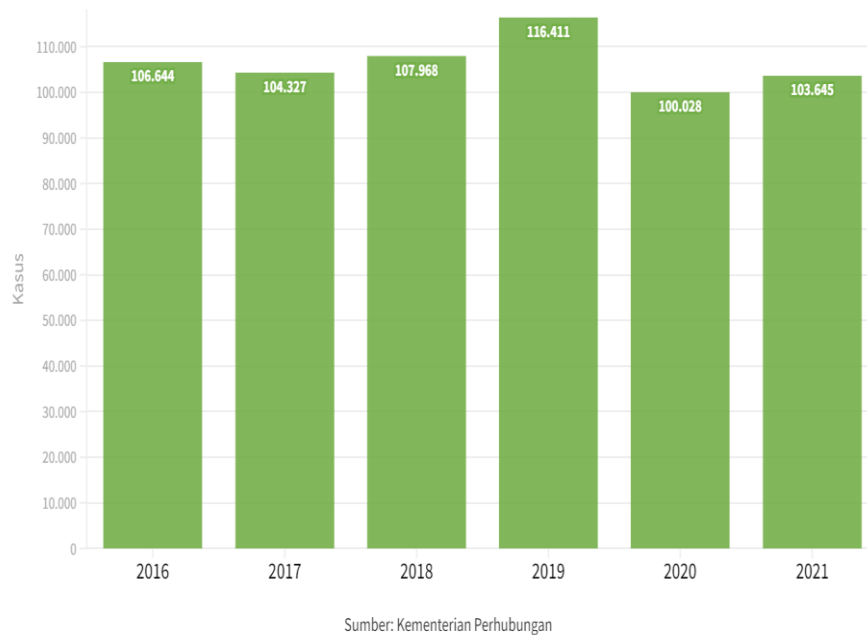
2.2 DASAR TEORI

2.2.1 Kasus Kecelakaan Lalu Lintas di Indonesia

Kecelakaan lalu lintas adalah suatu isu yang harus ditangani dengan serius karena dampak yang signifikan yang ditimbulkannya. Kecelakaan lalu lintas terjadi secara tiba-tiba dan tidak disengaja di jalan raya, melibatkan kendaraan baik ketika ada atau tidaknya kehadiran pengguna jalan lainnya, yang mengakibatkan korban manusia atau kerugian materi [21]. Kecelakaan lalu lintas merupakan salah satu jenis kecelakaan yang sering terjadi di Indonesia. Menurut data yang diterbitkan oleh Kementerian Perhubungan berdasarkan laporan Korlantas Polri, terdapat peningkatan jumlah kecelakaan lalu lintas di Indonesia dari 100.028 kasus pada tahun 2020 menjadi 103.645 kasus pada tahun 2021. Faktor manusia dianggap sebagai salah satu penyebab utama terjadinya kecelakaan, seperti mengantuk, tidur, kurang konsentrasi, dan lain sebagainya [3].

Faktor utama penyebab kecelakaan lalu lintas yaitu faktor manusia (*human factor*) dengan persentase sebesar 66,89%. Dalam faktor manusia, kurangnya antisipasi terhadap kondisi jalan dan lingkungan menjadi penyebab utama kecelakaan sebanyak 1.688 kasus, diikuti oleh keadaan mengantuk sebanyak 1.568 kasus [21]. Berdasarkan data Kementerian Perhubungan, tercatat bahwa jumlah kecelakaan lalu lintas darat di Indonesia pada tahun 2021 mencapai 103.645 kasus. Angka ini mengalami peningkatan sebesar 3,62% dibandingkan dengan tahun sebelumnya yang hanya 100.028 kasus. Pada tahun 2020, terjadi penurunan arus lalu lintas kendaraan akibat adanya pembatasan sosial yang dilakukan sebagai upaya untuk mengendalikan penyebaran pandemi Covid-19 [25]. Grafik jumlah kasus kecelakaan lalu lintas darat di Indonesia dari tahun 2016 hingga 2021 berdasarkan data dari Kementerian Perhubungan dapat dilihat pada Gambar 2.1.

