

BAB II

DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian pertama oleh Moh. Wahyu Septyanto, dkk. (2019) berjudul "Aplikasi Presensi Pengenalan Wajah dengan Algoritma *Haar Cascade Classifier*" membahas presensi wajah bagi pegawai. Penelitian menggunakan metode *Haar Cascade* karena komputasi cepat dan kemampuan mengidentifikasi wajah pada jarak jauh. Pengujian melibatkan 13 karyawan, setiap karyawan melakukan 30 kali presensi. Hasil menunjukkan sistem merancang tingkat akurasi yang baik, mencapai 87% keberhasilan dan 13% kegagalan dari 390 percobaan. Kegagalan disebabkan oleh faktor perubahan seperti pencahayaan, posisi kepala, dan atribut seperti topi dan kacamata [6].

Peneliti kedua, Banu Santoso, dkk. (2020), dalam penelitian "Implementasi Penggunaan *OpenCV* pada *Face Recognition* untuk Sistem Presensi Perkuliahan Mahasiswa," membahas presensi mahasiswa melalui pemindaian wajah. Studi ini mengadopsi metode *Face Recognition* berbasis *OpenCV* dengan *Haar Cascade Classifier* dan *Local Binary Patterns Histograms (LBPH)*. Sistem presensi ini dapat mendeteksi wajah terdaftar dalam sistem, optimal pada jarak 150 cm. Kendati demikian, keberhasilan *Face Recognition* terganggu oleh hambatan wajah atau jarak melebihi 150 cm. Hasil penelitian menunjukkan *Face Recognition* mampu mendeteksi beberapa objek wajah dalam satu bingkai yang terdaftar dalam sistem presensi, selama jarak optimal 150 cm terjaga [7].

Peneliti ketiga, Awang Hendrianto Pratomo, dkk (2019), dalam penelitian "Pengenalan Wajah untuk Pemantauan Kehadiran Pegawai Menggunakan Metode *Viola Jones* dan *Euclidean Distance*," memfokuskan pada sistem presensi pegawai dengan metode *Viola Jones*. Penelitian ini menerapkan metode *Viola Jones* dan *Principal Component Analysis (PCA)* untuk mengubah gambar menjadi nilai citra, yang kemudian dibandingkan menggunakan *Euclidean Distance* untuk menemukan kemiripan terdekat. Metodologi *waterfall* digunakan dalam penelitian ini. Hasil penelitian menunjukkan efisiensi waktu dalam penggunaan presensi serta peningkatan keamanan melalui pemantauan yang lebih baik terhadap orang yang

masuk dan keluar ruangan. *Metode Viola Jones* dan *Euclidean Distance* diimplementasikan dengan akurasi tertinggi mencapai 100%, memperkuat pendeteksian dan pengenalan wajah dalam sistem presensi [8].

Peneliti keempat, Rastri Prathivi, dkk (2020), dalam penelitian berjudul "Sistem Presensi Kelas Menggunakan Pengenalan Wajah dengan Metode *Haar Cascade Classifier*," menerapkan metode *Haar Cascade Classifier* dalam sistem presensi. Sistem ini terdiri dari tiga fase: deteksi wajah dan pengumpulan data, latihan pengenalan wajah, serta pengenalan wajah. Meskipun demikian, sistem ini memiliki keterbatasan dalam mengenali wajah siswa dalam kondisi pencahayaan redup dan hanya mampu mendeteksi wajah pada jarak sekitar 1 meter. Hasil penelitian ini mengungkapkan bahwa keseluruhan sistem dapat beroperasi dengan baik dalam mendeteksi dan mencatat kehadiran objek dengan akurasi keberhasilan sistem mendeteksi wajah sebesar 75%. Meskipun terdapat beberapa batasan, penelitian ini memberikan kontribusi dalam pengembangan sistem presensi dengan metode pengenalan wajah yang berpotensi untuk digunakan dalam lingkungan kelas [9].

Peneliti kelima, Miftah Khul Janah, dkk. (2021), mengungkap judul "Sistem Pencatatan Kehadiran Deteksi Wajah Menggunakan Metode *Haar Feature Cascade Classifier*." Penelitian ini melibatkan pendaftaran 450 wajah ke dalam sistem, yang kemudian disimpan dalam *dataset*. Setiap wajah diuji pengenalan sebanyak 30 kali dan diubah ke format *Grayscale*. Pengujian sistem dilakukan pada 10 mahasiswa yang terdaftar dalam *dataset*, dengan jarak paling mendekati 50 cm. Hasil pengujian menunjukkan bahwa jumlah wajah yang benar dikenali (identifikasi yang tepat) adalah 7 orang, sedangkan 2 orang diidentifikasi dengan ke tidak tepatan. Pencahayaan dan jarak terbukti mempengaruhi hasil pengenalan wajah, ditunjukkan melalui perbandingan data eksperimental 10 orang dengan variasi intensitas cahaya dan jarak yang berbeda. Penelitian ini menghasilkan akurasi pengenalan wajah dalam presensi mahasiswa sebesar 70% menggunakan metode *Haar Cascade Classifier*. Dengan persentase ini, sistem pengenalan wajah terbukti berguna untuk pencatatan kehadiran mahasiswa [10].

Tabel 2.1 Deskripsi singkat penelitian

Nama Peneliti/ tahun	Metode	Lokasi Penelitian	Kontribusi Penelitian yang akan di kerjakan	Parameter
Moh. Wahyu Septyanto, dkk./2019[6].	<i>Haar Cascade</i> Classifier dan <i>Adaboost</i> <i>learning</i>	Instansi	Merancang sistem absensi wajah untuk pegawai	- Jarak -Posisi wajah -Aksesoris -Intensitas cahaya
Banu Santoso, dkk./2020[7].	<i>Haar Cascade</i> dan algoritma LBPH	Universitas	Mengurangi kecurangan presensi kehadiran mahasiswa	-Jarak - <i>Singel</i> / <i>Multiple</i> -Penghalang
Awang Hendrianto Pratomo, dkk./2019[8].	<i>Viola Jones</i> dan <i>Eudclidean</i> <i>Distace</i> <i>Method</i>	Instansi	Memantau kinerja pegawai instansi dan sebagai sistem keamanan instansi	-Jarak -Posisi wajah -Intensitas Cahaya
Rastri Prathivi, dkk./2020[9].	<i>Haar Cascade</i> <i>Classifier</i>	Sekolah	Merancang sistem presensi untuk memudahkan presensi	-Jarak -Intensitas cahaya
Miftah Khul Janah, dkk./2021 [10].	<i>Haar Featur</i> <i>Cascade</i> <i>Classifier</i> LBP	Universitas	Mengurangi kecurangan presensi mahasiswa	-Jarak -Intensitas Cahaya

