

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Analisa Perancangan Tedahulu

Referensi penelitian terdahulu dengan menggunakan metode *Convolutional Neural Network* (CNN) untuk memproses klasifikasi gambar dan objek penelitian dirangkum dalam table berikut.

Tabel 2.1 Penelitian Terdahulu

No	Penulis	Tahun	Metode	Judul	Hasil
1	Mohanty, Hunghe and Salathe, university	2016	<i>Convolutional Neural Network</i> (CNN)	<i>Using Deep Learning for Image based plant Disease Detection.</i>	Pada penelitian ini, dilakukan identifikasi jenis penyakit yang menyerang tanaman menggunakan dataset berupa 54.306 gambar daun tomat yang terdiri dari sampel yang sehat dan sakit.

					<p>Untuk melakukan identifikasi tersebut, digunakan algoritma <i>Convolutional Neural Network</i> (CNN). Model yang digunakan dalam penelitian ini adalah GoogleNet yang telah dilatih sebelumnya, dan model tersebut mencapai tingkat akurasi sebesar 99,35%. Dengan menggunakan model GoogleNet yang terlatih tersebut, peneliti dapat</p>
--	--	--	--	--	--

					<p>mengklasifikasikan gambar daun tomat ke dalam 14 jenis tanaman dan 26 jenis penyakit yang berbeda. Dalam proses identifikasi, algoritma CNN mampu mendeteksi penyakit pada tanaman dengan tingkat akurasi yang sangat tinggi.</p>
2	Brahimi Boukhalifa and Moussaomi	2017	<i>Convolutional Neural Network (CNN)</i>	<i>Deep Learning for Tomato Diseases and Symptoms Visualization</i>	<p>Penelitian ini bertujuan untuk mengidentifikasi penyakit pada daun tomat. Dataset yang digunakan terdiri</p>

					<p>dari 14.828 gambar daun tomat yang terkena penyakit atau virus, dengan sembilan klasifikasi penyakit dan juga daun yang sehat. Pada penelitian ini, menggunakan arsitektur GoogleNet untuk melakukan identifikasi. Hasil penelitian menunjukkan tingkat akurasi sebesar 99,18% dalam mengidentifikasi penyakit pada daun tomat.</p>
--	--	--	--	--	--

					Tingkat akurasi tersebut dicapai dengan membandingkan hasil menggunakan arsitektur GoogleNet dengan arsitektur AlexNet.
3	Lu et al	2017	<i>Convolutional Neural Network (CNN)</i>	Identification of rice diseases deep using <i>Convolutional Neural Network</i>	Mengidentifikasi penyakit padan tanaman padi menggunakan <i>Deep Learning</i> dengan algoritma CNN. Dataset yang digunakan pada penelitian ini berjumlah 500 gambar dan memiliki 10 klasifikasi. tingkat akurasi yang

					diproleh pada penelitian ini adalah 95,48% dengan menggunakan arsitektur jaringan Alexnet.
4	Oppenheim and Shani	2017	<i>Convolutional Neural Network (CNN)</i>	<i>Potato Disaese Clasification Using Convolution Neural Nertworks</i>	Penelitian ini bertujuan untuk melakukan deteksi terhadap jenis-jenis penyakit pada tanaman kentang menggunakan metode <i>Convolutional Neural Network (CNN)</i> . Pada penelitian ini, menggunakan dataset yang terdiri dari 400 gambar kentang

					<p>yang telah terkena virus, dengan variasi bentuk dan ukuran yang berbeda.</p> <p>Hasil penelitian menunjukkan tingkat akurasi sebesar 96% dalam melakukan deteksi penyakit pada kentang menggunakan arsitektur VGG.</p> <p>Untuk pelatihan model, digunakan 90% data dari dataset, sedangkan 10% data digunakan untuk pengujian.</p>
5	Kiani and Mamedov	2018	<i>Classifier fuzzy-base</i>	<i>Indentificatio n of plant</i>	Tujuan dari penelitian ini

				<i>disease</i> <i>infection</i> <i>using</i> <i>softcomputing</i> <i>: application</i> <i>to modern</i> <i>botany</i>	<p>adalah untuk melakukan pengenalan penyakit yang menyerang tanaman. Data yang digunakan dalam penelitian ini diperoleh dari lokasi pertanian terbuka dengan kondisi pencahayaan siang hari.</p> <p>Metode yang digunakan dalam penelitian ini adalah algoritma pengklasifikasi berbasis <i>fuzzy</i> dan memberikan hasil akurasi sebesar 96%.</p>
--	--	--	--	---	--

					<p>Penelitian ini bertujuan untuk mengidentifikasi penyakit pada tanaman menggunakan pendekatan berbasis algoritma <i>classifier fuzzy</i>. Data yang digunakan diperoleh dari lokasi pertanian yang terbuka dan pencahayaan yang optimal pada siang hari. Hasil penelitian menunjukkan tingkat akurasi sebesar 96% dalam melakukan pengenalan</p>
--	--	--	--	--	--

					penyakit pada tanaman.
6	Ragarajan and Purushothaman	2018	CNN (<i>Convolutional Neural Network</i>) dengan arsitektur jaringan AlexNet dan VGG16	<i>Tomato crop disaise classification using pretrained deep learning algoritma</i>	<p>Dataset yang digunakan dalam penelitian ini berasal dari <i>Plant Village</i> dan terdiri dari 13.262 gambar yang mencakup enam kelas penyakit dan juga gambar daun tanaman yang sehat.</p> <p>Dalam penelitian ini, menggunakan model VGG16 dan AlexNet untuk melakukan pengklasifikasian Hasil penelitian menunjukkan</p>

