

BAB 2

DASAR TEORI

2.1 KAJIAN PUSTAKA

Sebuah penelitian terdahulu meneliti terkait sinyal EKG yang digunakan untuk mengklasifikasikan tingkat kecemasan seseorang melalui beberapa tahap, yaitu pengambilan data, pra-pemrosesan sinyal, ekstraksi fitur, dan klasifikasi. Fitur-fitur penting dari sinyal EKG seperti *Heart Rate Variability* (HRV) dan gelombang R-R interval diekstraksi dan dianalisis menggunakan algoritma klasifikasi *Support Vector Machine* (SVM). Hasil penelitian menunjukkan bahwa metode ini mampu mengklasifikasikan tingkat kecemasan dengan akurasi sekitar 85%. Namun, terdapat kekurangan seperti jumlah sampel yang terbatas dan variasi individu dalam respons fisiologis terhadap stres. Selain itu, aplikasi dalam situasi nyata mungkin menghadapi tantangan berbeda dibandingkan dengan kondisi laboratorium yang terkontrol. Penelitian ini lebih fokus pada analisis HRV dan gelombang R-R interval tanpa mengeksplorasi potensi fitur lain dari sinyal EKG yang mungkin juga relevan. Diperlukan penelitian lebih lanjut dengan sampel yang lebih besar dan kondisi yang lebih variatif untuk meningkatkan validitas hasil [7].

Dalam penelitian [8], sinyal EKG dari responden digunakan sebagai data uji untuk sinyal EKG normal, sedangkan untuk data tes kelas abnormal, data yang digunakan diperoleh dari situs web penelitian, yaitu *physionet* dengan kelas fibrilasi atrium. Proses desain dalam sistem ini mencakup proses akuisisi data, pelatihan, ekstraksi fitur, pengujian dan klasifikasi dengan jaringan saraf tiruan. Berdasarkan hasil kinerja perangkat ini untuk merekam sinyal EKG pada responden diperoleh sinyal EKG normal karena hasil rekaman sinyal EKG memiliki kesamaan pada gelombang PQRST dengan target yang telah ditentukan. Sistem ini dapat mendeteksi klasifikasi jantung dengan mengenali karakteristik statistik dari dua kelas sinyal dan dilatih menggunakan jaringan saraf. Berdasarkan proses pengujian menggunakan jaringan saraf tiruan diperoleh akurasi 76,9%

Penelitian [9], menyatakan sinyal EKG digunakan untuk mengklasifikasikan detak jantung menggunakan *Convolutional Neural Networks*

(CNN). Metodenya melibatkan pengumpulan data EKG, pra-pemrosesan sinyal untuk mengurangi *noise*, ekstraksi fitur, dan pelatihan model CNN. Data EKG dikumpulkan dari berbagai sumber, kemudian dipra-pemroses menggunakan teknik *filtering*. Fitur-fitur penting dari sinyal EKG diekstraksi dan digunakan sebagai *input* untuk model CNN yang dilatih untuk mengklasifikasikan jenis detak jantung. Hasil penelitian menunjukkan bahwa model CNN yang dikembangkan mampu mencapai akurasi klasifikasi yang signifikan dalam mendeteksi berbagai jenis detak jantung abnormal. Namun, penelitian ini memiliki beberapa kekurangan. Model ini membutuhkan *dataset* yang besar dan berkualitas tinggi untuk pelatihan yang efektif, yang mungkin sulit diperoleh. Selain itu, model masih rentan terhadap *overfitting* jika tidak dilatih dengan benar. Implementasi dalam aplikasi klinis juga memerlukan uji validasi lebih lanjut untuk memastikan keandalan dan keamanan penggunaannya dalam lingkungan medis

Penelitian [10], mengusulkan sistem berbasis IoT untuk pemantauan dan deteksi pasien jantung. Sistem ini menggunakan pendekatan dua tahap: tahap pertama melibatkan pengumpulan data menggunakan protokol *routing* yang efisien, dan tahap kedua menggunakan fitur dalam-dalam untuk mengklasifikasikan citra EKG. Pada tahap kedua, penelitian ini menggunakan CNN ringan yang dikombinasikan dengan modul perhatian untuk ekstraksi fitur dari citra EKG. *Dataset* yang digunakan terdiri dari citra EKG 12-lead dengan empat kategori: detak jantung abnormal, infark miokard, riwayat *infark miokard*, dan EKG normal. Model ini mencapai akurasi yang sangat tinggi, yaitu 98,39%, menunjukkan efektivitas sistem ini dalam mengidentifikasi gangguan jantung

Penelitian [11], mengusulkan model hibrida yang menggabungkan *Convolutional Neural Networks* (CNN) dan *Long Short-Term Memory* (LSTM) untuk mendeteksi aritmia berdasarkan sinyal EKG. Model ini menggunakan data dari database MIT-BIH dan LTAF untuk pelatihan dan pengujian, dimana sinyal EKG diolah melalui filtrasi kebisingan dan teknik resampling untuk homogenisasi data. CNN digunakan untuk ekstraksi fitur tingkat tinggi dari sinyal EKG mentah, sementara LSTM menangani ketergantungan jangka panjang dalam data sinyal, yang penting untuk mendeteksi pola aritmia yang kompleks. Arsitektur model ini mencakup beberapa blok konvolusi diikuti dengan lapisan LSTM dan lapisan

dropout untuk mencegah *overfitting*, serta beberapa lapisan dense untuk mengubah fitur menjadi vektor linear untuk klasifikasi akhir. Model ini menunjukkan kinerja yang sangat baik dalam mendeteksi berbagai jenis aritmia jantung dengan akurasi tinggi, menunjukkan potensi besar dalam aplikasi klinis untuk pemantauan kesehatan jantung berbasis IoT. Hasil penelitian menunjukkan bahwa kombinasi CNN dan LSTM memberikan keuntungan dalam menangkap fitur spasial dan temporal dari sinyal EKG, yang esensial untuk deteksi aritmia yang akurat

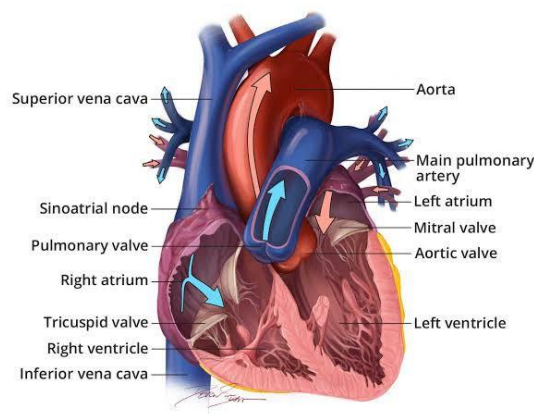
Penelitian [12], membahas penggunaan teknik pembelajaran mendalam untuk mengklasifikasikan aritmia jantung dari data EKG yang didigitalkan. Penelitian ini mengonversi gambar EKG menjadi sinyal *time-series* dan menerapkan berbagai model *deep learning*, seperti CNN (*Convolutional Neural Network*), LSTM (*Long Short-Term Memory*), dan model Pembelajaran Mandiri (SSL) menggunakan *autoencoder*. Model CNN mencapai akurasi tertinggi sekitar 92%, menunjukkan potensi besar dalam pemantauan EKG secara real-time dan akurat, yang penting untuk diagnosis dan perawatan jantung yang tepat waktu. Studi ini menunjukkan bahwa digitalisasi sinyal EKG dan penggunaan model pembelajaran mendalam dapat sangat membantu ahli jantung dalam memberikan diagnosis yang cepat dan akurat, meningkatkan kualitas perawatan kesehatan bagi pasien dengan penyakit kardiovaskular