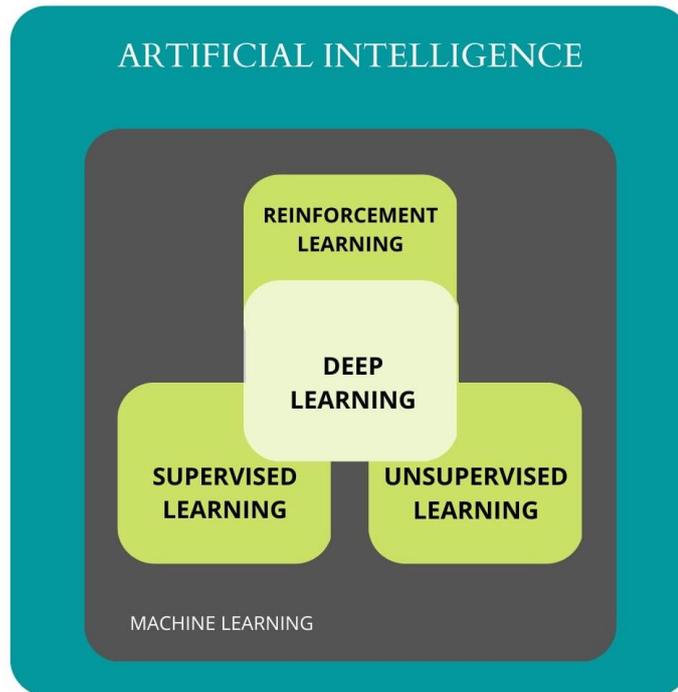
Nama Peneliti/ tahun	Metode	Lokasi Penelitian	Kontribusi Penelitian yang akan di kerjakan	Parameter
Penelitian ini	<i>Haar Cadcade, Algoritma LBPH dan JST</i>	Universitas	Merancang sistem presensi dan Mengurangi kecurangan dalam presensi mahasiswa	-Jarak -Posisi Wajah -Aksesoris -Intensitas cahaya

Penelitian ini menggunakan sistem presensi wajah dengan metode *Haar Cascade* dengan algoritma LBPH dan metode JST untuk merancang sistem presensi wajah, sistem ini dapat melakukan identifikasi wajah menggunakan pola unik dari wajah yang ada dalam *dataset* sehingga sistem yang akan di rancang dapat mengurangi kecurangan dalam melakukan presensi pada saat perkuliahan secara luring baik dalam kelas teori maupun pada kelas praktikum. Hasil dari penelitian ini Sistem presensi mampu mengenali wajah mahasiswa dengan rentang jarak 20cm sampai 40cm, dengan rata-rata perolehan akurasi mencapai 93% pada kondisi terang (268 *lux*) dengan rotasi wajah dan menggunakan aksesoris. sementara untuk kondisi redup (86 *lux*) perolehan akurasi mencapai 90% dengan berbagai rotasi wajah dan penggunaan aksesoris.

2.2 DASAR TEORI

2.2.1 AI (*Artificial intelligence*)

Artificial intelligence (AI) merupakan suatu kecerdasan buatan yang sering digunakan dalam berbagai bidang, baik dalam bisnis, industri, pekerjaan kantor, dan dunia Pendidikan. *Artificial intelligence* memberikan dorongan untuk merubah dunia menjadi semakin berkembang. AI termasuk dalam bidang ilmu komputer yang dapat mensimulasikan kecerdasan yang ada pada manusia serta menerapkan pada mesin dan di program serta dapat memecahkan masalah manusia yang kognitif [11].



Gambar 2.1 Hubungan Kecerdasan buatan *machine learning*, *Deep learning* dan *reinforcement learning* [12].

Pada Gambar 2.1 Kecerdasan buatan (*Artificial intelligence*) mempunyai dua macam cabang ilmu komputer yaitu *Machine Learning* (ML) dan *Deep learning* (DL). *machine learning* merupakan sebuah cabang ilmu komputer yang dikembangkan untuk pembelajaran *statistic* untuk belajar sendiri tanpa arahan *user*, proses pengambilan keputusan *Machine learning* dilakukan berdasarkan pengalaman ada tiga proses pengambilan keputusan diantara-Nya *Memorize* mengingat semua data atau kejadian, *Generalize* berdasarkan bentuk pola dari data, dan *Predict* memprediksi suatu Tindakan [13].

Sedangkan *Deep learning* merupakan salah satu dari ilmu komputer untuk mengajarkan mesin dalam pemrosesan data dengan cara meniru sistem kerja dari otak manusia baik dalam analisis maupun klasifikasi, *Deep learning* mempunyai empat arsitektur dasar yaitu DNN (*Deep Neural Network*), ANN (*Artificial Neural Network*), CNN (*Convolutional Neural Network*), dan RNN (*Recurrent Neural Network*) Dari empat arsitektur *Deep learning* sering kali digunakan untuk menggantikan tugas manusia dalam menjalankan tugas dan sistem pembelajaran *Deep learning* dapat meningkat bersamaan dengan jumlah data yang dilatih [14].

2.2.2 Deep Learning

Deep learning pertama kali di publikasikan oleh Geoffrey Hinton pada tahun 2006, *deep learning* merupakan mesin pembelajaran yang mampu memproses data dengan meniru sistem kerja otak manusia dalam pembelajaran mesin, *Deep learning* juga mampu menganalisis dan mengklasifikasi suatu citra dengan meningkatnya data yang di latih serta mampu memberikan kinerja yang baik. *Deep learning* dapat melakukan pembelajaran model (*supervised learning*) dengan algoritma dasar dari *Deep learning* [14].