**Jumlah Kasus Kecelakaan Lalu Lintas Darat di Indonesia
(2016-2021)**



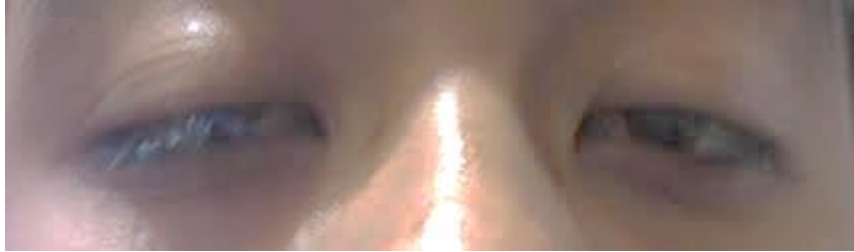
Gambar 2.1 Data kecelakaan lalu lintas darat di Indonesia

2.2.2 Mata Mengantuk

Rasa kantuk merupakan salah satu kondisi yang pastinya dialami semua orang. Beberapa faktor penyebab rasa kantuk adalah kurang beristirahat, kelelahan, dan pengaruh obat-obatan [2]. Rasa kantuk dapat merugikan dan berbahaya jika seseorang sedang mengemudi [8]. Terdapat istilah mengantuk atau tertidur selama beberapa detik yang disebut *microsleep*. Pengemudi yang mengalami *microsleep* disebabkan kelelahan. Durasi dari *microsleep* ini sangat singkat, biasanya antara 3 hingga 5 detik, meskipun dalam beberapa kasus bisa mencapai 10 detik [23]. Tanda-tanda mengantuk pada seseorang dapat terlihat dari gejala seperti rasa berat pada kelopak mata, penglihatan yang kabur, dan perasaan ketidakseimbangan pada kepala akibat menahan beban. Oleh karena itu, orang yang mengalami kondisi ini cenderung merasa perlu untuk berbaring dan istirahat [13].

Tanpa kita sadari, *microsleep* sering terjadi saat kita sedang melakukan aktivitas, terutama saat mengemudi. Fenomena ini menjadi salah satu faktor utama penyebab kecelakaan yang disebabkan oleh kesalahan manusia, seperti rasa mengantuk, ketidakfokusan, dan kelelahan. Faktor-faktor ini berkontribusi sebesar 61% dari total kecelakaan, sementara faktor kendaraan menyumbang

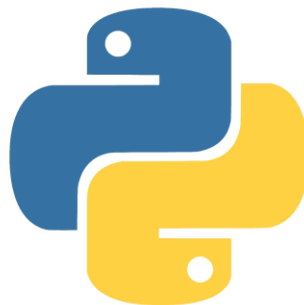
sebesar 9%, dan faktor prasarana serta lingkungan menyumbang sebesar 30% [26]. Contoh dari mata mengantuk dapat dilihat pada Gambar 2.2.



Gambar 2.2 Mata mengantuk

2.2.3 Bahasa Pemrograman *Python*

Python adalah bahasa pemrograman tingkat tinggi yang diciptakan dan dikembangkan oleh Guido van Rossum di Belanda tahun 1990. Bahasa pemrograman ini sudah banyak digunakan untuk mengembangkan berbagai aplikasi *desktop*, *web*, dan lain-lain [10]. Bahasa pemrograman ini bersifat *interpreter*, interaktif, dan *object-oriented* (berorientasi objek) dan dapat digunakan hampir di semua *platform*. Bahasa pemrograman *Python* mudah untuk dipelajari karena kesederhanaannya, memiliki sintaks yang jelas, dan punya pustaka yang luas [9]. Bahasa pemrograman *Python* semakin populer dan semakin banyak dipelajari oleh mahasiswa, terutama di kampus-kampus yang memiliki fokus pada bidang teknologi informasi. Mahasiswa menggunakan *Python* untuk menyelesaikan berbagai tugas kuliah, tugas akhir, dan tugas penelitian [11]. Logo bahasa pemrograman *Python* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Bahasa pemrograman *Python*