Penelitian [13], membahas pengembangan model *deep learning* berlapis untuk klasifikasi aritmia berdasarkan sinyal EKG. Penelitian ini bertujuan untuk meningkatkan akurasi dalam mendeteksi berbagai jenis aritmia, yang sangat penting untuk diagnosis dan perawatan yang tepat. Menggunakan *dataset* seperti MIT-BIH *Arrhythmia Database*, penelitian ini melakukan preprocessing terhadap sinyal EKG, termasuk normalisasi dan segmentasi. Model yang diusulkan menggunakan arsitektur *Convolutional Neural Network* (CNN) berlapis-lapis yang mencakup beberapa lapisan konvolusi, *pooling*, dan *fully connected* untuk mengekstraksi dan mengabstraksi fitur dari sinyal EKG. Hasil penelitian menunjukkan bahwa model ini memiliki kinerja yang sangat baik dengan akurasi tinggi dibandingkan metode tradisional. Analisis kesalahan dilakukan untuk mengidentifikasi dan memahami jenis kesalahan yang terjadi serta bagaimana model ini bisa ditingkatkan lebih lanjut. Penelitian ini menyimpulkan bahwa model

deep learning berlapis yang dikembangkan mampu mengklasifikasikan sinyal EKG dengan sangat akurat, memberikan kontribusi signifikan dalam bidang deteksi aritmia, dan menyarankan arah penelitian selanjutnya untuk menggunakan lebih banyak data dan teknik *deep learning* yang lebih canggih

2.2 DASAR TEORI

2.2.1 JANTUNG



Gambar 2.1 Anatomi Jantung [14].

Jantung adalah organ vital dalam sistem peredaran darah manusia yang berfungsi memompa darah ke seluruh tubuh. Jantung terletak di dalam rongga dada, sedikit di sebelah kiri garis tengah tubuh. Organ ini memiliki empat ruang utama: dua atrium di bagian atas dan dua ventrikel di bagian bawah. Atrium kanan menerima darah yang kekurangan oksigen dari seluruh tubuh dan mengirimkannya ke ventrikel kanan, yang kemudian memompa darah tersebut ke paru-paru untuk mendapatkan oksigen. Sebaliknya, atrium kiri menerima darah beroksigen dari paru-paru dan mengirimkannya ke ventrikel kiri, yang kemudian memompa darah tersebut ke seluruh tubuh melalui aorta [14].

Jantung bekerja secara otomatis dan terus-menerus tanpa henti, dipicu oleh impuls listrik yang dihasilkan oleh nodus sinoatrial (SA) yang terletak di atrium kanan bagian atas. Impuls ini menyebabkan kontraksi otot jantung secara teratur, yang dikenal sebagai detak jantung. Impuls yang dihasilkan oleh nodus sinoatrial ini memiliki frekuensi yang paling besar dan paling cepat. Jadi dapat dikatakan

bahwa nodus SA inilah yang mengatur ritme jantung dan merupakan alat pacu jantung yang alami (*pacemaker*). Oleh karena itu, saat terjadi gangguan pada nodus SA, maka akan terciptanya irama jantung yang abnormal [15].

Pada keadaan normal, jantung berdetak 60-100 kali per menit. Keadaan ini biasanya disebut normal sinus rhythm karena setiap aliran listrik yang dihasilkan berasal dari *SA Node* [16]. Frekuensi dan irama detak jantung dikendalikan oleh sistem saraf otonom, yang merespons berbagai faktor seperti aktivitas fisik, stres, dan kondisi kesehatan tubuh [17]. Selain *SA Node*, terdapat sumber kelistrikan lainnya yaitu *AV Node*. *AV Node* atau *Antriventrikular Node* merupakan *pacemaker* kedua setelah *SA Node*. Ukuran *AV Node* lebih kecil dibandingkan dengan *SA Node*. Letaknya di atrium kanan bagian bawah. Normalnya, *AV Node* mengeluarkan impuls antara 40-60 kali per menit [18]. Dengan demikian, nodus ini berfungsi sebagai pemacu penunjang disaat nodus SA gagal mengalirkan listrik [19].

Fungsi utama jantung adalah untuk memastikan pasokan darah yang kaya oksigen dan nutrisi ke semua jaringan tubuh, sekaligus membuang produk-produk metabolisme seperti karbon dioksida dan limbah lainnya. Kegagalan jantung untuk berfungsi dengan baik dapat menyebabkan berbagai masalah kesehatan serius, termasuk penyakit jantung koroner, gagal jantung, dan aritmia. Penyakit jantung merupakan salah satu penyebab utama kematian di dunia, sehingga penting untuk menjaga kesehatan jantung melalui gaya hidup sehat, seperti berolahraga secara teratur, mengonsumsi makanan bergizi, dan menghindari kebiasaan merokok [20].

Faktor risiko penyakit jantung dapat diidentifikasi melalui pengukuran lemak tubuh dan indeks massa tubuh (IMT). Tingginya IMT dan lemak tubuh, seperti yang dicatat dalam jurnal tersebut, sering kali menjadi indikator obesitas yang signifikan, yang berhubungan erat dengan peningkatan risiko penyakit jantung. Selain itu, kondisi ini dapat menunjukkan adanya gangguan metabolik seperti resistensi insulin, yang juga meningkatkan potensi terjadinya penyakit kardiovaskular. Gaya hidup yang tidak sehat, termasuk kurangnya aktivitas fisik dan pola makan tidak seimbang, juga menjadi faktor tambahan yang sering dikaitkan dengan peningkatan risiko ini. Dengan memperhatikan faktor-faktor ini, penting untuk melakukan pengelolaan berat badan yang sehat dan

mempertimbangkan kebutuhan akan aktivitas fisik yang cukup sebagai langkah pencegahan yang efektif terhadap penyakit jantung [21].

2.2.2 ARITMIA

Aritmia jantung merupakan perubahan pola yang cepat dari detak jantung normal dan mengacu pada masalah frekuensi, regulasi dan masalah asal atau situasi rangsangan listrik . Aritmia secara umum mengacu pada setiap denyut atau irama jantung abnormal. Gejala umumnya yaitu detak jantung yang berantakan, yaitu saat berdenyut lebih cepat disebut takikardia (denyut jantung di atas 100 bpm) sebaliknya jika berdenyut terlalu lambat disebut bradikardia (denyut jantung di bawah 60 bpm). Selain itu, gejala seringkali melibatkan kelelahan, nyeri dada, bahkan tidak sadarkan diri [22].