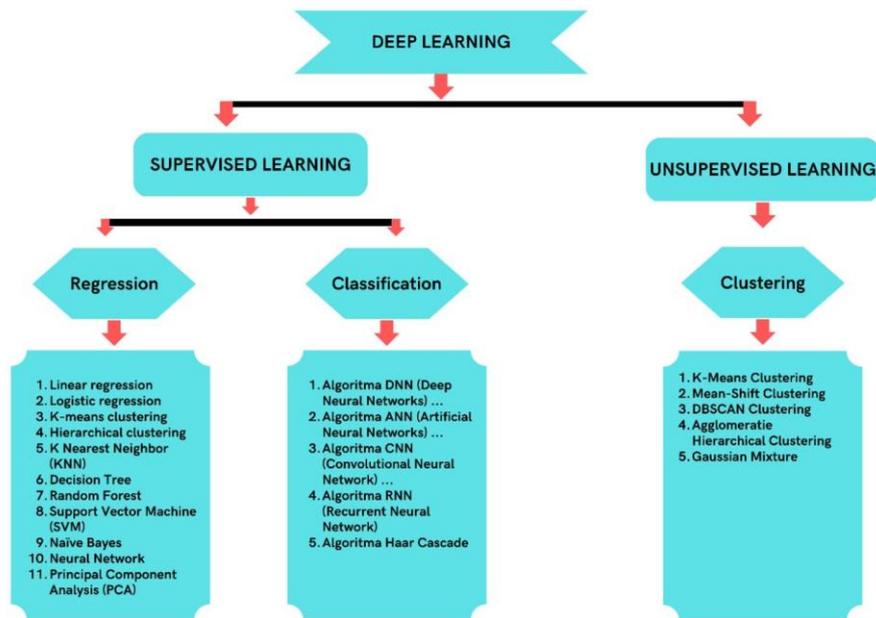
1. *Supervised learning*

Supervised learning merupakan Proses pembelajaran mesin yang terawasi, bertujuan untuk mempelajari ciri dari ilustrasi. pada pembelajaran mesin akan mempelajari data *training* yang di masukan dalam mesin pembelajaran akan dijadikan contoh informasi data, dan selanjutnya mesin akan mempelajari data tersebut dengan algoritma tertentu untuk membuat model berdasarkan data *training*. Ada beberapa algoritma yang termasuk dalam *supervised learning* diantaranya *algoritma Hebbian (Hebb Rule)*, *algoritma Perceptron*, *algoritma Adaline*, *algoritma Boltzman*, *algoritma Hapfield*, *algoritma Backpropagation*, *Haar Cascade* dan lain sebagainya [13].

Dalam pembelajaran *supervised learning* dapat dibagi menjadi dua tipe masalah yaitu *Regression* dan *Classification*. *Regression* merupakan algoritma dengan variabel *input* dan variabel *output* yang saling berhubungan algoritma ini sering di gunakan pada ramalan cuaca, prediksi variabel kontinu, dan tren pasar. Sedangkan *classification* merupakan algoritma yang variabel *input* dan *output*-nya memiliki tipe kategorial atau memiliki dua kelas seperti *true- false*, laki-laki perempuan, dan iya-tidak [15].

2. *Unsupervised learning*

Unsupervised learning adalah suatu metode pembelajaran yang menggunakan data tanpa label dan mempelajari pola dari data yang beri tanpa perintah apa yang harus di pelajari. untuk menyelesaikan dua masalah komputasi atau menarik kesimpulan dari data, *unsupervised learning* memiliki tiga metode pembelajaran diantaranya-nya Pengelompokan data (*clustering*), pengurangan dimensi data (*dimensional reduction*), dan pemberian *reward (reinforcement learning)* [16].



Gambar 2.2 Diagram *Deep learning*, *supervised learning*, dan *Unsupervised learning*[17].

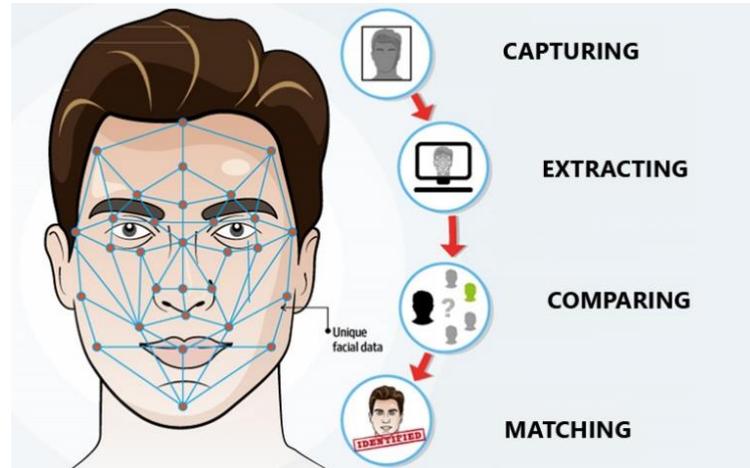
Pada gambar 2.2, dapat dijelaskan bahwa metode Haar Cascade adalah contoh dari teknik pembelajaran terawasi (*supervised learning*). Teknik ini memanfaatkan dataset untuk melatih classifier dan kemudian menerapkan classifier tersebut untuk mendeteksi objek dalam citra wajah [18].

2.2.3 Biometrik

Biometrik berasal dari bahasa Yunani "*bios*" yang berarti hidup, dan "*metron*" yang berarti mengukur. Biometrik melibatkan pengukuran dan perhitungan karakteristik fisik atau unik manusia melalui identifikasi atau otentikasi. Sistem biometrik digunakan luas dalam keamanan karena setiap individu memiliki karakteristik unik yang dapat digunakan untuk tujuan tersebut. Proses biometrik memerlukan pengumpulan data fisik, seperti wajah, suara, dan sidik jari, yang disimpan dalam sebuah *database*. Data ini kemudian dibandingkan dengan data orang lain untuk verifikasi identitas. Teknologi biometrik memiliki peran penting dalam mencegah penipuan, pencurian, dan kejahatan *online* [19].

Perkembangan teknologi dalam bidang Biometrik memiliki peranan positif dalam mencegah terjadinya penipuan, pencurian, kejahatan dalam dunia maya dan lain sebagainya. Dalam teknologi Biometrik terdapat berbagai macam jenis sistem

keamanan seperti pemindai retina, pemindai iris, pemindai *fingerprint*, pemindai wajah, dan pengenalan suara. Sistem keamanan ini sering digunakan dalam berbagai bidang baik dalam keamanan hukum, kesehatan, militer ataupun aplikasi komersial [20].



Gambar 2.3 Sistem pengenalan wajah *Biometrik*[21].