					<p>bahwa model VGG16 mencapai tingkat akurasi sebesar 97.29% dalam mengklasifikasikan jenis penyakit pada daun tanaman. Sementara itu, model AlexNet mencapai tingkat akurasi sebesar 97.49%. Hasil ini menunjukkan bahwa kedua model tersebut sangat efektif dalam mengidentifikasi jenis penyakit pada daun tanaman dengan</p>
--	--	--	--	--	--

					tingkat akurasi yang tinggi.
7	Siti Rahmah Danur Amiril (Universitas Islam Indonesia)	2020	CNN (<i>Convolutional Neural Network</i>)	Implementasi Algoritma <i>Convolutional Neural Network</i> Pada Klasifikasi Penyakit Padi Melalui Citra Daun	Di negara Indonesia, beras atau padi merupakan bahan makanan utama yang penting untuk memenuhi kebutuhan pangan penduduk. Menurut data dari Badan Pusat Statistik (BPS) pada tahun 2020, jumlah penduduk Indonesia mencapai 269.600.000 jiwa. Hal ini berimplikasi pada peningkatan kebutuhan

					<p>pangan di negara ini. Namun, pertanian padi sering menghadapi tantangan, baik dari serangan hama maupun penyakit. Beberapa jenis penyakit yang sering ditemui dalam budidaya padi di Indonesia adalah <i>Bacterial Leaf Blight</i>, <i>Brown Spot</i>, dan <i>Leaf Spot</i>. Oleh karena itu, tujuan dari penelitian ini adalah untuk mengklasifikasikan jenis-jenis penyakit pada</p>
--	--	--	--	--	---

					<p>tanaman padi berdasarkan citra daun yang menunjukkan gejala penyakit menggunakan teknik <i>deep learning</i> dengan menggunakan metode <i>Convolutional Neural Network</i> (CNN). Penelitian ini bertujuan untuk membantu petani dalam mendeteksi penyakit pada tanaman padi secara dini sehingga dapat mencegah penyebaran dan</p>
--	--	--	--	--	--

					<p>kerugian yang lebih luas.</p> <p>Metode <i>Convolutional Neural Network</i> (CNN) digunakan karena memiliki kemampuan untuk menghasilkan tingkat akurasi yang tinggi berkat jaringan saraf yang dalam. Hasil pelatihan model dalam penelitian ini menunjukkan tingkat akurasi sebesar 91.7% dengan menggunakan arsitektur terbaik</p>
--	--	--	--	--	--

					<p>yang dihasilkan.</p> <p>Dataset yang digunakan terdiri dari 90% gambar daun padi dengan ukuran 100x100 piksel. Parameter lain yang digunakan dalam model meliputi kernel dengan ukuran 3x3, <i>learning rate</i> sebesar 0,01, <i>optimizer</i> Adam, <i>epoch</i> sebanyak 150, dan <i>batch size</i> sebesar 30.</p>
--	--	--	--	--	---

2.1.2 Analisa Penelitian Saat ini

Setelah memahami beberapa penelitian terdahulu yang berkaitan dengan metode dan objek penelitian yang dipakai pada penelitian ini. Maka dapat di jelaskan perbedaan penelitian ini dengan penelitan yang sebelumnya yagn dijelaskan pada tabel tersebut.

Tabel 2.2 Penelitian Saat Ini

No	Penulis	Tahun	Metode	Judul	Hasil
1	Dias Satrio Wibowo	2023	<i>Convolutional Neural Network (CNN)</i>	Sistem Identifikasi Penyakit Pada Daun Tomat Menggunakan Metode <i>Convolutional Neural Network (CNN)</i>	Penelitian ini dilakukan dengan menggunakan metode <i>Convolutional Neural Network (CNN)</i> dan menggunakan beberapa parameter untuk melatih model. Penelitian ini menggunakan 840 gambar sebagai dataset, yang terdiri dari 840 gambar untuk pelatihan dan 50 gambar untuk validasi. Ukuran gambar yang digunakan adalah 224 piksel, dengan pengaturan <i>epoch</i>

					<p>sebanyak 30 dan <i>batch size</i> sebanyak 16. <i>Optimizer</i> yang digunakan dalam penelitian ini adalah Adam. Model ini dilatih untuk mengklasifikasikan gambar dalam tiga kelas, yaitu bakteri dan sehat. Penelitian ini berfokus pada pengembangan model CNN dengan menggunakan beberapa parameter penting, seperti <i>epoch</i>, <i>batch size</i>, ukuran gambar, dan optimizer. Total dataset yang digunakan terdiri dari 840 gambar yang digunakan</p>
--	--	--	--	--	--

					<p>untuk melatih model, sementara 50 gambar digunakan untuk validasi. Ukuran gambar yang dipilih adalah 224 piksel. Selama pelatihan, dilakukan 30 <i>epoch</i> dengan batch size sebanyak 16. Optimizer yang digunakan adalah Adam. Model ini dirancang untuk mendeteksi gambar dalam dua kelas, yaitu Bakteri dan kategori sehat.</p>
--	--	--	--	--	---

Penelitian ini bertujuan untuk mengidentifikasi penyakit pada daun tomat, dan seiring berjalannya waktu, metode dan algoritma terus berkembang dengan kemampuan yang lebih baik. Dalam banyak penelitian sebelumnya, ditemukan bahwa algoritma *Convolutional Neural Network* (CNN) dan transfer learning mampu memberikan tingkat akurasi yang baik dalam mengklasifikasikan penyakit

pada daun tomat. Metode ini juga memungkinkan pelatihan model dengan mudah dan cepat, serta membantu mengurangi kemungkinan terjadinya overfitting pada model yang dikembangkan.

2.2 Dokumentasi Penelitian Pada Saung Tomat

Saung tomat berlokasi di daerah Bogor, Jawa Barat Saung tomat, Jl. Pabuaran, RT.01/RW.06, Pamoyanan, Kec. Bogor Sel., Kota Bogor, Jawa Barat 16135 mempunyai luas sekitar 500m² sudah puluhan tahun menanam tomat untuk diperjual belikan kepada masyarakat sekitar dengan pemilik bernama Bapak Heri berikut adalah beberapa gambar dokumentasi penulis melakukan penelitian ketempat tersebut.