Ada berbagai macam gangguan irama jantung, dengan fibrilasi atrium merupakan kasus paling umum. Meskipun tidak berbahaya secara langsung, akan menjadi persoalan jika tidak ditangani sebaik mungkin dikarenakan bisa mempengaruhi fungsi jantung, serta dapat menyebabkan kematian mendadak. Stroke memiliki peningkatan resiko sampai lima kali lipat, gagal jantung memiliki peningkatan sampai tiga kali lipat, sedangkan kematian beresiko meningkat sampai dua kali lipat.

Beberapa kasus gangguan irama jantung tidak mengancam kesehatan, tapi di kasus lainnya aritmia dapat menyebabkan masalah serius bagi kesehatan seperti pusing, sesak nafas, pingsan atau masalah serius pada jantung. Dari hasil wawancara dengan dokter jantung menyatakan pada kasus tertentu kerja jantung memiliki irama yang tidak normal contohnya pada saat seseorang bangun dari tidur, atau dari kondisi berlari menjadi kondisi berhenti secara tiba-tiba. Gangguan irama jantung bisa diamati dengan parameter dalam sinyal EKG, yang dimana berupa gelombang P, Q, R, S, dan T.

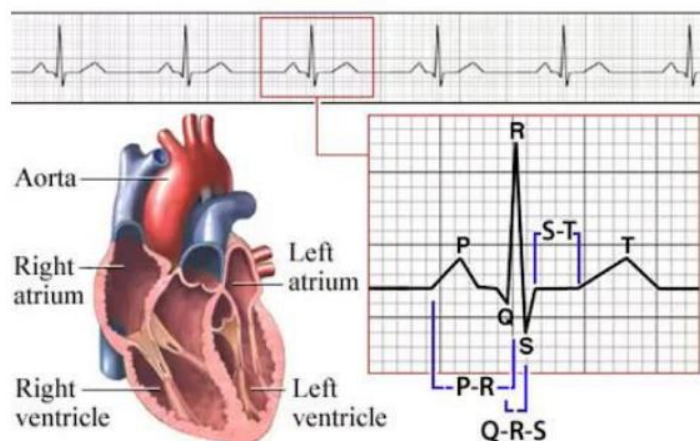
Kecemasan berlebih bisa menjadi penyebab aritmia tetapi bukan berarti semua aritmia merupakan kecemasan. Fibrilasi atrium, aritmia ventrikel, dan aritmia atrium merupakan indikasi aritmia yang berkaitan dengan kecemasan. Fibrilasi atrium (AF) disebabkan oleh karena sinyal listrik tidak terorganisir di atrium dan ventrikel, mengakibatkan denyut jantung menjadi lebih cepat, lambat,

atau tak beraturan [23]. Total kasus AF di Indonesia adalah 2,2 juta individu, dan 40% dari mereka dengan stroke.

Aritmia ventrikel merupakan gangguan irama jantung di ventrikel (bilik bawah jantung) sementara aritmia atrium merupakan gangguan irama jantung supraventrikular dan berlangsung di serambi atas jantung atau atrium. Selain itu, ada juga Atrioventrikular *Block* (AV Blok) yang merupakan . Gejala yang mungkin timbul dari keadaan ini yaitu sesak, cepat lelah saat beraktivitas, pusing, lemas, pingsan, dan nyeri dada [24]. Blok AV Derajat Pertama - Blok atrioventrikular (AV) derajat pertama adalah penundaan dalam sistem konduksi AV dan ditandai dengan perpanjangan interval PR melebihi batas maksimum yang dianggap normal (umumnya 0,20 detik). Sampai saat ini, blok AV derajat pertama dianggap sebagai kondisi yang sepenuhnya tidak berbahaya.

2.2.3 ELEKTROKARDIOGRAM

Elektrokardiogram (EKG) merupakan alat pendiagnosa untuk deteksi aliran listrik dalam jantung dengan bentuk grafis yang merekam perubahan listrik lalu dikaitkan dengan waktu. Pada EKG, sinyal terlihat sebagai sinyal P, QRS dan T seperti pada Gambar 1. Fungsi EKG meliputi pengidentifikasian bahwa adanya gangguan irama jantung, menilai kelainan otot jantung, mengevaluasi fungsi jantung, memperkirakan jika terdapat perubahan bentuk jantung menjadi lebih besar atau hipertropi atrium dan ventrikel, serta menilai efek obat jantung [8].



Gambar 2. 2 Sinyal Jantung pada EKG [8]

Variabilitas Denyut Jantung (HRV) adalah salah satu parameter utama yang diukur dari sinyal EKG. HRV mencerminkan variasi dalam interval waktu antara detak jantung yang berturut-turut, dan ini dipengaruhi oleh keseimbangan antara sistem saraf simpatis dan parasimpatis.

Cara menghitung Heart Rate yaitu dengan melihat jarak dari pada puncak gelombang R. Jika dilihat dengan mata masih sulit karena kotak-kotaknya kecil, maka cara praktis adalah dengan mengambil sehelai kertas kosong lalu ukur jarak gelombang R pada beat pertama dan beat kedua (dengan memberikan tanda pada kertas) lalu jarak yang sudah didapatkan dijadikan patokan untuk mengukur jarak puncak gelombang R pada beat-beat selanjutnya, jika semua jaraknya sama berarti irama teratur, dan sebaliknya jika tidak sama berarti irama tidak teratur. Irama EKG yang normal impulsnya (sumber listriknya) berasal dari nodus SA, maka iramanya disebut dengan irama sinus (“ *Sinus Rhythm*”).

Cara menentukan Heart Rate dengan irama teratur bisa dilihat sebagai berikut :

$$HR = \frac{300 \text{ (jumlah kotak besar dalam 60 detik)}}{\text{Jumlah kotak besar antara R-R}} \quad (2,1)$$

Atau

$$HR = \frac{1500 \text{ (jumlah kotak kecil dalam 60 detik)}}{\text{Jumlah kotak kecil antara R-R}} \quad (2.2)$$

Cara menentukan Heart Rate dengan irama abnormal bisa dilihat sebagai berikut :

Ambil EKG strip sepanjang 6 detik, hitung jumlah QRS dan kalikan 10 atau ambil dalam 12 detik dikalikan setengahnya (5).

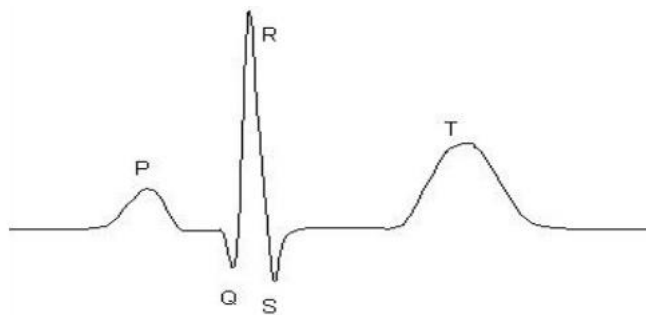
Proses pemrosesan sinyal dimulai dengan pra-pemrosesan sinyal EKG mentah untuk menghilangkan noise dan artefak yang dapat mengganggu analisis. Langkah-langkah ini termasuk filtrasi dan normalisasi untuk memastikan bahwa fitur yang diekstraksi akurat dan konsisten. Setelah sinyal diproses, mereka dibagi menjadi interval waktu 5, 10, dan 30 menit untuk membentuk *dataset* yang digunakan dalam pelatihan dan pengujian model *machine learning*. Dari sinyal EKG, penelitian ini mengekstraksi 23 fitur utama yang mencakup metrik domain waktu dan frekuensi. Fitur-fitur ini meliputi akar kuadrat rata-rata dari perbedaan berturut-turut (RMSSD), rasio LF ke HF (LF/HF), dan berbagai parameter lain yang

membantu dalam membedakan antara kondisi otonom normal dan disfungsional [25].