Pada Gambar 2.3 merupakan teknologi Biometrik untuk mengenali ciri unik pada wajah manusia dan memproses menjadi digital, proses ini dilakukan dalam beberapa tahapan diantara-Nya [19] :

1. Menangkap citra wajah
Sistem akan mencari lokasi wajah dalam gambar dengan menggunakan teknik deteksi objek seperti *Haar Cascade*
2. Mengekstraksi ciri wajah
Setelah gambar wajah disesuaikan, sistem akan mengekstraksi ciri-ciri penting pada wajah seperti bentuk bibir, mata, hidung, dan lain-lain.
3. Membandingkan ciri wajah
sistem akan memutuskan apakah wajah yang dikenali cocok dengan wajah yang ada di dalam *data base*.
4. Pencocokan wajah
Sistem akan membandingkan ciri-ciri wajah yang telah diekstrak dengan ciri-ciri yang ada di dalam data Base untuk mencari kemiripan.

2.2.4 Face Recognition

Wajah merupakan salah satu dalam sistem keamanan *Biometrik*, saat ini telah dikembangkan teknologi pengenalan wajah yaitu metode *face recognition*. *Face recognition* digunakan untuk menemukan fitur pada daerah wajah maka diperlukan pengenalan atau identifikasi wajah, ini merupakan salah satu dalam teknologi pengolahan citra untuk mengenali informasi dari sebuah wajah [22]. sistem ini mampu melakukan pencocokan wajah manusia dalam gambar digital dengan *data Base*. Dengan menggunakan *library OpenCV* yang digunakan untuk mengelola gambar dan video. Untuk melakukan pencocokan wajah ada beberapa faktor yang dapat mempengaruhi pengenalan wajah diantara-Nya ada bagian wajah yang tertutup seperti kacamata, penutup kepala, kulit wajah yang berbeda, ekspresi wajah, arah gambar dan lain sebagainya [20].

2.2.5 Haar Cascade

Haar Cascade merupakan metode yang sering digunakan untuk mendeteksi objek berupa gambar digital yang menampilkan fungsi matematis berupa kotak dengan tampilan nilai RGB dari setiap *pixel* lalu Viola-Jones selaku pengembang dari metode *Haar Cascade* mengembangkan algoritmanya dan menghasilkan nilai yang berupa daerah gelap dan terang yang akan menjadi dasar suatu pemrosesan gambar dikenal sebagai *Haar-Like Feature* [18]. Pada proses ekstraksi fitur haar like feature dapat di jelaskan dengan persamaan :

$$F_{\text{Haar}} = \sum F_{\text{putih}} - \sum F_{\text{hitam}} \dots \dots \dots (2.1)$$

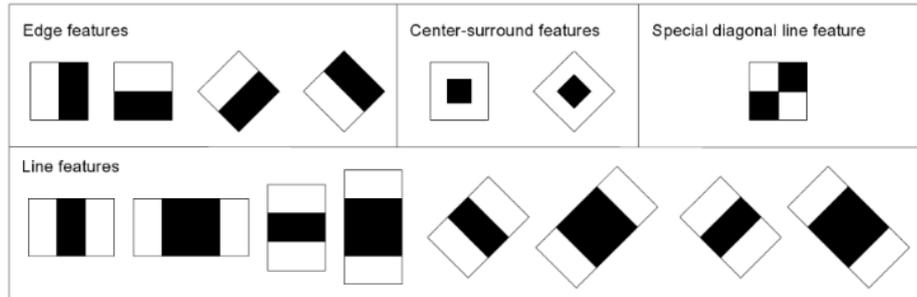
Keterangan:

F_{Haar} = Nilai fitur total

$\sum F_{\text{putih}}$ = Nilai fitur pada daerah terang

$\sum F_{\text{hitam}}$ = Nilai fitur pada daerah gelap

Feature akan mendeteksi wajah menggunakan beberapa bentuk yang berbeda sesuai dengan bentuk wajah yang ada pada gambar 2.4 dimana bentuk feature terdapat sisi gelap dan terang yang memiliki nilai yang berbeda kemudian akan dihitung pada persamaan 2.1 di atas [23].



Gambar 2.4 Haar like-feature [23].

Pada gambar 2.4 di atas menjelaskan berbagai macam bentuk *feature* dalam algoritma *haar cascade* fitur-fitur dalam *haar cascade* pada umumnya berbentuk *rectangular feature* atau fitur-fitur segi empat, fitur tersebut terdiri dari beberapa blok *pixel*. Secara umum *rectangular feature* hanya memiliki tiga jenis yaitu *two-rectangle feature*, *three-rectangle feature*, dan *four-rectangle feature*. Namun terdapat beberapa fitur lain yang dapat digunakan dalam algoritma *haar cascade* yaitu *diagonal feature* dan *line feature* [18].

2.2.6 Haar Like-Feature

Haar like-feature merupakan bentuk *rectangular features* mengindikasikan citra gambar secara spesifik untuk mengenali objek berdasar pada nilai sederhana setiap fitur akan tetapi nilai sederhana tersebut bukan merupakan *pixel* dari nilai *image* dengan komputasi yang sangat cepat dengan bergantung pada jumlah *pixel*. Untuk mendeteksi wajah adalah dengan menggunakan titik geser Windows di setiap gambar untuk mencari apakah ada bagian dari gambar bentuknya seperti wajah atau tidak [24].

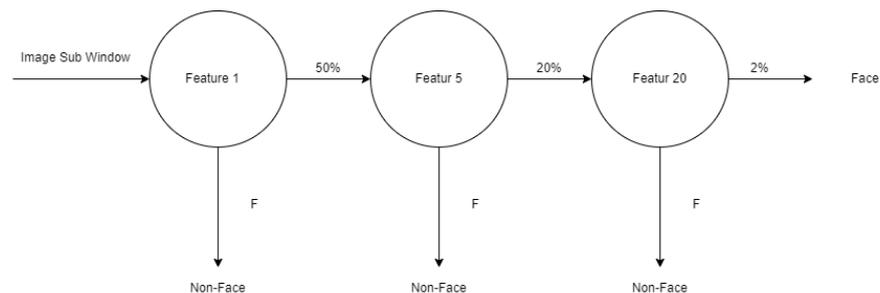


Gambar 2.5 Proses Deteksi Haar Like-Feature [25]

Pada proses representasi pola dari sebuah citra dapat dilihat pada gambar 2.5 di mana keseluruhan gambar (*sub Windows*) dilakukan identifikasi pola citra dengan *rectangular feature* pada proses *classifier*. Fitur ini berguna untuk membedakan objek mana yang ingin di deteksi dan objek yang tidak ingin di deteksi [24].