2.2.4 Deep Learning

Deep learning merupakan salah satu cabang di bidang *machine learning* yang memanfaatkan jaringan saraf tiruan untuk memecahkan masalah yang melibatkan dataset yang besar. Teknik *deep learning* memberikan keunggulan dalam pembelajaran terawasi (*supervised learning*), di mana model dapat belajar secara mandiri dengan representasi yang lebih kompleks. Dengan menambahkan lebih banyak lapisan ke dalam jaringan saraf, model pembelajaran dapat mampu menggambarkan data citra yang berlabel dengan lebih akurat dan mendalam. Prinsip ini terus berkembang sehingga *deep learning* semakin populer di kalangan komunitas riset dan industri sebagai solusi untuk mengatasi berbagai tantangan dalam pemrosesan data besar, seperti *Computer Vision*, *Speech Recognition*, dan *Natural Language Processing*. *Deep learning* merupakan salah satu teknik yang termasuk dalam bidang *machine learning* dan memiliki struktur arsitektur yang lebih kompleks dibandingkan dengan teknik *machine learning* lainnya, sehingga mampu mengatasi masalah prediksi dan klasifikasi dengan lebih baik [28].

2.2.5 Computer Vision

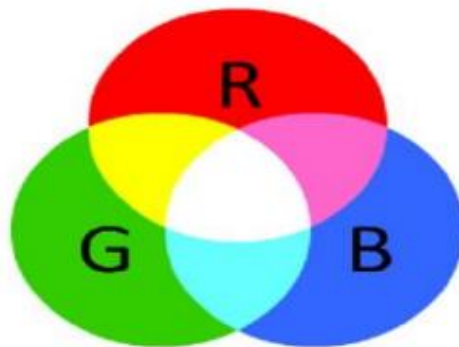
Pengolahan citra adalah suatu proses pengolahan citra yang melibatkan penggunaan komputer untuk memproses citra dan menghasilkan citra baru. Pengolahan citra memiliki hubungan dengan *computer vision* di mana pengolahan citra berfungsi sebagai tahap awal (*preprocessing*) dalam pengolahan *computer vision*. *Computer vision* berupaya untuk meniru mekanisme visual manusia (*human vision*). Pada dasarnya *human vision* sangatlah kompleks, di mana manusia melihat objek melalui indera penglihatan (mata) dan informasi citra objek tersebut diteruskan ke otak untuk diinterpretasikan, sehingga manusia dapat memahami apa yang terlihat dalam pandangan mata mereka. *Computer vision* adalah suatu proses otomatis yang menggabungkan berbagai tahapan untuk melakukan persepsi visual, termasuk akuisisi citra, pengolahan citra, klasifikasi, pengenalan, dan pengambilan keputusan [29].

2.2.6 Citra dan RGB

Saat ini, perkembangan teknik di bidang *image processing* sudah sangat berkembang sehingga menjadi daya tarik manusia untuk mempelajarinya. Citra merupakan representasi objek dua dimensi untuk objek tiga dimensi, yang juga

merupakan kumpulan piksel-piksel atau titik-titik berwarna dua dimensi [15]. Pengolahan citra digital adalah suatu bidang ilmu yang fokus pada metode dan teknik untuk membentuk, mengolah, dan menganalisis citra dengan tujuan menghasilkan informasi yang dapat dipahami manusia dan komputer. Citra tersebut dapat berupa foto atau gambar bergerak (video).

RGB (*Red-Green-Blue*) adalah tiga warna dasar (*primary colors*) yang secara umum dijadikan acuan warna [16]. Seperti yang kita ketahui, RGB atau *Red, Green, Blue* adalah sistem pengkodean warna yang digunakan dalam tampilan digital. Sistem ini banyak digunakan pada monitor komputer, video, layar ponsel, dan berbagai perangkat lainnya. Pengolahan citra digital adalah proses manipulasi dan interpretasi citra menggunakan komputer. Pada pengolahan citra, citra digunakan sebagai *input*, dan *output*-nya adalah citra yang telah mengalami proses pengolahan. Secara umum, pengolahan citra digital dapat didefinisikan sebagai proses pengolahan citra dua dimensi menggunakan komputer [18]. Kombinasi warna RGB dapat dilihat pada Gambar 2.4.