2.2.4 GELOMBANG EKG NORMAL

Sinyal yang diperoleh dari EKG normal akan terlihat menyerupai gambar 2. Karakteristik gelombang EKG normal antara lain:

- Gelombang P memiliki amplitudo kurang dari 0,3 mV dan perioda kurang dari 0,11 detik. Peningkatan durasi atau amplitudo gelombang P dapat menunjukkan adanya hipertrofi atrium atau kondisi lain yang mempengaruhi atrium.
- Gelombang Q memiliki amplituda minus 25% dari amplituda gelombang R. . Ini penting karena gelombang Q yang lebih besar dari itu dapat mengindikasikan adanya kerusakan pada septum atau infark miokard.



Gambar 2. 3 Gelombang EKG Normal [26]

- Gelombang R memiliki amplituda maksimum 3 mV.
- Gelombang S adalah defleksi negatif setelah gelombang R.
- Kompleks QRS merupakan gelombang Q, R dan S dengan durasi perioda 0,06-0,10 detik durasi perioda rata-rata 0,08 detik.
- Gelombang T memiliki amplituda minimum 0,1 mV[27].

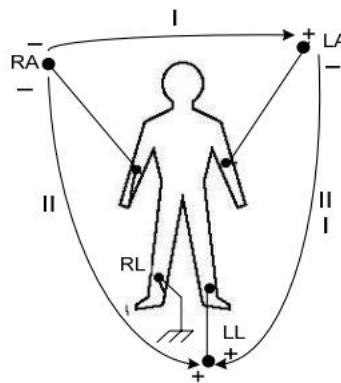
Amplitudo dan durasi kompleks QRS yang abnormal dapat mengindikasikan berbagai kondisi, termasuk hipertrofi ventrikel atau blokade cabang berkas. Gelombang T, yang mencerminkan repolarisasi ventrikel, biasanya memiliki amplitudo minimal 0,1 mV. Normalnya, gelombang T mengikuti arah gelombang QRS dalam *lead* yang sama. Namun, dalam *lead* precordial kanan, gelombang T bisa negatif. Perubahan dalam morfologi atau amplitudo gelombang

T dapat menunjukkan adanya gangguan repolarisasi ventrikel, seperti pada iskemia miokard atau ketidakseimbangan elektrolit [28].

2.2.5 SISTEM PEMANTAUAN *LEAD* EKG

Sinyal EKG yang dianalisis merupakan sinyal yang diambil dengan 3 *lead* sesuai segitiga Einthoven. Dalam sistem ini, sinyal EKG setiap *lead* mewakili perbedaan potensial antara anggota tubuh, termasuk :

- *Lead I* : perbedaan potensial antara lengan kiri (LA) dengan lengan kanan (RA)
- *Lead II* : perbedaan potensial antara kaki kiri (LL) dengan lengan kanan ((RA)
- *Lead III* : perbedaan potensial antara kaki kiri (LL) dengan lengan kiri (LA)[27].



Gambar 2. 4 Segitiga Einthoven [27]

2.2.6 CITRA

Citra digital adalah representasi visual dari objek atau pemandangan yang dikonversi ke dalam format digital. Biasanya, citra digital tersusun dari matriks piksel, di mana setiap piksel menyimpan informasi mengenai intensitas cahaya dan warna. Ada dua jenis utama citra digital, yaitu citra raster dan citra vektor. Citra raster terdiri dari piksel-piksel individu yang membentuk gambar, sedangkan citra vektor menggunakan garis dan bentuk yang didefinisikan secara matematis untuk membentuk gambar. Citra digital sangat penting dalam berbagai bidang seperti fotografi, pengolahan gambar medis, dan pengenalan wajah karena memudahkan analisis dan manipulasi gambar dengan menggunakan komputer [29].

Citra raster dan citra vektor memiliki struktur yang berbeda. Citra raster, seperti yang digunakan dalam fotografi digital, tersusun atas grid piksel yang setiap titiknya menyimpan informasi warna. Ukuran dan resolusi citra raster sangat bergantung pada jumlah piksel yang digunakan. Sebaliknya, citra vektor tidak bergantung pada resolusi karena dibentuk oleh objek geometris seperti garis, kurva, dan poligon yang didefinisikan oleh persamaan matematis. Jenis citra ini sering digunakan dalam desain grafis dan ilustrasi karena dapat diskalakan tanpa kehilangan kualitas.

Perbedaan utama antara citra raster dan vektor terletak pada cara mereka menyimpan dan mewakili informasi visual. Citra raster memiliki keunggulan dalam merepresentasikan detail dan gradasi warna yang halus, sehingga ideal untuk fotografi dan gambar yang memerlukan detail tinggi. Namun, citra raster cenderung memiliki ukuran file yang lebih besar dan kehilangan kualitas saat diperbesar. Sebaliknya, citra vektor, dengan basis matematisnya, lebih efisien dalam hal penyimpanan dan dapat diskalakan ke berbagai ukuran tanpa kehilangan kualitas. Ini membuat citra vektor sangat cocok untuk logo, ilustrasi, dan desain grafis yang memerlukan fleksibilitas ukuran [30].

Citra digital memainkan peran penting dalam berbagai bidang karena kemampuannya untuk dianalisis dan dimanipulasi dengan mudah menggunakan komputer. Dalam bidang medis, citra digital digunakan dalam pencitraan diagnostik seperti sinar-X, MRI, dan CT scan untuk membantu dokter dalam mendiagnosis dan merawat pasien. Di bidang keamanan, teknologi pengenalan wajah menggunakan citra digital untuk mengidentifikasi individu secara akurat. Dalam industri hiburan, citra digital digunakan dalam pembuatan film animasi dan efek visual. Selain itu, dalam bidang pendidikan, citra digital membantu dalam pembuatan bahan ajar yang lebih interaktif dan menarik [31].

Citra digital memiliki banyak keuntungan, termasuk kemudahan penyimpanan, kemampuan untuk dimanipulasi, dan kompatibilitas dengan berbagai perangkat lunak dan perangkat keras. Namun, ada juga beberapa kekurangan yang perlu dipertimbangkan. Misalnya, citra digital dapat terdegradasi kualitasnya saat mengalami kompresi atau penyimpanan berulang kali. Selain itu, citra raster dapat memerlukan ruang penyimpanan yang besar, terutama untuk gambar dengan

resolusi tinggi. Meskipun demikian, perkembangan teknologi terus meningkatkan kualitas dan efisiensi penggunaan citra digital, menjadikannya alat yang semakin penting dalam kehidupan sehari-hari [32].