2.2.7 Cascade Classifier

Cascade classifier melakukan proses dari banyak fitur-fitur dengan mengorganisir dengan bentuk klasifikasi bertingkat. Terdapat tiga buah klasifikasi untuk menentukan apakah benar atau tidak ada fitur wajah pada fitur yang sudah dipilih. Pada *filter* pertama sub citra akan melalui klasifikasi dengan satu *filter* apa bila hasil nilai *filter* tidak memenuhi kriteria dari *filter* ke satu maka hasil akan ditolak [26], kemudian sub citra yang sudah memenuhi kriteria akan langsung masuk ke *sub Window* selanjutnya dan melakukan perhitungan nilai fitur Kembali, jika terdapat nilai ambang (*threshold*) maka dilanjutkan ke tahap selanjutnya sehingga jumlah sub citra yang lolos dalam klasifikasi semakin berkurang sampai mendekati *image* yang di deteksi [24].



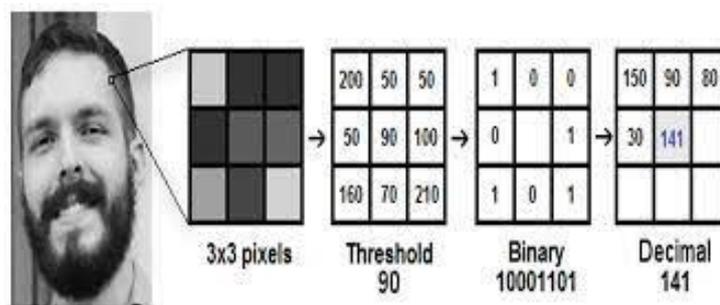
Gambar 2.6 Proses *Cascade Classifier*.

Pada gambar 2.6 merupakan serangkaian proses *filter* yang harus dilewati oleh setiap *classifier*. Setelah dilakukan serangkaian proses seperti pemilihan fitur dan klasifikasi bertingkat maka akan didapatkan sebuah hasil pendeteksian. Hasil pendeteksian bisa berupa wajah atau bukan wajah. proses klasifikasi bertingkat dilakukan maka, pada *image* tersebut akan ditandai dengan sebuah *rectangle* pada daerah wajah yang terdeteksi dan apabila tidak ada wajah terdeteksi maka, *image* tersebut tidak akan ditandai oleh sebuah *rectangle* [18]. Di bawah ini merupakan penjelasan tentang *classifier* :

1. *filter* pertama dipilih satu fitur *classifier* dengan persentase tingkat pendeteksian sebesar 100% dan sekitar 50% tingkat kesalahan.
2. *filter* kedua dipilih lima buah fitur *classifier* dengan persentase tingkat pendeteksian sebesar 100% dan 20% tingkat kesalahan (20% kumulatif).
3. *filter* ketiga dipilih 20 fitur *classifier* dengan persentase tingkat pendeteksian sebesar 100% dengan tingkat kesalahan sebesar 2% kumulatif.

2.2.8 Algoritma LBPH

LBPH adalah pengembangan dari metode *Local Binary Pattern* (LBP), digunakan untuk mengklasifikasi citra dengan performa yang lebih baik dalam identifikasi wajah. Metode ini menggabungkan histogram dan teknik LBP untuk mencirikan motif, corak, dan pola pada ciri-ciri wajah secara *real-time*. Penggunaan LBPH, terutama pada perangkat bergerak dengan sumber daya terbatas, memiliki keuntungan karena kalkulasinya yang sederhana. Metode ini memanfaatkan kamera pada laptop untuk membandingkan citra wajah *real-time* dengan data citra yang sudah ada dalam *data base*. Proses ini melibatkan ekstraksi fitur yang konsisten untuk membedakan objek atau karakter tertentu. Dalam tahap ini, histogram dihitung untuk membandingkan dan mencocokkan wajah yang terdeteksi pada kamera dengan data yang ada di dalam *data base*. LBPH memberikan peningkatan dalam mengidentifikasi wajah dalam situasi nyata dengan fokus pada karakteristik unik dalam citra [28].



Gambar 2.7 Sistem kerja algoritma LBPH [29].

Pada gambar 2.7 untuk melakukan perhitungan nilai histogram pada *haar cascade* dilakukan ekstraksi fitur dengan membandingkan intensitas *pixel* pada suatu titik dengan intensitas *pixel* pada titik sekitarnya. Algoritma LBPH dapat dijadikan alternatif dari *haar cascade* dalam proses pelatihan. Untuk mendapatkan

nilai histogram algoritma LBPH memilih nilai *threshold* yang kemudian dibandingkan dengan nilai yang ada pada titik sekitar maka akan di dapatkan pola biner, dari hasil pola biner yang di dapatkan kemudian diubah menjadi desimal dengan menjumlahkan nilai dalam pola biner yang memiliki nilai satu.

2.2.9 Integral Image

Integral *image* adalah suatu Teknik untuk menghitung nilai fitur dari citra secara cepat dengan menghitung nilai citra integral di setiap titik dengan merubah nilai dari setiap *pixel* menjadi citra baru pada citra abu-abu [30]. Untuk melakukan perhitungan fitur *haar* menggunakan area *tabel (Summed)* yang sering dikenal sebagai *integral image*, nilai *pixel* dari integral pada area *tabele* dapat dihitung dengan persamaan di bawah ini [31].

$$s(x,y) = i(x,y) + s(x-1,y) + s(x,y-1) - s(x-1,y-1) \dots \dots \dots (2.2)$$

Dimana :

- $s(x,y)$: merupakan nilai *pixel* pada area *tabel* yang dicari nilai integral posisi (x,y)
- $i(x,y)$: Nilai *pixel* asli citra pada posisi (x,y).
- $s(x-1,y)$: Nilai *pixel* pada Area *Tabel* pada posisi kiri *pixel* posisi (x,y).
- $s(x,y-1)$: Nilai *pixel* pada Area *Tabel* pada posisi atas *pixel* posisi (x,y).
- $s(x-1,y-1)$: Nilai *pixel* pada Area *Tabel* pada posisi kiri-atas *pixel* posisi (x,y).