Gambar 2.4 Warna RGB [18]

2.2.7 Pengolahan Citra (*Image Processing*)

Dalam hal pembentukan sinyal, citra dapat diklasifikasikan menjadi dua jenis, yaitu citra analog dan citra digital. Citra analog terbentuk melalui sinyal analog yang memiliki karakteristik kontinu, sedangkan citra digital terbentuk melalui sinyal digital yang memiliki karakteristik diskrit. Contoh dari citra analog meliputi gambar yang dapat dilihat oleh mata manusia, serta foto dan film yang diambil menggunakan kamera analog. Pengolahan citra merupakan proses untuk meningkatkan kualitas citra menggunakan komputer. Pengolahan citra merupakan salah satu teknologi yang digunakan untuk menyelesaikan masalah dalam

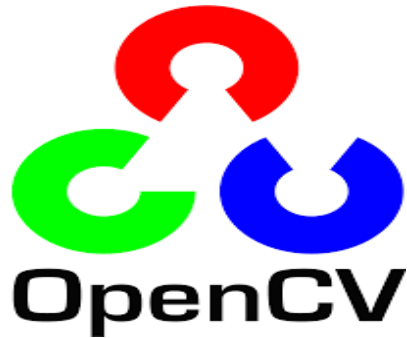
pemrosesan gambar, sedangkan *computer vision* bertujuan untuk membuat keputusan tentang objek fisik nyata berdasarkan data yang diperoleh dari perangkat atau sensor. Pengolahan citra melibatkan berbagai operasi seperti perbaikan, analisis, dan transformasi citra [24]. Pengolahan citra bertujuan untuk memfasilitasi interpretasi manusia maupun mesin (komputer) terhadap citra yang mengalami gangguan, sehingga menjadi lebih mudah dipahami. Ada beberapa contoh operasi pengolahan citra yang dapat dilakukan, seperti mengubah kontras citra, menghapus derau (*noise*) menggunakan operasi penapisan (*filtering*), meningkatkan ketajaman (*sharpening*), memberikan warna semu (*pseudocoloring*), dan sejenisnya [29].

2.2.8 Face Detection

Wajah adalah salah satu bagian unik dari tubuh manusia yang memiliki karakteristik yang berbeda-beda pada setiap individu. Bagian-bagian wajah yaitu mata, hidung, pipi, mulut, dahi, alis, dagu, dan rambut. *Face detection* adalah suatu teknologi yang digunakan untuk mengenali keberadaan wajah dalam format gambar digital. Teknologi ini menggunakan algoritma khusus yang dirancang untuk mendeteksi wajah manusia pada gambar digital. Biasanya, algoritma pendeteksian wajah dimulai dengan fokus pada mata manusia karena mata merupakan salah satu fitur yang paling mudah dideteksi [20]. Deteksi wajah adalah sebuah metode yang digunakan untuk mengidentifikasi fitur wajah dengan tujuan pengenalan atau pendeteksian wajah. Pengenalan wajah merupakan salah satu teknologi dalam bidang pengolahan citra (*computer vision*) yang digunakan untuk mengenali identitas seseorang atau mendapatkan informasi dari sebuah wajah [22].

2.2.9 OpenCV Library

OpenCV (Open Source Computer Vision Library) adalah salah satu pustaka (*library*) dari *computer vision* yang berbasis *open source project* dibuat oleh Intel untuk pengolahan citra, baik dalam bentuk gambar atau video. *OpenCV* memiliki lebih dari 2.500 algoritma dimana algoritma tersebut dapat digunakan untuk pendeteksian, pengenalan wajah manusia atau hewan, pelacakan pergerakan kamera, dan lain sebagainya [11]. Logo *OpenCV* dapat dilihat pada Gambar 2.5.