2.2.7 IMAGE PROCESSING

Pengolahan citra (*image processing*) adalah teknik dalam bidang komputer dan teknologi informasi yang berkaitan dengan manipulasi dan analisis citra digital. Pengolahan citra mencakup berbagai operasi, mulai dari perbaikan kualitas gambar (*image enhancement*), pemulihan gambar (*image restoration*), hingga segmentasi gambar dan pengenalan pola. Teknik-teknik ini digunakan dalam berbagai aplikasi, seperti kedokteran (pencitraan medis), satelit (penginderaan jauh), dan keamanan (pengenalan wajah).

Teknik-teknik pengolahan citra ini sangat penting dalam berbagai aplikasi. Dalam bidang kedokteran, pengolahan citra digunakan dalam pencitraan medis untuk mendiagnosis dan memantau penyakit, seperti dalam MRI atau CT scan. Dalam bidang satelit dan penginderaan jauh, teknik ini digunakan untuk memproses citra satelit guna memantau perubahan lingkungan, pertanian, dan perencanaan tata kota. Di bidang keamanan, pengolahan citra berperan dalam pengenalan wajah dan sistem keamanan biometrik lainnya. Pengolahan citra juga digunakan dalam industri otomotif, misalnya dalam sistem bantuan pengemudi dan kendaraan otonom.

Perkembangan teknologi pengolahan citra juga didukung oleh kemajuan dalam komputasi dan algoritma, termasuk pembelajaran mesin dan jaringan saraf tiruan (*neural networks*). Algoritma ini memungkinkan analisis citra yang lebih cepat dan akurat, serta kemampuan untuk memproses data dalam jumlah besar. Sebagai hasilnya, pengolahan citra terus berkembang dan diterapkan dalam lebih banyak bidang, memperluas potensinya untuk memberikan solusi inovatif dalam berbagai masalah [33]. Pengolahan citra terdiri dari beberapa tahap utama:

1. Pra-pemrosesan (*Pre-processing*): Tahap ini melibatkan peningkatan kualitas gambar dengan mengurangi noise, meningkatkan kontras, dan melakukan transformasi geometris.

2. Segmentasi (*Segmentation*): Tahap ini bertujuan untuk membagi citra menjadi beberapa bagian atau objek yang berarti. Segmentasi sering digunakan untuk isolasi objek dalam citra yang akan dianalisis lebih lanjut.
3. Ekstraksi Fitur (*Feature Extraction*): Tahap ini melibatkan pengambilan karakteristik penting dari citra, seperti tepi, tekstur, dan bentuk objek.
4. Klasifikasi (*Classification*): Tahap ini melibatkan pengelompokan objek dalam citra ke dalam kategori tertentu berdasarkan fitur yang diekstraksi

2.2.8 KLASIFIKASI

Pengolahan Klasifikasi dalam pengolahan citra adalah proses mengkategorikan piksel dalam citra digital ke dalam kelas tertentu berdasarkan fitur yang diidentifikasi. Teknik klasifikasi sering digunakan dalam pengenalan objek, deteksi tepi, dan segmentasi citra. Dalam implementasinya, klasifikasi citra biasanya melibatkan dua tahap utama: ekstraksi fitur dan pembuatan model klasifikasi. Tahap ekstraksi fitur melibatkan identifikasi dan ekstraksi atribut-atribut atau fitur-fitur tertentu dari citra yang dapat digunakan untuk membedakan antara kelas-kelas yang berbeda. Fitur-fitur ini bisa mencakup warna, tekstur, bentuk, atau pola yang ada dalam citra.

Setelah fitur-fitur diekstraksi, langkah berikutnya adalah membangun model klasifikasi. Model ini bisa berupa pendekatan statistik sederhana seperti klasifikasi linier atau non-linier, atau menggunakan pendekatan yang lebih kompleks seperti jaringan saraf tiruan (*neural networks*) atau metode pembelajaran mesin lainnya. Penting untuk memilih fitur-fitur yang sesuai dan model klasifikasi yang tepat tergantung pada jenis data citra yang dihadapi dan tujuan dari analisis tersebut. Misalnya, dalam pengenalan objek, klasifikasi citra dapat digunakan untuk mengidentifikasi objek tertentu seperti manusia, mobil, atau hewan berdasarkan pada fitur-fitur yang telah dipelajari oleh model.

Terakhir, evaluasi dan validasi model klasifikasi citra merupakan langkah penting untuk memastikan keakuratan dan kehandalan dari hasil klasifikasi yang dihasilkan. Hal ini melibatkan pengujian model pada data yang belum pernah dilihat sebelumnya (*testing set*) untuk mengukur kinerja dan generalisasi dari model

terhadap data baru yang tidak terlibat dalam proses pelatihan. Secara keseluruhan, klasifikasi citra merupakan teknik yang penting dan sering digunakan dalam berbagai aplikasi pengolahan citra untuk mendukung pengambilan keputusan dan analisis berbasis visual. Beberapa metode klasifikasi yang umum digunakan termasuk:

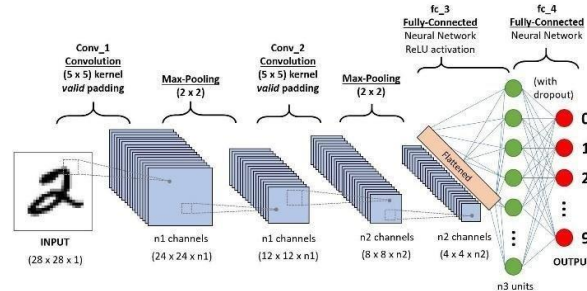
1. *K-Nearest Neighbors* (K-NN): Metode ini mengklasifikasikan objek berdasarkan kedekatan dengan objek-objek lain yang sudah diberi label.
2. *Support Vector Machines* (SVM): Metode ini mencari *hyperplane* optimal yang memisahkan data dari berbagai kelas.
3. *Convolutional Neural Networks* (CNN): Jaringan saraf tiruan yang digunakan untuk pengenalan pola dalam citra.

2.2.9 CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional Neural Network (CNN) adalah jenis jaringan saraf tiruan yang dirancang khusus untuk pemrosesan data *grid* seperti gambar. CNN bekerja dengan memproses citra melalui serangkaian lapisan konvolusi dan *pooling* yang mengekstraksi fitur-fitur penting dari citra, sehingga memungkinkan jaringan untuk mengenali dan mengklasifikasikan objek di dalam gambar dengan sangat efektif. Pada lapisan konvolusi, *filter* atau kernel kecil diterapkan pada citra untuk mendeteksi fitur-fitur lokal seperti tepi, sudut, dan tekstur. Setiap *filter* menghasilkan peta fitur (*feature map*) yang menunjukkan lokasi fitur tertentu dalam citra. Proses ini membantu jaringan untuk menangkap informasi spasial dan hierarkis dari gambar, yang sangat penting untuk tugas pengenalan pola dan klasifikasi [33].