2.2.10 Grayscale

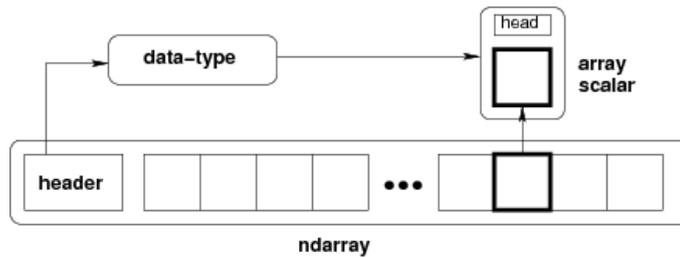
Grayscale merupakan salah satu citra yang nilai dari sebuah intensitas *pixel* berdasar pada derajat keabuan. Derajat warna hitam sampai dengan putih pada citra *Grayscale 8 bit* dibagi ke dalam 256 derajat keabuan. Dengan hasil akhir, warna hitam bernilai sempurna dengan representasi nilai 0 dan warna putih bernilai sempurna dengan representasi nilai 255 [32]. Citra RGB bisa dikonversi menjadi citra *Grayscale* dengan hasil hanya satu kanal warna yang ada pada gambar 2.8 . Persamaan yang digunakan untuk konversi citra RGB 24 *bit* menjadi citra *Grayscale 8 bit* adalah $Grayscale = 0,2989 * R + 0,5870 * G + 0,1140 * B$ [22].



Gambar 2.8 Contoh *image Grayscale* [33].

2.2.11 *Library Numpy*

Numpy (*Numerical Python*) adalah sebuah *library* pada bahasa pemrograman *python* yang memiliki fungsi untuk mengolah data berupa *numerical*. *Numpy* dilengkapi dengan sumber daya yang mampu merubah *python* menjadi suatu alat analisis dan pemodelan yang kuat. *Numpy* memiliki sebuah kelebihan yaitu penggunaan ruang memori yang lebih kecil dan *run-time* dilakukan secara cepat[14]. *Numpy* juga menyediakan penghubung yang efisien untuk mengoperasikan serta menyimpan data karena ukuran dari *array* bertambah besar yang membentuk inti dari seluruh ekosistem pembelajaran. *Library Python* menyediakan berbagai objek turunan seperti *matriks* dan *array* setra berbagai macam operasi cepat pada *array* termasuk matematika, logika, pengurutan, pemilihan, manipulasi bentuk dan lain sebagainya. *Array Numpy* dapat di definisikan sebagai objek terpenting dalam *Numpy* dengan tipe data *array* N-dimensi atau *ndarray*, *ndarray* dapat dijelaskan sebagai jenis koleksi item yang sama dengan menggunakan indeks nol. Setiap *ndarray* memiliki ukuran blok yang sama di memori masing-masing *ndarray* memiliki objek dari objek dengan tipe data *dtype* setiap item yang akan di ekstraksi *ndarray* maka akan di wakili oleh objek [34].



Gambar 2.9 Diagram hubungan antar *array*, objek tipe data, dan tipe data skala *array* [35].

Pada gambar 2.9 di atas merupakan contoh hubungan antara tiga elemen pokok yang digunakan untuk mewakili data dalam sebuah larik. Awalnya, terdapat larik *ndarray* yang bertindak sebagai wadah utama untuk menyimpan data dalam struktur tertentu. Selanjutnya, objek tipe data menjelaskan tata letak elemen tetap di dalam larik, memastikan interpretasi yang tepat saat data diambil. Terakhir, muncul objek skalar larik *Python* ketika suatu elemen dalam larik diakses, memungkinkan manipulasi data dan operasi pada tingkat elemen tunggal. Skema ini memberikan pandangan menyeluruh tentang cara ketiga komponen ini saling berhubungan, memberikan dasar yang kokoh dalam memahami pengelolaan data dalam larik untuk berbagai keperluan analisis dan pengolahan data di berbagai bidang. [34].

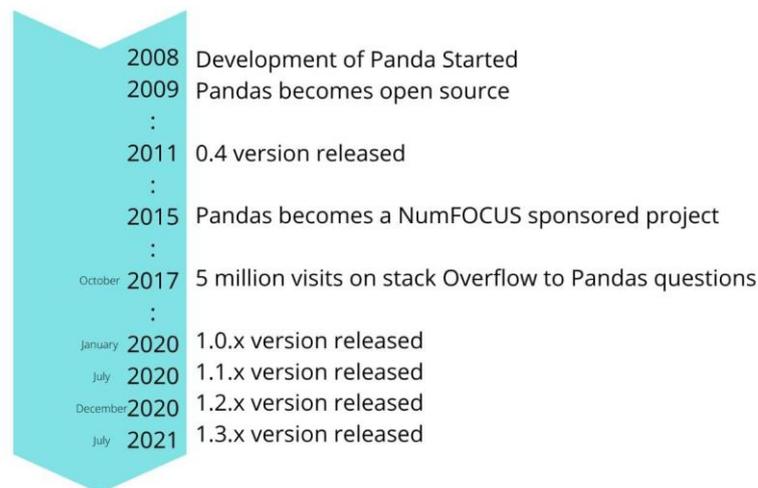
Tabel 2.2 Tipe data skalar berbeda yang ditentukan dalam *Numpy*.

No.	Tipe data	Keterangan
1.	bool	Boolean disimpan sebagai byte
2.	int	Tipe integer default
3.	uint	Bilangan bulat yang tidak bertanda
4.	float	Float presisi ganda
5.	complex	Bilangan yang diwakili oleh dua float

Tabel 2.2 Tabel tersebut menjelaskan tipe data dalam pemrograman dan penyimpanan data komputer. Bool merepresentasikan nilai kebenaran sebagai *True* atau *False*, disimpan sebagai *byte*. *Int* adalah tipe data *integer default* untuk bilangan bulat. *Uint* adalah *unsigned integer*, mewakili bilangan bulat positif. *Float* merepresentasikan bilangan pecahan dengan presisi ganda. *Complex* digunakan untuk bilangan kompleks, diwakili oleh dua nilai *float*.