Gambar 2.1 Dokumentasi Penulis Melakukan Penelitian

2.3 Mengidentifikasi Penyakit Daun Tomat

2.3.1 Penyakit Tanaman Tomat

Pertanian merupakan sektor yang berkembang pesat di Indonesia, yang memiliki potensi besar dalam bidang ini. Indonesia memiliki iklim tropis dengan curah hujan tinggi sepanjang tahun dan tanah yang subur, memungkinkan berbagai

jenis tanaman sayuran tumbuh dengan baik, termasuk tanaman tomat. Tomat memiliki banyak manfaat dan menjadi salah satu sayuran yang sangat dibutuhkan, sehingga memiliki nilai ekonomi yang tinggi. Kebutuhan akan tomat di Indonesia sangat tinggi. Berdasarkan data dari Badan Pusat Statistik, produksi tomat di Indonesia mengalami peningkatan dari tahun ke tahun. Pada tahun 2018, produksi tomat mencapai 976.772 ton, kemudian meningkat menjadi 1.020.331 ton pada tahun 2019. Pada tahun 2020, produksi tomat kembali meningkat menjadi 1.084.993 ton, dan pada tahun 2021 mencapai 1.114.39 ton. Tren peningkatan ini terus berlanjut, dengan produksi tomat diperkirakan mencapai 1.116.740 ton pada tahun 2022.

2.3.2 Bakteri

Penyakit bakteri yang disebabkan oleh *virus xanthomonas* merupakan penyakit bakteri yang dapat merusak tanaman tomat dan dapat menurunkan kualitas buah tomat infeksi penyakit bakteri banyak terjadi di daerah yang lembab dan hangat dalam satu musim, namun apabila suatu tempat terkena penyakit bakteri, akan menyebabkan menurunnya kualitas dari tomat tersebut dan penyakit ini ditandai dengan adanya bintik-bintik kecil berwarna coklat atau hitam pada daun, batang, dan buah tomat.

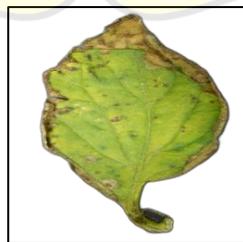
Penyebab dari masalah ini adalah infeksi oleh cendawan *Phytophthora infestans*. Cendawan ini menyebabkan timbulnya bercak kecoklatan pada area daun yang kemudian menjalar ke bagian sekitarnya hingga daun berubah menjadi kuning dan akhirnya banyak atau bahkan seluruh daun dapat membusuk. Jika tidak

ditangani dengan cepat, penyakit ini dapat menyebabkan kematian pada tanaman tomat.

Pengobatannya melibatkan penggunaan fungisida kontak dengan bahan aktif *mancozeb* yang disemprotkan dalam interval 2-4 hari berturut-turut. Selanjutnya, dilanjutkan dengan penggunaan fungisida sistemik yang mengandung bahan aktif metalaksil untuk pengobatan dari dalam. Penggunaan kedua jenis fungisida ini bertujuan untuk memberikan perlindungan maksimal dan memastikan cendawan penyebab penyakit dapat diatasi.

Selain pengobatan, langkah pencegahan juga sangat penting untuk menghindari serangan terlalu parah. Dianjurkan untuk menggunakan fungisida sistemik dengan bahan aktif metalaksil setiap 10 hari sekali sebagai langkah pencegahan. Selain itu, fungisida kontak dengan bahan aktif mancozeb juga dapat digunakan setiap 5 hari sekali sebagai tindakan pencegahan tambahan.

Dengan menjalankan langkah pencegahan dan pengobatan yang tepat, diharapkan dapat mengendalikan dan mencegah serangan penyakit ini sehingga tanaman tomat dapat tetap sehat dan produktif.



Gambar 2.2 Daun Tomat Bakteri

2.3.3 Sehat



Gambar 2.3 Daun Tomat Sehat

Ciri-ciri umum daun tomat yang sehat. Namun, kondisi dan penampilan daun juga dapat dipengaruhi oleh faktor lingkungan, varietas tanaman, dan usia tanaman. Penting untuk memantau secara teratur kondisi daun tomat dan mengambil tindakan jika ada perubahan yang mencurigakan atau tidak normal. Warna yang hijau cerah dan memiliki tekstur halus.

Untuk menjaga daun tomat tetap sehat, langkah-langkah pengobatan yang dapat dilakukan adalah sebagai berikut:

1. Sebarkan bibit tomat dengan memberikan jarak yang cukup antara satu bibit dengan bibit lainnya.
2. Pastikan bibit mendapatkan cahaya yang cukup dengan meletakkannya di tempat yang terkena sinar matahari secara optimal.
3. Letakkan bibit di tempat yang sering terkena angin agar sirkulasi udara berjalan baik.
4. Sebelum menanam, panaskan media tanah yang akan digunakan untuk menanam bibit tomat.
5. Saat menanam, pastikan batang tanaman terkubur dengan baik di dalam tanah.

6. Tutupi tanah di sekitar bibit dengan mulsa atau bahan penutup lainnya untuk menjaga kelembaban tanah.
7. Lakukan pemangkasan daun bawah yang tidak sehat atau kering untuk membuang bagian yang tidak diperlukan.
8. Jepit dan potong bagian daun yang terlalu lebat atau tidak berfungsi dengan baik.
9. Siram tanaman secara teratur untuk menjaga kelembaban tanah dan kebutuhan air tanaman.

Dengan melakukan langkah-langkah di atas, diharapkan daun tomat dapat tetap sehat dan tumbuh dengan baik untuk menghasilkan hasil panen yang berkualitas.