Lapisan *pooling*, yang biasanya mengikuti lapisan konvolusi, berfungsi untuk mengurangi dimensi peta fitur sambil mempertahankan informasi yang paling penting. *Max pooling* adalah metode *pooling* yang umum digunakan, di mana nilai maksimum dari setiap wilayah *non-overlapping* diambil untuk menghasilkan peta fitur yang lebih kecil. Ini tidak hanya mengurangi ukuran data, tetapi juga membuat model lebih tahan terhadap pergeseran dan distorsi kecil dalam gambar. Dalam proses pelatihan CNN, jaringan mempelajari bobot *filter* yang optimal untuk mengekstraksi fitur yang paling relevan untuk tugas tertentu.

Backpropagation dan optimasi gradien digunakan untuk memperbarui bobot ini berdasarkan kesalahan yang dihitung antara prediksi jaringan dan label sebenarnya. Dengan cara ini, CNN dapat belajar mengenali pola kompleks dalam data gambar secara otomatis, tanpa memerlukan ekstraksi fitur manual [34].



Gambar 2. 5 Arsitektur CNN [35]

Keberhasilan CNN dalam pengenalan dan klasifikasi citra tidak terlepas dari kemampuannya untuk mengembangkan representasi fitur yang lebih abstrak dan bermakna pada setiap lapisan berturut-turut. Lapisan awal mungkin mendeteksi fitur rendah seperti tepi, sementara lapisan lebih dalam dapat mendeteksi bentuk dan objek yang lebih kompleks. Hal ini memungkinkan CNN untuk menggeneralisasi dari data pelatihan ke data baru dengan sangat baik .

Selain itu, arsitektur CNN modern sering kali menggunakan teknik seperti *batch normalization*, *dropout*, dan berbagai strategi regularisasi lainnya untuk meningkatkan kinerja dan mencegah *overfitting*. *Batch normalization* menormalkan aktivasi di setiap mini-*batch*, membantu stabilitas dan konvergensi pelatihan. *Dropout* secara acak mengabaikan unit dalam jaringan selama pelatihan untuk mencegah jaringan menjadi terlalu tergantung pada fitur-fitur tertentu dan meningkatkan generalisasi. Implementasi CNN yang populer termasuk arsitektur seperti AlexNet, VGGNet, GoogleNet, dan ResNet, masing-masing memperkenalkan inovasi yang meningkatkan kinerja pada tugas pengenalan citra. AlexNet, misalnya, memperkenalkan penggunaan ReLU dan *dropout* yang luas, sementara ResNet memperkenalkan koneksi residual yang memungkinkan pelatihan jaringan sangat dalam tanpa masalah degradasi [35].

a Lapisan Konvolusi (*Convolutional Layer*)

Lapisan ini adalah inti dari CNN, yang melakukan operasi konvolusi pada citra *input* menggunakan *filter* atau kernel untuk menghasilkan peta

fitur. Setiap *filter* mendeteksi fitur tertentu dalam citra, seperti tepi, tekstur, atau pola. Proses ini menghasilkan peta fitur yang memperlihatkan keberadaan fitur-fitur tersebut di berbagai lokasi dalam citra.

$$(I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n) \quad (2.3)$$

Keterangan :

I adalah matriks *input* (misalnya, gambar).

K adalah kernel (*filter*) yang diterapkan pada *input*.

(i,j)(i, j)(i,j) adalah koordinat dalam matriks *output* [36].

b Lapisan *Softmax*

Lapisan *softmax* adalah salah satu jenis lapisan yang umum digunakan dalam jaringan saraf terutama pada tugas klasifikasi multikelas. Fungsi utama dari lapisan *softmax* adalah untuk menghasilkan distribusi probabilitas yang normalisasi dari keluaran jaringan saraf sehingga setiap *output* mewakili probabilitas prediksi untuk setiap kelas yang mungkin.

Secara matematis, lapisan *softmax* memproses vektor *input* $z = (z_1, z_2, \dots, z_K)$ di mana z_k adalah nilai *input* untuk kelas *ke-k*. Lapisan ini menghitung probabilitas p_i untuk setiap kelas *i* sebagai berikut:

$$P_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.4)$$

Di mana e adalah bilangan Euler (e), dan K adalah jumlah kelas. Operasi *softmax* ini memastikan bahwa semua nilai probabilitas p_i berada dalam rentang $[0, 1]$ dan total probabilitas untuk semua kelas adalah 1, mirip dengan distribusi probabilitas yang valid.

Lapisan *softmax* sering digunakan sebagai lapisan *output* dalam jaringan saraf untuk klasifikasi multikelas karena sifatnya yang menghasilkan distribusi probabilitas. Ketika melakukan pelatihan (*Training*), *softmax* sering dipasangkan dengan fungsi *loss* seperti *cross-entropy loss*, yang memungkinkan jaringan untuk meminimalkan kesalahan prediksi probabilitas terhadap label yang sebenarnya.

Selain digunakan dalam *output* klasifikasi, *softmax* juga dapat digunakan sebagai penghubung antara lapisan-lapisan dalam jaringan saraf untuk mengintegrasikan *output* dari lapisan sebelumnya ke dalam format yang dapat diinterpretasikan sebagai probabilitas kelas-kelas yang berbeda.

c *Batch Normalization Layer*

Layer ini menormalkan *output* dari *layer* sebelumnya, biasanya digunakan setelah *convolutional layer* dan sebelum *activation function*. *Batch Normalization Layer* ini berfungsi mengurangi perubahan distribusi *input* antar *layer*, mempercepat pelatihan, dan membantu stabilisasi jaringan [37].

d *Dropout Layer*

Layer ini bekerja dengan mengabaikan beberapa neuron secara acak selama proses pelatihan berlangsung. Fungsi *dropout layer* yaitu untuk mengurangi *overfitting* dengan mencegah neuron bergantung terlalu banyak satu sama lain [38].

e *Flatten Layer*

Flatten mengubah matriks fitur yang semula 2D menjadi *vector* 1D. Fungsi *layer* ini yaitu untuk mempersiapkan data untuk masuk ke dalam *fully connected layer* [39].

f *Optimizer Adam W*

Optimizer ini merupakan salah satu bagian dari *Adam Optimizer* dengan penambahan *weight decay* untuk regularisasi. Fungsi dari *optimizer* ini yaitu untuk mengoptimalkan proses pelatihan dengan menyesuaikan tingkat pembelajaran (*learning rate*) adaptif. Selain itu, *weight decay* pada *optimizer* ini berfungsi untuk mencegah terjadinya *overfitting* [40].

g *Lapisan Pooling (Pooling Layer)*

Lapisan *pooling* mengurangi dimensi peta fitur yang dihasilkan oleh lapisan konvolusi sambil mempertahankan informasi penting. *Pooling* biasanya dilakukan dengan operasi maksimal (*max pooling*) atau rata-rata (*average pooling*), yang memilih nilai maksimum atau rata-rata dalam jendela kecil dari peta fitur. *Pooling* membantu mengurangi jumlah parameter dan komputasi dalam jaringan, serta memberikan sedikit ketahanan terhadap perubahan kecil dalam posisi fitur.

$$MaxPool(X)_{i,j} = \max_{(m, n) \in pool_region} X_{i+m,j+n} \quad (2.5)$$

Keterangan :

X adalah matriks *input*.