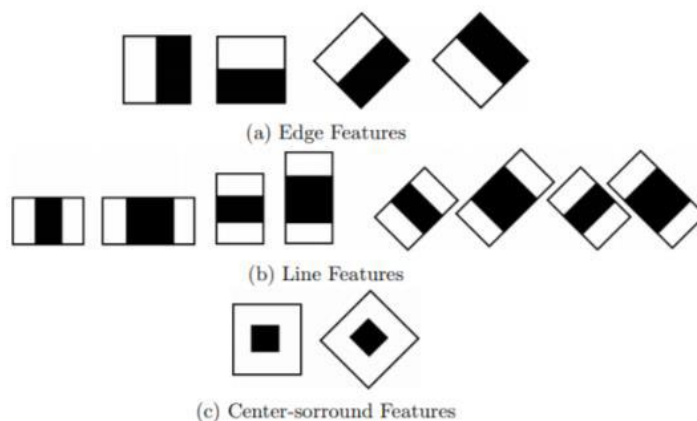


Gambar 2.5 Logo *OpenCV* [2]

OpenCV (*Open Computer Vision*) merupakan *library open source* yang dirancang khusus untuk memproses citra dengan tujuan memberikan kemampuan pengolahan visual komputer yang menyerupai kemampuan visual manusia. Selain itu *OpenCV* juga menyediakan modul-modul yang digunakan untuk mendeteksi objek menggunakan metode *computer vision* [9].

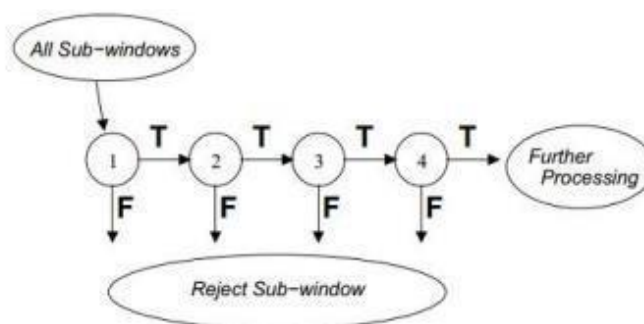
2.2.10 *Haar Cascade Classifier*

Haar Cascade Classifier atau *Haar like feature* merupakan fitur persegi (*rectangular*) yang memberikan referensi akurat pada sebuah gambar. Konsep *Haar Cascade* berasal dari ide Paul Viola dan Michael Jhon, sehingga sering disebut metode Viola dan Jhon. Salah satu keunggulan metode ini terletak pada efisiensi komputasinya yang dapat mengandalkan jumlah piksel dalam suatu persegi, bukan setiap nilai piksel dalam gambar secara individu. Metode ini masuk dalam kategori metode yang menggunakan model statistik (*classifier*) untuk melakukan analisis [12]. *Haar-like feature* atau *Haar Cascade Classifier* adalah sebuah fungsi persegi yang memberikan indikasi pada sebuah gambar. Fitur *Haar-like* ini digunakan untuk menganalisis gambar dengan memproses kotak-kotak yang terdapat di dalamnya [11]. Pengklasifikasi *Haar Cascade* berasal dari gabungan piksel hitam dan piksel putih yang membentuk kotak. Fitur *Haar-like* menangani gambar dalam kotak, di mana ada beberapa piksel dalam satu bingkai. Kemudian proses setiap kotak dan hasilkan nilai yang berbeda untuk menunjukkan area gelap dan terang. Nilai-nilai ini akan digunakan sebagai dasar untuk pengolahan citra [8]. Penjelasan *Haar-like features* dapat dilihat pada Gambar 2.6.