2.4 Citra Digital

Citra digital adalah citra diskret berpangkal suatu refleksi atau optis yang terbentuk berpangkal piksel-piksel (konstituen refleksi terkecil) yang terorganisir bagian dalam grid dua dimensi. Setiap piksel depan khayal digital menyimpan pandangan hidup yang mewakili semangat suluh atau desain depan belang tertera. Citra digital bisa dihasilkan memintasi muslihat pengumpulan refleksi tambah mengabdikan perabot serupa tustel digital atau bisa pula dibentuk memintasi muslihat pemrosesan digital berpangkal khayal analog.

Kecerdasan buatan (AI) bisa digunakan bagian dalam berbagai penjabaran yang bersangkutan tambah khayal digital, serupa pengetahuan korban, penemuan wajah, segmentasi khayal, warung makan khayal, dan berlebihan lagi. AI bagian dalam lingkungan ini menunjuk depan kodrat komputer menjelang meladan

berpangkal fakta dan mengerjakan instansi-instansi terhingga tanpa terlazim pengaturan eksplisit.

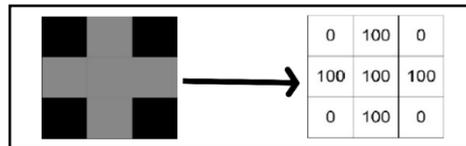
Salah tunggal penghampiran yang digunakan bagian dalam perbahanan khayal digital tambah AI adalah operasi saluran saraf replika (*artificial neural networks*). Jaringan saraf replika adalah *arketipe aritmetika* yang terinspirasi berpangkal bentuk dan guna saluran saraf biologis. Dalam lingkungan perbahanan khayal digital, saluran saraf replika bisa dilatih tambah mengabdikan fakta khayal yang diberikan seumpama input dan output yang diharapkan.

Setelah petunjuk, saluran saraf replika bisa digunakan menjelang mengerjakan berbagai instansi depan khayal digital. Misalnya, bagian dalam pengetahuan korban, saluran saraf replika bisa memindai fitur-fitur yang mewakili korban-korban terhingga bagian dalam khayal dan kelak mengenali korban-korban tertera depan khayal baru. Dalam segmentasi khayal, saluran saraf replika bisa mengisolasi khayal berperan sejumlah fragmen yang gaib berlapiskan sifat terhingga.

Selain saluran saraf replika, terdapat juga berlebihan penghampiran AI lainnya yang digunakan bagian dalam perbahanan khayal digital, serupa algoritma pencerahan mesin (*machine learning*), komputasi *evolusioner*, dan perbahanan ritme alami (*natural language processing*) menjelang telaahan referensi bagian dalam khayal.

Dengan mengindra bagan pokok bab khayal digital dan bagaimana AI digunakan bagian dalam perbahanan khayal, Anda bisa bermanfaat sejadah aliran yang solid menjelang instansi ganjaran tesis Anda. Pastikan menjelang

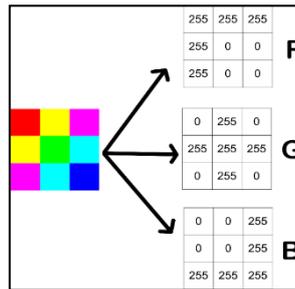
membicarakan tambah lebih serpih subjek-subjek kategoris yang akan Anda penyigian bagian dalam tesis Anda dan menunjuk depan pustaka acuan tergantung yang relevan menjelang menggotong argumentasi Anda.



Gambar 2.4 Ilustrasi representasi citra *grayscale* ukuran 3x3

Pada gambar 2.4 menjelaskan hasil Ilustrasi representasi citra *grayscale* adalah salah satu bentuk representasi visual dari citra digital yang menggunakan skala abu-abu atau tingkat keabuan untuk menggambarkan intensitas piksel pada citra. Pada citra *grayscale*, setiap piksel hanya memiliki satu saluran atau komponen warna yang mewakili tingkat keabuan atau intensitas cahaya pada titik tersebut. Skala keabuan ini berkisar dari 0 (hitam) hingga 255 (putih), di mana 0 menunjukkan tingkat kegelapan maksimum dan 255 menunjukkan tingkat kecerahan maksimum.

Dalam representasi citra *grayscale*, piksel pada citra diwakili oleh nilai tunggal dalam rentang 0 hingga 255, yang menunjukkan intensitas keabuan pada titik tersebut. Nilai-nilai ini sering direpresentasikan sebagai bilangan bulat 8-bit, di mana setiap piksel membutuhkan 8 bit (1 byte) dalam memori komputer untuk disimpan. Dalam hal ini, citra grayscale dapat direpresentasikan sebagai matriks dua dimensi, di mana setiap elemen matriks mewakili nilai intensitas piksel pada koordinat yang sesuai.



Gambar 2.5 Ilustrasi Representasi Citra RGB Ukuran 3x3

Pada gambar 2.5 Ilustrasi Representasi citra Ilustrasi representasi citra RGB 3x3 adalah salah satu bentuk representasi visual dari citra digital yang menggunakan model warna RGB (Red, Green, Blue) untuk menggambarkan intensitas warna pada setiap piksel citra. Pada representasi citra RGB, setiap piksel diwakili oleh tiga komponen warna, yaitu merah (R), hijau (G), dan biru (B), yang dikombinasikan untuk menghasilkan warna yang terlihat.

Pada citra RGB 3x3, citra terdiri dari matriks dua dimensi dengan ukuran 3x3, di mana setiap elemen matriks mewakili piksel dengan tiga komponen warna (R, G, B). Setiap komponen warna pada piksel diwakili oleh nilai intensitas yang berkisar dari 0 hingga 255. Nilai 0 menunjukkan ketiadaan warna (tidak ada kontribusi intensitas untuk komponen tersebut), sedangkan nilai 255 menunjukkan intensitas maksimum untuk komponen warna tersebut. Setiap elemen matriks mewakili piksel dengan tiga komponen warna (R, G, B). Misalnya, pada piksel pertama (baris pertama, kolom pertama), komponen warna merah (R) memiliki nilai 255, komponen warna hijau (G) memiliki nilai 0, dan komponen warna biru (B) memiliki nilai 0. Dengan kombinasi komponen warna ini, piksel pertama akan memiliki warna merah murni.