$pool_region$ adalah ukuran daerah *pooling* (misalnya, 2x2) [41].

h ReLu (*Rectified Linear Unit*)

Fungsi aktivasi ini diterapkan setelah konvolusi untuk menambahkan non-linearitas ke dalam model. Rumusnya adalah:

$$F(x) = \max(0, x) \quad (2.6)$$

i Lapisan *Fully Connected (Dense Layer)*

Lapisan *fully connected*, atau *dense layer*, adalah lapisan di mana setiap neuron terhubung ke semua neuron di lapisan sebelumnya. Setelah peta fitur dari lapisan konvolusi dan *pooling* diratakan menjadi vektor, vektor ini diberikan ke lapisan *fully connected* untuk klasifikasi akhir. Lapisan ini bertanggung jawab untuk menggabungkan fitur-fitur yang diekstraksi dan menghasilkan *output* akhir dari jaringan, seperti prediksi kelas dalam masalah klasifikasi.

$$y = f(Wx + b) \quad (2.7)$$

Keterangan :

W adalah matriks bobot.

x adalah vektor *input*.

b adalah bias.

f adalah fungsi aktivasi, seperti ReLU atau *Softmax*.

Dengan menggunakan kombinasi lapisan-lapisan ini, CNN mampu belajar dan mengenali pola kompleks dalam citra, membuatnya sangat efektif untuk berbagai aplikasi pengenalan pola dan klasifikasi citra [42].

2.2.9.1 MATRIK EVALUASI

a. Akurasi

$$Akurasi = \frac{(TP+TN)}{TP+FP+FN+TN} \quad (2.8)$$

Keterangan :

TP (<i>True Positive</i>)	Jumlah kasus positif yang diprediksi dengan benar.
TN (<i>True Negative</i>)	Jumlah kasus negatif yang diprediksi dengan benar.
FP (<i>False Positive</i>)	Jumlah kasus negatif yang diprediksi sebagai positif (kesalahan tipe I).
FN (<i>False Negative</i>)	Jumlah kasus positif yang diprediksi sebagai negatif (kesalahan tipe II).

b. *Loss*

$$\text{BCE Loss} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(y_i) + (1 - y_i) \log(1 - y_i)] \quad (2.9)$$

Keterangan :

N	: Jumlah total sampel data.
y_i	: Nilai sebenarnya dari sampel data ke-i (0 atau 1).
$(1 - y_i)$: Kebalikan dari nilai sebenarnya y_i (jika $y_i = 1$, maka $1 - y_i = 0$, dan sebaliknya).
$\log(1 - y_i)$: Logaritma dari kebalikan nilai sebenarnya y_i .
$\log(y_i)$: Logaritma dari nilai sebenarnya y_i .

Sebagai fungsi objektif, merupakan nilai invers dari IoU dan dihitung sebagai 1 dikurangi IoU. Fungsi objektif ini menjadi parameter yang dioptimalkan selama pelatihan model untuk meningkatkan ketepatan prediksi terhadap label sebenarnya.

2.2.10 **MOBILENETV2**

Mobilenetv2 adalah arsitektur jaringan neural konvolusi (CNN) yang diperkenalkan oleh Sandler et al. pada tahun 2018 dan dikembangkan oleh *Google* dengan fokus pada efisiensi komputasi untuk aplikasi *mobile* dan *embedded*. Arsitektur ini menggabungkan beberapa inovasi untuk meningkatkan efisiensi dan kinerja dibanding pendahulunya *MobileNetV1*. *Mobilenetv2* dirancang untuk perangkat dengan keterbatasan daya dan memori, seperti perangkat seluler dan *embedded systems*, tanpa mengorbankan akurasi.

Mobilenetv2 menggunakan blok dasar yang disebut "*Inverted Residual Block*" dengan penggunaan intensif *depthwise separable convolutions*. Bagian-bagian yang penting dari *Mobilenetv2* terdiri dari tiga bagian penting yaitu :

Depthwise Separable Convolution: Mobilenetv2 menggunakan *layer depthwise separable convolution*, yang terdiri dari dua tahap: *depthwise convolution* (menerapkan kernel pada setiap saluran input secara terpisah) dan *pointwise convolution* (kombinasi dengan 1×1 *convolution* untuk menggabungkan saluran hasil dari tahap sebelumnya).

Linear Bottleneck: Mobilenetv2 menggunakan *bottleneck residual block* yang dilengkapi dengan fungsi aktivasi linear di antara lapisan konvolusi.

Inverted Residuals with Linear Bottleneck: Mobilenetv2 Struktur ini memungkinkan untuk meningkatkan kapasitas model dengan menambahkan residu *invert* (yang mengurangi dimensi di dalam dan memperluasnya di luar) [43].

Selain itu terdapat bagian blok yang bernama *Inverted Residual Block*, dimana blok ini menggunakan konsep *residual connection* seperti pada *ResNet*, namun dengan inversi struktur. Dalam blok ini, fitur dimensi rendah diperbesar ke dimensi yang lebih tinggi sebelum dilakukan *convolusi depthwise*, kemudian dikembalikan ke dimensi yang lebih rendah. Hal ini membantu dalam menjaga informasi penting selama proses konvolusi.

Arsitektur dari *Mobilenetv2* terdiri dari serangkaian *inverted residual blocks* yang dikombinasikan dengan beberapa lapisan standar seperti *fully connected layers* di bagian akhir. Arsitektur ini dirancang untuk menjadi lebih efisien dalam komputasi dan penggunaan memori dibandingkan arsitektur CNN konvensional. Pengaplikasiannya dimana *Mobilenetv2* ini sering digunakan dalam aplikasi yang memerlukan inferensi cepat dan efisien di perangkat dengan sumber daya terbatas, seperti pengenalan objek pada ponsel pintar, deteksi wajah, dan klasifikasi gambar [44].

2.2.11 Visual Geometry Group (VGG)

Visual Geometry Group (VGG) adalah arsitektur jaringan neural konvolusi yang dikembangkan oleh *Visual Geometry Group* dari University of Oxford. Arsitektur ini pertama kali diperkenalkan oleh Simonyan dan Zisserman pada tahun

2014. VGG adalah arsitektur CNN yang terkenal karena kesederhanaan dan kedalamannya serta penggunaan lapisan konvolusi kecil berukuran (3x3) secara berulang dalam semua lapisan. Meskipun lebih berat secara komputasi dibandingkan dengan arsitektur *modern* seperti *Mobilenetv2*, VGG terkenal karena keefektifannya dalam mempelajari representasi gambar yang abstrak. Dua varian utama dari VGG adalah VGG16 dan VGG19, yang dinamai berdasarkan jumlah total lapisan dalam jaringan. VGG16 terdiri dari 13 lapisan konvolusi dan 3 lapisan *fully connected* sedangkan VGG19 terdiri dari 16 lapisan konvolusi dan 3 lapisan *fully connected* [45].