Gambar 2.6 Haar-Like Features [11]

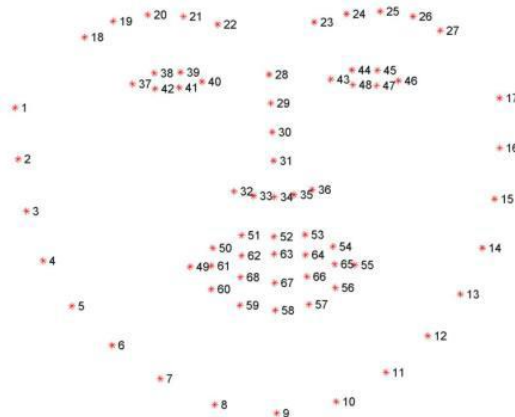
Cascade classifier adalah suatu rangkaian tahapan *classifier* yang digunakan untuk mengidentifikasi keberadaan objek yang ingin dideteksi ke dalam *sub-window* gambar [11]. Pengklasifikasi *Cascade* adalah langkah penghitungan berulang kali nilai Fitur Haar untuk mendapatkan hasil yang lebih akurat. Gambar 2.7 menunjukkan alur kerja dari pengklasifikasi *Cascade*. Pada tahap klasifikasi 1, setiap sub-citra akan diklasifikasikan dengan ciri, jika hasilnya tidak memenuhi standar maka hasilnya akan ditolak. Pada klasifikasi tahap 2, setiap sub-citra akan direklasifikasi. Jika ambang batas yang diperlukan diperoleh, tahap filter berikutnya (tahap klasifikasi 3) dimasukkan. Sampai sub-gambar yang lewat dikurangi menjadi mendekati gambar dalam sampel yang telah diuji [8]. Algoritma *Cascade Classifier* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Cascade Classifier [11]

2.2.11 Facial Landmark

Facial landmark adalah metode yang digunakan untuk menentukan posisi dan lokalisasi daerah wajah seperti mata, alis, hidung, mulut, garis rahang. Proses identifikasi dapat dilakukan dengan cara menetapkan parameter pada bagian yang akan dideteksi [13]. Proses ekstraksi titik-titik wajah menggunakan metode *Dlib Regression Tree*. Metode ini digunakan untuk mengidentifikasi dan menentukan posisi 68 titik yang telah dilatih pada wajah. Dengan menggunakan metode *facial landmark*, kita dapat mendeteksi area wajah dengan menempatkan titik-titik khusus pada wajah sehingga sistem dapat mengenali wajah tersebut. *Facial landmark* digunakan untuk berbagai tujuan seperti pengenalan wajah (*face recognition*), rekonstruksi wajah (*face reconstruction*), dan analisis atribut wajah (*facial attributes*) seperti usia, jenis kelamin, senyuman, dan penggunaan kacamata [19]. Koordinat pada wajah (*facial landmark*) dapat dilihat pada Gambar 2.8.

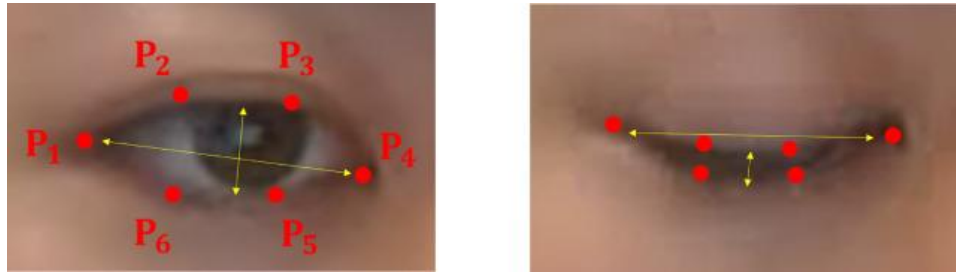


Gambar 2.8 Koordinat wajah *facial landmark* [13]

2.2.12 Eye Aspect Ratio (EAR)

Eye Aspect Ratio adalah salah satu library *DLIB Python* yang digunakan untuk menghitung dan menentukan nilai ambang batas pada mata. Perhitungan EAR dilakukan menggunakan koordinat piksel pada titik-titik *facial landmark* wajah yang berhubungan dengan bagian mata [13]. Pengembangan dari deteksi wajah adalah deteksi mata (*eyes detection*). Deteksi mata merupakan tahap lanjutan dari deteksi wajah, di mana citra wajah manusia yang berhasil dideteksi akan diproses lebih lanjut untuk mengidentifikasi dan mendeteksi kedua mata yang terdapat pada wajah tersebut [20]. Algoritma *Eye Aspect Ratio* (EAR) dapat

dilihat pada Gambar 2.9.