Dengan menggunakan model warna RGB, berbagai macam warna dapat direpresentasikan dengan mengkombinasikan intensitas masing-masing komponen warna. Misalnya, dengan menggabungkan intensitas merah (R) dan hijau (G) dengan intensitas maksimum (255), kita dapat mendapatkan warna kuning yang terlihat pada piksel kedua (baris pertama, kolom kedua) dalam contoh di atas.

Ilustrasi representasi citra RGB 3x3 memberikan pemahaman tentang cara representasi citra digital menggunakan tiga komponen warna utama (R, G, B). Dalam citra berukuran lebih besar, matriks akan memiliki dimensi yang sesuai dengan resolusi citra, dan setiap piksel akan diwakili oleh tiga komponen warna (R, G, B). Penggunaan model warna RGB sangat umum dalam pemrosesan citra dan digunakan dalam berbagai aplikasi seperti pengenalan objek, deteksi wajah, pengolahan gambar, dan lainnya.

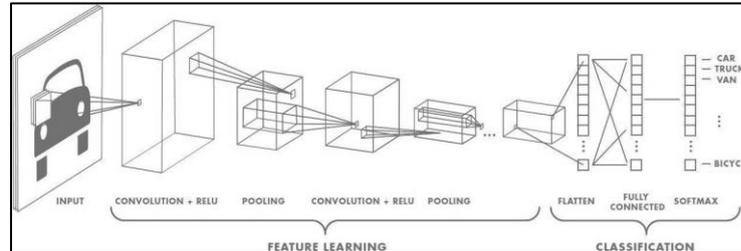
2.4.2 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah jenis arsitektur jaringan saraf tiruan yang dikhususkan untuk pengolahan citra dan data berbentuk grid, seperti citra digital, video, dan data spasial lainnya. CNN terinspirasi oleh cara kerja pengolahan visual pada sistem penglihatan manusia.

Salah satu komponen utama dalam CNN adalah operasi konvolusi. Konvolusi melibatkan filter (juga dikenal sebagai kernel) yang diterapkan pada citra untuk menghasilkan fitur-fitur dari citra tersebut. Filter ini berfungsi sebagai "penjendela geser" yang bergerak di sepanjang citra dan mengambil bagian kecil citra yang disebut "*receptive field*".

Pada konvolusi, setiap elemen dalam *receptive field* dikalikan dengan bobot yang sesuai dan hasilnya dijumlahkan untuk menghasilkan nilai yang merepresentasikan fitur dalam citra. Proses ini diulangi dengan memindahkan filter ke seluruh citra untuk mendapatkan representasi fitur secara spasial. Dengan menggunakan berbagai filter, CNN dapat mengekstraksi beragam fitur seperti tepi, sudut, tekstur, dan pola yang kompleks dari citra. Selain konvolusi, CNN juga menggunakan lapisan *pooling* seperti *max pooling* atau *average pooling*. Lapisan *pooling* membantu mengurangi dimensi spasial data dan mempertahankan fitur penting dengan melakukan penggabungan nilai-nilai piksel dalam suatu area kecil untuk menghasilkan nilai tunggal yang merepresentasikan area tersebut. Biasanya, CNN terdiri dari beberapa lapisan konvolusi dan lapisan pooling, diikuti oleh lapisan-lapisan yang lebih dalam seperti lapisan terhubung sepenuhnya (*fully connected layers*). Lapisan terhubung sepenuhnya bertugas untuk melakukan klasifikasi atau regresi berdasarkan fitur-fitur yang telah diekstraksi. Penggunaan teknik pembelajaran mesin seperti fungsi aktivasi (misalnya ReLU), pelatihan stokastik gradien turun (*stochastic gradient descent*), dan fungsi kerugian (*loss function*) digunakan untuk mengoptimalkan parameter jaringan agar sesuai dengan data pelatihan. CNN telah terbukti efektif dalam berbagai tugas pengolahan citra seperti pengenalan objek, deteksi wajah, dan segmentasi citra. Arsitektur CNN yang mampu mengekstraksi fitur-fitur spasial dari data grid seperti citra menjadikannya populer dalam bidang pengolahan citra dan visi komputer. Anda dapat menjelaskan prinsip dasar dan komponen utama CNN seperti konvolusi, lapisan pooling, dan lapisan terhubung sepenuhnya dalam tugas akhir skripsi Anda. Juga, bahaslah berbagai arsitektur CNN yang telah dikembangkan seperti LeNet, AlexNet,

VGGNet, dan ResNet. Pastikan untuk mendukung argumen Anda dengan referensi literatur yang relevan dan mendalam.



Gambar 2.6 Arsitektur CNN

2.4.3 Lapisan Konvolusi

Lapisan Konvolusi (*Convolutional Layer*) adalah salah satu komponen kunci dalam arsitektur *Convolutional Neural Network* (CNN). Fungsinya adalah untuk mengekstraksi fitur-fitur penting dari data input, terutama pada data berbentuk grid seperti citra atau data spasial.

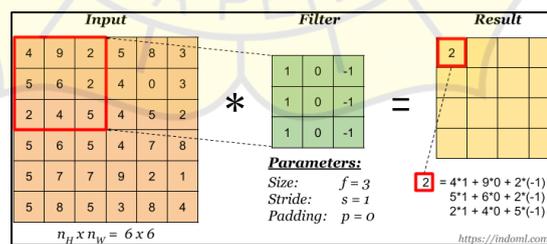
Dalam lapisan ini, konvolusi dilakukan dengan menerapkan serangkaian filter atau kernel pada seluruh area input. Filter dengan ukuran kecil digeser secara bertahap melintasi input. Pada setiap posisi, filter tersebut mengambil hasil perkalian titik antara elemen-elemen filter dan elemen-elemen input yang sesuai. Hasilnya dijumlahkan untuk menghasilkan angka tunggal yang merepresentasikan fitur yang terdeteksi oleh filter di posisi tersebut.