Arsitektur dari VGG menggunakan blok konvolusi yang terdiri dari beberapa lapisan konvolusi 3x3, diikuti dengan lapisan *pooling* (*max-pooling*). Seluruh jaringan diakhiri dengan beberapa lapisan *fully connected* dan lapisan *softmax* untuk klasifikasi. Selain itu, *convolutional layers*nya menggunakan filter 3x3 dengan *stride* dan *padding* yang tetap. Hal ini berfungsi untuk menjaga dimensi *spatial* dari *input*. Untuk arsitektur VGG selanjutnya yaitu *pooling layers*. Menggunakan *max-pooling 2x2* dengan *stride 2* untuk mengurangi dimensi *spatial*. Terakhir, ada arsitektur Bernama *Fully Connected Layers* yang dimana terdapat beberapa lapisan *fully connected* yang diakhiri dengan lapisan *softmax* untuk menghasilkan probabilitas klasifikasi.

Keunggulan utama dari VGG adalah kesederhanaan arsitekturnya yang hanya menggunakan konvolusi 3x3 dan *pooling 2x2* secara konsisten. Hal ini membuat VGG mudah untuk diimplementasikan dan dioptimalkan. Namun VGG juga memiliki kelemahan. Salah satu kelemahan VGG adalah ukurannya yang besar dan penggunaan memori yang tinggi, membuatnya kurang efisien untuk diterapkan pada perangkat dengan keterbatasan sumber daya seperti perangkat seluler. Untuk penggunaannya, VGG sendiri telah digunakan secara luas dalam berbagai tugas visi komputer seperti klasifikasi gambar, deteksi objek, dan segmentasi citra. Arsitektur ini juga sering digunakan sebagai *backbone* untuk model yang lebih kompleks dalam *transfer learning* [45].

2.2.12 PYTHON

Python adalah bahasa pemrograman tingkat tinggi yang dirancang untuk kemudahan penggunaan dan keterbacaan kode. *Python* terkenal dengan sintaks

yang sederhana dan elegan, yang membuatnya mudah dipelajari dan digunakan oleh pemula sekalipun [46]. Bahasa ini mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek, prosedural, dan fungsional. Sampai saat ini, *Python* juga merupakan salah satu bahasa pemrograman yang banyak digunakan oleh perusahaan besar maupun para developer untuk mengembangkan berbagai macam aplikasi berbasis desktop, *web* dan *mobile* .

Di bidang analisis data dan *machine learning*, *Python* memiliki pustaka-pustaka terkenal seperti *Pandas*, *NumPy*, dan *TensorFlow* yang memungkinkan para ilmuwan data untuk melakukan analisis kompleks dan membangun model prediktif. *Python* juga sering digunakan untuk pengolahan file dan manajemen sistem [47].

Beberapa fitur penting *Python* yang membuatnya unggul adalah ekosistem pustaka yang luas, komunitas yang aktif, dan dukungan lintas *platform*. Pustaka *Python* mencakup berbagai bidang, mulai dari *web scraping* (*BeautifulSoup*), visualisasi data (*Matplotlib*), hingga kecerdasan buatan (*Keras*). Komunitas *Python* yang aktif menyediakan berbagai tutorial, dokumentasi, dan dukungan bagi pengguna baru. Selain itu, *Python* dapat dijalankan di berbagai sistem operasi, termasuk Windows, MacOS, dan Linux, yang membuatnya sangat fleksibel untuk berbagai proyek. *Python* juga sering digunakan dalam pendidikan karena kesederhanaannya, yang membantu siswa memahami konsep pemrograman dasar dengan lebih cepat.

2.2.13 GOOGLE COLAB

Google Colab adalah sebuah IDE untuk pemrograman *Python* dimana pemrosesan akan dilakukan oleh server *Google* yang memiliki perangkat keras dengan performa yang tinggi [11]. Dari sisi perangkat lunak, *Google Colab* telah menyediakan hampir sebagian besar pustaka (*library*) yang dibutuhkan. Dalam penelitian ini pustaka yang dibutuhkan adalah *Keras*, *TensorFlow*, *NumPy*, *Pandas*, dan pendukung lainnya, misalnya untuk pembuatan grafik lewat *Matplotlib*. Bahkan seluruh versi disediakan, misalnya *TensorFlow* versi 1.x maupun 2.x tersedia, begitu juga versi *Python* yang mulai dari versi 2.x hingga 3.x. Dari sisi perangkat keras, *Google Colab* menyediakan layanan berupa media penyimpanan yang terintegrasi dengan *Google Drive*, prosesor yang berupa CPU, GPU, dan TPU,

serta RAM. Dengan jaminan kemampuan servernya yang stabil hampir keseluruhan pemrosesan tidak menemukan kendala dengan *Google Colab* selama koneksi jaringan internet lancar.

Salah satu fitur unggulan dari *Google Colab* adalah dukungannya terhadap GPU (*Graphics Processing Unit*) dan TPU (*Tensor Processing Unit*), yang sangat penting untuk mempercepat komputasi dalam *deep learning* dan tugas-tugas berbasis data lainnya. Pengguna dapat memilih antara berbagai akselerator *hardware* berdasarkan kebutuhan spesifik mereka dan kompleksitas model yang digunakan. Kemampuan ini secara signifikan mengurangi waktu pelatihan untuk model *deep learning* dibandingkan dengan menjalankannya pada CPU tradisional, sehingga meningkatkan produktivitas dan kecepatan eksperimen.

Dalam hal dukungan perangkat lunak, *Colab* sudah dilengkapi dengan banyak pustaka *Python* yang umum digunakan, termasuk untuk *deep learning* dan visualisasi data. Ini termasuk *TensorFlow* dan *PyTorch* untuk *deep learning*, *scikit-learn* untuk algoritma *machine learning*, serta *Matplotlib* dan *seaborn* untuk visualisasi data. Platform ini juga memungkinkan pengguna untuk menginstal pustaka tambahan sesuai kebutuhan, memberikan fleksibilitas dalam menyesuaikan lingkungan pengembangan mereka sesuai dengan persyaratan proyek.

Selain itu, integrasi yang lancar dengan *Google Drive* mempermudah manajemen dan penyimpanan data. Pengguna dapat dengan mudah mengimpor *dataset* dari *Google Drive* ke *notebook Colab* mereka dan mengeksport hasil atau model yang sudah dilatih kembali ke *Drive*. Integrasi ini menyederhanakan alur kerja dan memastikan bahwa data tersimpan dengan aman dan dapat diakses, meningkatkan utilitas platform ini untuk riset dan pengembangan berbasis data.

Secara keseluruhan, *Google Colab* merupakan alat yang serbaguna dan kuat bagi pengembang *Python* dan ilmuwan data, menawarkan sumber daya komputasi berkinerja tinggi, dukungan pustaka yang luas, fitur kolaboratif, dan integrasi *cloud* yang lancar. Baik untuk *prototyping* model *machine learning*, menganalisis *dataset* besar, atau mengajar pemrograman *Python*, *Colab* menyediakan lingkungan yang ramah pengguna dan dapat diskalakan yang mempercepat inovasi dan kolaborasi dalam bidang yang didorong oleh data [48].