2.2.12 *Library* pandas

Pandas merupakan modul *python* yang wajib di kuasai serta memiliki performansi tinggi dengan struktur data yang mudah digunakan dan tools analisis data untuk pemrograman *Python* dalam data *science* atau *big data*. *library* pandas menyediakan struktur dan analisis data yang dapat digunakan dengan Bahasa *python*, pandas digunakan untuk membersihkan data serta ditampilkan dalam bentuk *tabel* dan dapat membaca berbagai macam ekstrasi *File* diantaranya *csv*, *txt*, *tsv* dan lain sebagainya[36]. pengembangan *library* pandas dilakukan pada tahun 2008 oleh Wes McKinney yang menciptakan alat untuk menganalisis kuantitatif pada data, banyak perusahaan menggunakan perangkat lunak komersil berdasarkan pengguna seperti skrip atau makro pada data *spreadsheet* seperti *excel* lalu perkembangan *library* pandas melakukan kelanjutan pada tahun 2009 McKnney mempunyai struktur data serta modul fungsi yang masih berbentuk *prototipe*, nanti akan menjadi nama dari *library* pandas[37].



Gambar 2.10 Sejarah *Library* pandas. [37]

Gambar 2.10 menjelaskan tentang perkembangan *Library pandas* Pada tahun 2008-2009, Wes McKinney menciptakan library Pandas saat bekerja di AQR Capital Management. Versi awalnya mendapatkan perhatian di komunitas analisis data Python. Pada 2012-2013, Pandas mendapatkan struktur data kuat, *DataFrame*, untuk mengatasi data besar dan metode untuk analisis deret waktu. Pada 2014-2015, Pandas memudahkan analisis agregat dengan fitur *grouping* dan *pivot*. Pada 2018-2019, fitur *nullable integer* dan *string* ditambahkan. Pada 2020-2021, performa

ditingkatkan dan fitur untuk data semi-struktur diperkenalkan. Pandas terus berkembang sebagai library analisis data utama dalam Python.

2.2.13 Library Open CV

Open CV berperan aktif dalam pertumbuhan Komputer Vision dimana *Open CV* merupakan suatu *library* khusus yang digunakan untuk penglihatan komputer secara langsung (*real-time*) *Open CV* di kembangkan oleh peneliti intel Rusia, *library* ini dapat di unduh siapa saja tanpa crack atau berbayar karena *library* ini sudah dirilis oleh *Berkley Software Distributions* (BSD).[20] *Library Open CV* digunakan untuk mengelola sebuah citra dari *web cam* atau pun kamera yang akan di konversi menjadi bentuk digital dan diolah dalam komputer dengan tujuan untuk menyempurnakan kualitas gambar dalam proses identifikasi, *library* ini dapat digunakan dalam berbagai macam Bahasa pemrograman seperti *Python*, *java*, *C++*, *C*, dan *library* ini juga dapat digunakan di berbagai macam operasi sistem *Windows*, *MacOS*, *iOS*, *Linux* serta *Android* tanpa melakukan perubahan dalam source code.[38] *OpenCV* dapat dimanfaatkan untuk melakukan proses *Multi-core*, *CUDA*, dan *OpenGL* dalam pemrosesan parallel sebab *Library* ini sangat ringan. Guna meningkatkan *library OpenCV* banyak komunitas yang terus mengembangkan *library* ini sehingga jutaan unduhan untuk *Library* ini meningkat setiap harinya, selain *OpenCV* alat untuk melakukan *Image Processing* dan *Visi Computer* adalah *MATLAB*. Sering kali banyak orang bertanya tentang keuntungan *OpenCV* dibanding *MATLAB* [39].

Tabel 2.3 Perbandingan *OpenCV* dengan Matlab.

Parameter	OpenCV	Matlab
Kecepatan Program	Lebih Tinggi karena menggunakan Bahasa pemrograman C dan C++.	Lebih rendah dibandingkan dengan <i>OpenCV</i> .
Probabilitas	Dapat berjalan di semua sistem operasi.	Hanya dapat berjalan di <i>Windows</i> , <i>Linux</i> , dan <i>MAC</i> .

Parameter	OpenCV	Matlab
Sumber daya yang dibutuhkan	<i>Library</i> ini sangat ringan hanya menggunakan sedikit memori baik dari hard disk maupun ram membutuhkan sekurangnya 100 MB RAM.	MATLAB sangat besar, untuk mengistalnya membutuhkan lebih dari 15 GB Hard disk serta ram lebih dari 1 GB pada saat digunakan.
biaya	Dalam aplikasi komersil <i>OpenCV</i> sepenuhnya gratis .	Matlab merupakan perangkat yang berlisensi sehingga harus membayar untuk menggunakannya.
Kemudahan	<i>OpenCV</i> sulit digunakan sebab pada Matlab hanya memiliki sedikit dokumentasi dan sintak yang sulit di ingat.	Lingkungan pengembangan dengan sumber daya bantuan bawaan, yang memudahkan pemrogram baru untuk menggunakannya.

Pada table 2.3 Dalam membandingkan OpenCV dan Matlab, beberapa faktor utama menjadi sorotan. Kecepatan program OpenCV lebih unggul karena menggunakan bahasa pemrograman C dan C++. Fleksibilitas probabilitasnya terlihat dengan kemampuannya berjalan di semua sistem operasi, berbeda dari Matlab yang hanya terbatas pada Windows, Linux, dan MAC, OpenCV juga lebih ringan dalam penggunaan sumber daya, dengan memori dan RAM yang lebih minim, berbeda dengan Matlab yang memerlukan sumber daya yang lebih besar. Dari segi biaya, OpenCV gratis untuk aplikasi komersial dan non-komersial, sementara Matlab berbayar Kemudahan penggunaan juga menjadi pembeda. OpenCV mungkin lebih kompleks karena kurangnya dokumentasi dan sintaks yang kompleks. Di sisi lain, Matlab menyediakan lingkungan pengembangan yang lebih bersahabat dengan sumber daya bantuan internal yang kuat[38].