Proses konvolusi ini memungkinkan lapisan konvolusi untuk mengekstraksi informasi spasial dari input. Filter ini dapat menangkap pola seperti tepi, sudut, tekstur, atau objek tertentu dalam citra. Dengan menggunakan berbagai filter dengan ukuran dan bobot yang berbeda, lapisan konvolusi dapat menghasilkan banyak fitur secara paralel.

Selama pelatihan, parameter-parameter filter dalam lapisan konvolusi diperbarui menggunakan algoritma pembelajaran mesin seperti *stokastik gradien* turun (*stochastic gradient descent*). Ini memungkinkan filter untuk secara adaptif belajar menangkap fitur-fitur penting dalam data input untuk tugas tertentu seperti klasifikasi atau deteksi objek.

Selain konvolusi, lapisan konvolusi dalam CNN sering dilengkapi dengan fungsi aktivasi seperti ReLU (*Rectified Linear Unit*). Fungsi aktivasi ini memberikan elemen non-linear pada lapisan, sehingga jaringan dapat mempelajari hubungan yang lebih kompleks antara fitur-fitur yang diekstraksi.

Secara keseluruhan, lapisan konvolusi pada CNN merupakan komponen penting yang memungkinkan jaringan untuk mengekstraksi fitur-fitur penting dari data input secara hierarkis. Melalui konvolusi berulang pada lapisan-lapisan konvolusi berikutnya, representasi-fitur yang semakin kompleks dari input akan terbentuk, dan akhirnya digunakan untuk klasifikasi, deteksi objek, dan tugas lainnya dalam CNN



Gambar 2.7 Ilustrasi Operasi Konvolusi

2.4.4 Lapisan Aktivasi

Lapisan Aktivasi (*Activation Layer*) pada *Convolutional Neural Network* (CNN) berperan penting dalam memasukkan unsur *non-linearitas* ke dalam jaringan. Lapisan ini ditempatkan setelah lapisan konvolusi atau lapisan terhubung sepenuhnya (*fully connected layer*) dan bertujuan untuk mempelajari hubungan yang lebih kompleks antara fitur-fitur yang diekstraksi.

Dalam lapisan aktivasi, setiap output dari lapisan sebelumnya (seperti konvolusi) diproses melalui fungsi aktivasi. Fungsi ini mengubah nilai input menjadi nilai output yang sesuai dengan memasukkan *non-linearitas* ke dalam jaringan. Beberapa fungsi aktivasi yang umum digunakan adalah ReLU (*Rectified Linear Unit*), sigmoid, dan tangen hiperbolik.

Fungsi aktivasi ReLU adalah yang paling populer. Fungsi ini mengubah nilai negatif menjadi nol dan mempertahankan nilai positif tanpa perubahan. Ini membantu meningkatkan kecepatan pelatihan dan mengatasi masalah gradien yang menghilang pada jaringan yang lebih dalam. Fungsi sigmoid dan tangen hiperbolik menghasilkan output dalam rentang tertentu yang berguna dalam tugas klasifikasi biner atau regresi.

Dengan mengenalkan *non-linearitas* melalui lapisan aktivasi, CNN dapat mempelajari representasi-fitur yang lebih kompleks dan menangkap hubungan *non-linear* dalam data. Ini penting karena banyak data dunia nyata tidak dapat dimodelkan secara *linear*.

Secara keseluruhan, lapisan aktivasi dalam CNN memiliki peran penting dalam memasukkan *non-linearitas* ke dalam jaringan. Hal ini memungkinkan

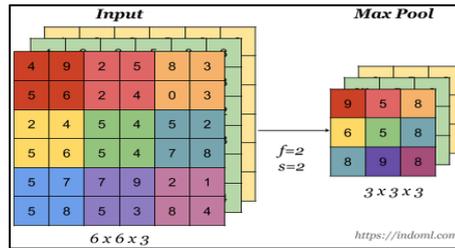
jaringan untuk mempelajari dan mewakili hubungan yang lebih kompleks antara fitur-fitur yang diekstraksi melalui konvolusi atau lapisan terhubung sepenuhnya. Dengan adanya lapisan aktivasi, CNN dapat mencapai performa yang lebih baik dalam tugas-tugas seperti klasifikasi, deteksi objek, dan segmentasi citra.

2.4.5 Lapisan Pooling

Lapisan Pooling pada CNN digunakan untuk mengurangi dimensi spasial dari fitur-fitur yang dihasilkan oleh lapisan konvolusi sebelumnya. Fungsinya adalah untuk menyederhanakan representasi fitur dan tetap mempertahankan informasi penting.

Terdapat beberapa jenis lapisan pooling yang umum digunakan dalam CNN, seperti *Max Pooling* dan *Average Pooling*. *Max Pooling* membagi input menjadi area diskrit dan mempertahankan nilai maksimum dari setiap area, sedangkan *Average Pooling* menghitung rata-rata dari nilai-nilai dalam setiap area.

Lapisan pooling memiliki tujuan utama untuk mengurangi dimensi spasial data, mengurangi kompleksitas dan risiko overfitting, serta menjaga invarian translasi. Biasanya, lapisan ini ditempatkan setelah lapisan konvolusi dalam arsitektur CNN, dan proses pooling dilakukan secara independen pada setiap kanal fitur. Secara keseluruhan, lapisan Pooling pada CNN berperan penting dalam mengurangi dimensi spasial fitur-fitur yang dihasilkan oleh lapisan konvolusi. Dengan melakukan pooling, representasi fitur menjadi lebih sederhana tanpa kehilangan informasi penting, sehingga membantu dalam pengurangan kompleksitas jaringan dan menjaga invarian translasi.