Gambar 2.9 Eye Aspect Ratio

Perhitungan *Eye Aspect Ratio* (EAR) ditunjukkan pada persamaan 2.1. [13]

Rumus :

$$EAR = \frac{||p2 - p6|| + ||p3 - p5||}{2||p1 - p4||} \quad (2.1)$$

Keterangan rumus [13]:

- P1 sampai P6 mewakili posisi *landmark* dua dimensi (2D) pada retina mata.
- P2, P3, P5, dan P6 mewakili untuk mengukur tinggi retina mata.
- P1 dan P4 mewakili untuk mengukur lebar retina mata.

2.2.13 PyCharm Community Edition

PyCharm Community Edition adalah sebuah platform IDE bahasa pemrograman *Python*. IDE ini memiliki kelebihan dari segi persaingan karena produktivitas tools yang tersedia seperti *quick fixes* (perbaikan cepat). Platform ini terdiri dari tiga versi, yaitu versi *Community* berlisensi *Apache*, versi *Education*, dan versi *Professional* berbayar. Versi *Community* pada platform ini sangat menarik bagi para pengembang karena memiliki fitur seperti penyorotan sintaks, *auto-completion*, dan verifikasi kode secara langsung. Logo dari platform *PyCharm Community* dapat dilihat pada Gambar 2.10. Berikut beberapa kelebihan dan kekurangan versi *Community* [27]:

- Kelebihan
 - Dukungan komunitas yang aktif
 - Memiliki fitur penyorotan sintaks dan *auto-completion*
 - Dapat menjalankan pengeditan dan debug *Python* tanpa

persyaratan eksternal

- Kekurangan
 - Waktu pemuatan lebih lambat
 - Pengaturan default yang memerlukan penyesuaian sebelum proyek dapat digunakan.



Gambar 2.10 Logo *Pycharm* [27]

2.2.14 *Confusion Matrix*

Confusion matrix adalah metode pengukuran untuk mengevaluasi kinerja dan tingkat keakuratan suatu proses klasifikasi. Berisi tabel yang menyajikan informasi jumlah data yang diklasifikasikan secara benar dan salah [17]. Prinsip utama dari *confusion matrix* adalah memberikan informasi perbandingan hasil klasifikasi dari sistem dengan hasil klasifikasi yang seharusnya. Dalam mengukur performa sistem menggunakan *confusion matrix*, terdapat empat istilah yang digunakan untuk mewakili hasil proses klasifikasi [24]. Tabel *Confusion Matrix* dapat dilihat pada Tabel 2.2.

Tabel 2.2 *Confusion Matrix* [17]

| | | AKTUAL | |
|----------|----------|----------------------------|----------------------------|
| | | Positive | Negative |
| PREDIKSI | Positive | <i>True Positive (TP)</i> | <i>False Positive (FP)</i> |
| | Negative | <i>False Negative (FN)</i> | <i>True Negative (TN)</i> |

Penjelasan singkat dari sejumlah keterangan yang ada pada tabel 2.2 meliputi :

1. *True Positive* (TP)

True Positive (TP) berisi jumlah data pada kelas aktual positif yang diprediksi sistem sebagai kelas positif [17].

2. *True Negative* (TN)

True Negative (TN) berisi jumlah data pada kelas aktual negatif yang diprediksi sistem sebagai kelas negatif [17].

3. *False Positive* (FP)

False Positive (FP) berisi jumlah data pada kelas aktual negatif yang diprediksi sistem sebagai kelas positif [17].

4. *False Negative* (FN)

False Negative (FN) berisi jumlah data pada kelas aktual positif yang diprediksi sistem sebagai kelas negatif [17].