2.2.14 *Tensorflow dan Keras*

TensorFlow adalah sebuah perpustakaan (*library*) sumber terbuka yang dikembangkan oleh tim Google Brain. Tujuannya adalah untuk melakukan komputasi numerik dan *machine learning* dalam skala besar. Dalam *TensorFlow*,

terdapat banyak model yang digabungkan menggunakan algoritma *machine learning* dan *deep learning*. Untuk membangun aplikasi menggunakan *framework* TensorFlow, digunakan *Python* sebagai antarmuka pemrograman aplikasi (API) yang tidak hanya menyediakan antarmuka pengguna, tetapi juga memberikan performa yang tinggi saat menjalankan aplikasi tersebut. Dalam proses kerjanya, *TensorFlow* mampu menangani berbagai jenis masalah, seperti melatih dan menjalankan jaringan saraf tiruan untuk klasifikasi tulisan tangan, pengenalan gambar, penyisipan kata-kata, jaringan saraf rekuren, dan model urutan ke urutan untuk terjemahan. Selain itu, *TensorFlow* juga dapat digunakan untuk melakukan simulasi berbasis PDE (*Persamaan Diferensial Parsial*).[40] Di sisi lain, Keras merupakan sebuah antarmuka pemrograman neural *network* tingkat tinggi (*high-level*) yang ditulis dalam bahasa *Python*, dan dapat berjalan di atas *TensorFlow* [41].

2.2.15 Matplotlib

Matplotlib adalah perpustakaan yang berperan dalam pembuatan aplikasi grafis. Keunggulan perpustakaan ini terletak pada kemudahannya dalam membuat grafik dan plot dalam bahasa pemrograman Python. Matplotlib menjadi alat penting untuk mengembangkan aplikasi interaktif dengan grafik dua dan tiga dimensi, serta menghasilkan gambar berkualitas tinggi. Pengembang dapat dengan mudah menciptakan berbagai jenis grafik seperti batang, garis, kolom, dan pie, serta menggunakan fitur-fitur seperti histogram dan analisis spektrum daya[42]. Matplotlib memberikan fleksibilitas dalam menciptakan aplikasi dengan visualisasi data yang jelas dan beragam, menjadikannya alat yang sangat berharga bagi ilmuwan dan profesional di berbagai bidang. Selain itu, Matplotlib juga memungkinkan pembuatan gambar berkualitas tinggi, sementara membantu pengembang menciptakan aplikasi yang mudah digunakan dan menarik secara visual. Dengan Matplotlib, berbagai jenis grafik seperti batang, garis, kolom, dan pie dapat dihasilkan, serta histogram dan aplikasi yang berkaitan dengan analisis spektrum daya. Penggunaan Matplotlib memungkinkan pembuatan aplikasi dengan visualisasi data yang jelas dan beragam[43].

2.2.16 Jaringan Syaraf Tiruan (JST)

Dalam perkembangan teknik pengenalan wajah, Penggunaan Jaringan Saraf Tiruan (JST) telah menjadi pendekatan yang populer. Terobosan terbaru dalam teknik pengenalan wajah banyak melibatkan penerapan JST dengan pendekatan yang paling dominan adalah Jaringan Saraf Konvolusional (*Convolutional Neural Networks/CNN*). Kemampuan CNN dalam mengambil fitur-fitur kunci dari gambar, seperti tepi, bentuk, dan tekstur, berperan penting dalam deteksi wajah. Melalui pelatihan menggunakan data set yang berisi contoh wajah dan non-wajah, JST CNN mampu memahami pola-pola ini serta membedakan antara wajah dan objek lain dalam citra. Oleh karena itu, pendekatan ini memungkinkan pengembangan kemampuan deteksi wajah yang semakin canggih dan akurat[44].

Selain CNN, terdapat juga penggunaan arsitektur jaringan syaraf tiruan terkini, seperti jaringan syaraf *piksel-level* (*Pixel-level Neural Networks/PNN*) dalam deteksi wajah. PNN menggabungkan konsep pemrosesan piksel-level dengan kemampuan jaringan syaraf untuk mendeteksi wajah secara akurat. Dalam PNN, setiap piksel pada gambar dianggap sebagai *input* ke dalam jaringan syaraf dan melalui proses pelatihan, JST mempelajari representasi wajah yang kompleks dan mengklasifikasikan apakah piksel tersebut termasuk dalam wajah atau bukan. Pendekatan ini telah membawa peningkatan yang signifikan dalam deteksi wajah dan memberikan hasil yang lebih baik dalam menghadapi variasi kondisi pencahayaan dan transformasi geometris [45]. Salah satu perkembangan terkini dalam deteksi wajah menggunakan JST adalah penggunaan jaringan syaraf dengan perhatian visual (*Visual Attention Neural Networks/VANN*). VANN memodelkan kemampuan manusia dalam memusatkan perhatian pada bagian-bagian penting dalam gambar wajah. Dengan mempelajari pola distribusi perhatian visual pada wajah manusia, VANN dapat mengidentifikasi bagian wajah yang relevan seperti mata, hidung, dan mulut. Pendekatan ini memungkinkan deteksi wajah yang lebih akurat dan robust terhadap variasi kondisi gambar dan posisi wajah [44].

2.2.17 Graphical User Interface (GUI)

Graphical User Interface (GUI) sebuah program yang menyediakan tampilan jendela dengan tampilan kontrol grafik untuk digunakan *user* dalam melakukan interaksi dengan program. Bahasa pemrograman *python* tidak memiliki

fitur untuk membuat tampilan GUI maka dari itu *python* memerlukan modul *Tkinter* untuk membuat program GUI sederhana, Modul *tkinter* memuat 15 elemen GUI yang dapat membaca dan menerima masukan seperti *Button*, *Canvas*, *CheckButton*, *Entry*, *frame*, *Label*, *Listbox*, *Menu*, *Menubutton*, *Message*, *Radiobutton*, *Scale*, *Scrollbar*, *Text*, dan *Toplevel* [46]. *Tkinter* merupakan sistem antarmuka *Python* dalam *library tk GUI* serta menjadi *library* standar dari *Python* pada 1994 dengan rilisnya *Python* versi 1, Bahasa pemrograman *python* memiliki beberapa *tool* yang dapat di pilih dengan teknologi glamor yang dapat digambarkan dengan kata trending dan *hype* namun, tidak hanya memadai untuk beberapa aplikasi akan tetapi juga memiliki beberapa keunggulan diantaranya *tkinter* ada di *library* standar, *tkinter* stabil, *tkinter* hanya perangkat stabil, tampilan *default tkinter kuno*, *tkinter* tidak memiliki *widget* yang Komplek, dan *tkinter* sederhana dapat digunakan secara efektif [47].