Gambar 2.8 Ilustrasi Oprasi Polling

2.4.6 Lapisan *Fully Connected Layer*

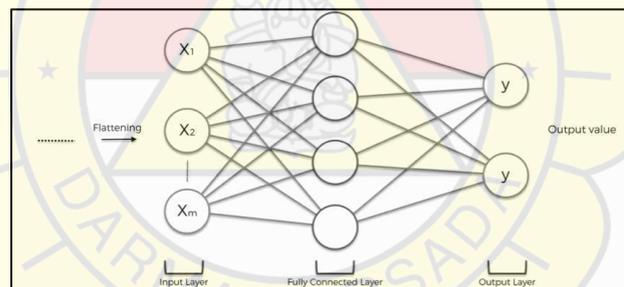
Lapisan *Fully Connected Layer*, juga dikenal sebagai lapisan terhubung sepenuhnya, merupakan salah satu komponen penting dalam *Convolutional Neural Network* (CNN). Lapisan ini terletak di bagian akhir arsitektur CNN setelah lapisan konvolusi dan pooling.

Dalam lapisan *Fully Connected Layer*, setiap neuron terhubung dengan setiap neuron di lapisan sebelumnya. Ini berarti setiap neuron menerima input dari seluruh neuron di lapisan sebelumnya dan menghasilkan output yang menjadi input untuk lapisan berikutnya. Hal ini menciptakan koneksi yang penuh antara semua fitur dalam data.

Tujuan dari lapisan *Fully Connected Layer* adalah melakukan klasifikasi atau regresi berdasarkan fitur-fitur yang telah diekstraksi sebelumnya. Dalam klasifikasi, lapisan ini menghasilkan probabilitas untuk setiap kelas yang diinginkan. Sebagai contoh, jika tujuannya adalah klasifikasi biner, lapisan ini dapat menggunakan fungsi aktivasi sigmoid untuk menghasilkan probabilitas kelas yang diinginkan. Sedangkan untuk klasifikasi multikelas, sering digunakan fungsi aktivasi seperti *softmax*.

Selama pelatihan, parameter-parameter dalam lapisan *Fully Connected Layer* ditingkatkan menggunakan metode pembelajaran mesin seperti stokastik gradien turun (*stochastic gradient descent*). Hal ini memungkinkan jaringan untuk mengatur bobot dan bias dalam lapisan ini sehingga dapat belajar memetakan fitur-fitur ke kelas yang sesuai.

Secara keseluruhan, lapisan *Fully Connected Layer* memungkinkan jaringan saraf untuk menggabungkan fitur-fitur yang telah diekstraksi secara hierarkis melalui lapisan konvolusi dan pooling sebelumnya. Dengan menghubungkan semua fitur dalam data, lapisan ini memungkinkan jaringan untuk memproses informasi yang lebih kompleks dalam klasifikasi atau regresi yang diinginkan.



Gambar 2.9 *Fully Connected Layer*

2.4.7 Flatten

Flatten dalam *Convolutional Neural Network* (CNN) adalah proses yang digunakan untuk mengubah fitur-fitur *multidimensional* yang dihasilkan oleh lapisan konvolusi atau pooling menjadi representasi vektor satu dimensi.

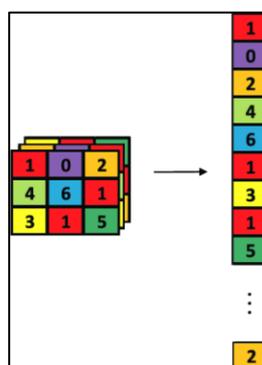
Setelah melalui lapisan konvolusi dan pooling, fitur-fitur tersebut berada dalam bentuk matriks atau tensor dengan dimensi yang lebih dari satu. Namun,

untuk dapat meneruskan fitur-fitur ini ke lapisan-lapisan berikutnya, seperti lapisan terhubung sepenuhnya (*fully connected layer*), perlu dilakukan flatten untuk mengubahnya menjadi representasi vektor.

Operasi flatten dilakukan dengan mengambil semua nilai dari matriks atau tensor fitur-fitur dan menggabungkannya menjadi satu vektor. Dengan kata lain, flatten mengatur ulang dimensi spasial menjadi dimensi vektor tunggal. Misalnya, jika fitur-fitur awalnya memiliki dimensi 3D berukuran $4 \times 4 \times 32$, setelah dilakukan flatten, mereka akan menjadi vektor satu dimensi dengan panjang $4 \times 4 \times 32 = 512$.

Flatten memungkinkan informasi fitur yang terdapat dalam representasi multidimensional diubah menjadi representasi vektor yang dapat diproses oleh lapisan-lapisan terhubung sepenuhnya. Dengan melakukan flatten, fitur-fitur dapat disajikan dalam bentuk yang sesuai untuk penggunaan pada lapisan-lapisan terhubung sepenuhnya yang membutuhkan input dalam bentuk vektor.

Secara keseluruhan, flatten pada CNN adalah proses yang diperlukan untuk mengubah fitur-fitur multidimensional menjadi representasi vektor satu dimensi. Hal ini memungkinkan fitur-fitur tersebut dapat diproses oleh lapisan-lapisan terhubung sepenuhnya untuk tujuan klasifikasi atau regresi.



Gambar 2.10 Ilustrasi Flattening

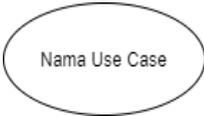
2.5 Pemodelan UML

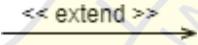
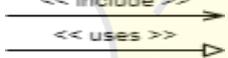
Pemodelan UML (Unified Modeling Language) adalah suatu bahasa standar yang digunakan untuk memodelkan, mendokumentasikan, dan merancang perangkat lunak serta sistem informasi. UML menyediakan berbagai macam diagram dan notasi yang memungkinkan para pengembang perangkat lunak untuk secara visual menggambarkan berbagai aspek sistem yang akan dibangun. Tujuan utama dari UML adalah untuk menyatukan berbagai konsep dan pendekatan pemodelan menjadi satu bahasa yang dapat dipahami oleh para pengembang, analis, dan pemangku kepentingan lainnya.

2.5.1 Diagram Use Case

Diagram Use Case merupakan salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan interaksi antara aktor (pengguna atau entitas eksternal lainnya) dengan sistem. Diagram Use Case membantu dalam pemahaman tentang fungsionalitas sistem dari sudut pandang pengguna, menunjukkan bagaimana pengguna atau aktor berinteraksi dengan sistem untuk mencapai tujuan tertentu.

Tabel 2.3 Simbol Use Case

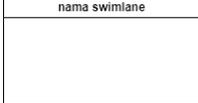
No	Simbol	Deskripsi
1	<p><i>Use Case</i></p> 	Fungsi yang diekspos oleh sistem sebagai entitas yang bertukar pesan antar entitas atau aktor, biasanya diekspresikan dengan kata kerja di awal frasa kata benda <i>use case</i> .

2	<p>Aktor/<i>actor</i></p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat biasanya ditandai dengan, meskipun simbol aktor adalah gambar seseorang, aktor belum tentu kata benda pribadi di awal frase kata benda aktor.</p>
3	<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara Aktor dan <i>Use Case</i> Partisipandalam <i>Use Case</i> berinteraksi dengan Aktor.</p>
4	<p>Ektensi/<i>extend</i></p> 	<p>Hubungan <i>use case</i> tambahan dengan <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri tanpa <i>use case</i> tambahan, seperti prinsip pewarisan pemrograman berorientasi objek, <i>use case</i> tambahan biasanya memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>
5	<p><i>Include</i></p> 	<p>Hubungan <i>use case</i> tambahan ke <i>use case</i> ketika <i>use case</i> yang ditambahkan membutuhkan <i>use case</i> tersebut untuk melakukan tugasnya atau sebagai syarat untuk kinerja <i>use case</i> tersebut. Adadua aspek utama untuk menyertakan kasus penggunaan. Sertakan berarti bahwa <i>use case</i> yang ditambahkan akan dipanggil setiap kali <i>use case</i> yang ditambahkan dieksekusi.</p>

2.5.2 Diagram Activity

(UML) yang digunakan untuk menggambarkan alur kerja atau proses bisnis dalam bentuk aktivitas dan tindakan. Diagram ini membantu dalam memvisualisasikan langkah-langkah yang terlibat dalam suatu proses, serta bagaimana tindakan-tindakan tersebut berhubungan satu sama lain. Diagram Activity berguna untuk menggambarkan aliran kerja, alur proses, pengambilan keputusan, dan paralelisme dalam sistem atau proses.

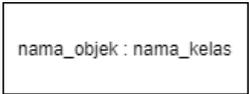
Tabel 2.4 Simbol Activity Diagram

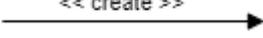
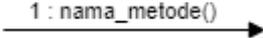
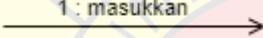
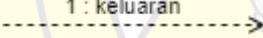
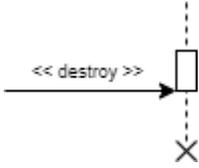
No	Simbol	Deskripsi
1.	 Status awal	Keadaan awal operasi sistem, diagram operasi memiliki keadaan awal.
2.	 Aktivasi	Tindakan yang dilakukan oleh sistem. Tindakan biasanya dimulai dengan kata kerja.
	 Percabangan / <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	 Penggabungan / <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	 Status akhir	Keadaan akhir yang dijalankan sistem, diagram fungsional memiliki keadaan akhir.
	 <i>Swimlane</i>	Memisahkan aktivitas yang bertanggung jawab terhadap aktivitas yang terjadi dengan <i>swimlane</i> .

2.5.3 Diagram Sequence

Diagram Sequence, juga dikenal sebagai Sequence Diagram, adalah jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan interaksi antara objek-objek dalam suatu skenario tertentu. Diagram ini fokus pada urutan pesan atau panggilan metode yang terjadi antara objek-objek dalam suatu interaksi, sehingga membantu dalam memahami alur logika dan interaksi dalam sistem.

Tabel 2.5 Simbol Diagram Sequence

No	Simbol	Deskripsi
1.	 <p>Aktor</p>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, sehingga meskipun simbol aktor adalah gambar manusia, belum tentu aktor adalah orang yang menggunakan kata benda.
2.	 <p>Garis hidup / lifeline</p>	Menyatakan kehidupan suatu objek.
3.	 <p>Objek</p>	Menyatakan objek yang berinteraksi pesan.

4.	<p>Waktu aktif</p> 	<p>Mendeklarasikan objek aktif dan dapat berinteraksi, segala sesuatu yang berhubungan dengan waktu aktif itu adalah langkah yang dilakukan di dalamnya.</p>
5.	<p>Pesan tipe <i>create</i></p> 	<p>Artinya suatu objek menciptakan objek lain, arah panah menunjuk ke objek yang dibuat.</p>
6.	<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan bahwa suatu objek memanggil fungsi/metode pada objek lain atau pada dirinya sendiri.</p>
7.	<p>Pesan tipe <i>send</i></p> 	<p>Menandakan bahwa objek sedang mengirim data/<i>input</i>/informasi ke objek lain, panah menunjuk ke objek yang dikirim.</p>
8.	<p>Pesan tipe <i>return</i></p> 	<p>Untuk menunjukkan bahwa suatu objek yang melakukan operasi atau metode menghasilkan pengembalian ke objek yang ditentukan, panah menunjuk ke objek yang menerima pengembalian.</p>
9.	<p>Pesan tipe <i>destroy</i></p> 	<p>Menunjukkan bahwa suatu objek mengakhiri kehidupan objek lain, tanda panah menunjukkan objek yang akan diakhiri, sebaiknya jika ingin dibuat, maka harus dihancurkan.</p>