

BAB II

KAJIAN LITERATUR

2.1 Citra Sintetik

Citra merupakan gambaran dua dimensi dari suatu objek yang berasal dari dunia visual dan digunakan dalam berbagai bidang ilmu, seperti seni, penglihatan manusia, astronomi, teknik, serta bidang lainnya. Citra tersusun atas kumpulan piksel atau titik-titik berwarna yang membentuk suatu tampilan dua dimensi [12]. Citra sintetik merupakan gambar yang dihasilkan oleh model kecerdasan buatan (AI), seperti *Generative Adversarial Networks* (GAN), dan *diffusion models*. Berbeda dengan citra nyata yang terbentuk secara alami melalui interaksi fisik antara cahaya, objek, dan sensor kamera, citra sintetik dibuat melalui proses komputasi yang melibatkan optimisasi matematis dan metode statistik yang kompleks [13]. Proses pembuatan citra sintetik dapat dilakukan melalui beberapa metode, seperti menciptakan gambar baru yang sebelumnya belum pernah ada, memanipulasi gambar asli, serta melakukan analisis dan pengenalan pola pada citra digital. Perkembangan teknologi generatif, seperti *Generative Adversarial Networks* (GAN), *Variational Autoencoders* (VAE), dan model difusi, juga telah memungkinkan terciptanya citra sintetik dengan tingkat kemiripan yang sangat tinggi terhadap citra asli. Karakteristik dari citra sintetik meliputi [14]:

1. Realisme Visual Tinggi: Citra sintetik memiliki tingkat kemiripan visual yang sangat kuat dengan citra asli, sehingga semakin sulit dibedakan oleh persepsi manusia.
2. Anomali Tekstur Mikro: Meskipun secara semantik tampak meyakinkan, proses AI generatif sering kali meninggalkan jejak anomali atau ketidakaturan pada tingkat tekstur mikro dan statistik piksel yang halus.
3. Ketidakkonsistenan Statistik: Citra sintetik sering kali gagal mereplikasi konsistensi statistik tekstur alami secara sempurna dibandingkan dengan citra yang dihasilkan oleh sistem pencitraan fisik.

2.2 Pengolahan Citra Sintetik

Dunia pengolahan citra sedang mengalami transformasi fundamental, bergeser dari manipulasi piksel tradisional menuju paradigma AI generatif. Transisi ini bukan sekadar evolusi teknis, melainkan perubahan strategis yang mendefinisikan ulang batas antara

keaktivitas manusia dan kapabilitas mesin. Bagi industri kreatif, pemasaran, dan teknologi, kemampuan untuk mensintesis visual kompleks secara instan merupakan aset strategis yang mengakselerasi siklus produksi dan inovasi. Berdasarkan konteks teknis saat ini, citra sintetik didefinisikan sebagai sub-bidang AI yang berfokus pada penghasilan visual melalui deskripsi tekstual (*text-to-image*). Evolusi bidang ini bermula dari metode awal seperti *Variational Autoencoders* (VAEs) yang mengutamakan kompresi data, diikuti oleh dominasi *Generative Adversarial Networks* (GANs) [15].

Generative Adversarial Networks (GAN) merupakan metode pembelajaran mesin tanpa pengawasan (*unsupervised learning*) yang digunakan untuk mempelajari pola dari data masukan secara otomatis. Metode ini mampu menghasilkan data baru yang memiliki karakteristik serupa dengan dataset asli. GAN dibangun menggunakan dua jaringan saraf yang bekerja secara bersamaan dalam proses kompetitif. Jaringan pertama disebut *generator* yang bertugas membuat citra sintetik baru dengan tujuan menghasilkan data yang menyerupai data asli. Jaringan kedua disebut *discriminator* yang berfungsi membedakan apakah citra yang diterima berasal dari dataset nyata atau merupakan hasil buatan *generator* [16].

Dalam proses pengolahan citra sintetik, pembuatan gambar dilakukan dengan memanfaatkan algoritma berbasis *Artificial Neural Network* (ANN) yang dirancang untuk mempelajari pola dan karakteristik dari data visual. Teknologi ini memungkinkan sistem melakukan manipulasi terhadap berbagai komponen digital, seperti wajah, latar belakang, ekspresi wajah, sehingga menghasilkan konten yang tampak realistis. Tingkat realisme yang tinggi tersebut menyebabkan citra sintetik sering kali sulit dibedakan dari media asli karena mampu meniru karakteristik visual secara sangat mendetail. Pengolahan citra sintetik umumnya memanfaatkan pendekatan berbasis *deep learning*, seperti *Convolutional Neural Networks* (CNN) dan arsitektur *autoencoder*, termasuk *Variational Autoencoder* (VAE). Metode-metode tersebut bekerja dengan mempelajari pola kompleks pada data visual untuk menghasilkan citra baru atau memodifikasi citra yang telah ada. Selain itu, dalam konteks pengolahan data, citra sintetik juga dapat dihasilkan melalui teknik *data augmentation*, yaitu proses menambah variasi data dari dataset yang telah tersedia. Pendekatan ini bertujuan memperkaya keragaman dataset sehingga dapat meningkatkan kualitas pelatihan model serta memperbaiki kemampuan sistem dalam melakukan pengenalan dan klasifikasi citra [17].

Citra sintetik memiliki potensi yang besar dalam mendukung perkembangan industri kreatif melalui berbagai bentuk inovasi digital. Salah satu penerapan nyatanya dapat

ditemukan pada aplikasi yang memungkinkan pengguna melakukan pertukaran wajah (*face swap*) dengan hasil yang tampak alami. Dalam bidang penelitian dan pengembangan teknologi, citra sintetis dimanfaatkan untuk mendukung berbagai tujuan teknis dan perlindungan digital. Salah satunya adalah dalam pelatihan model deteksi, di mana metode GAN digunakan untuk menghasilkan citra palsu yang realistis sebagai data pelatihan bagi sistem kecerdasan buatan. Pendekatan ini bertujuan meningkatkan kemampuan sistem agar menjadi lebih adaptif dan akurat dalam mendeteksi konten manipulatif di masa mendatang. Selain itu, kajian terhadap citra sintetis juga berperan penting dalam bidang keamanan digital dan forensik untuk membantu mengidentifikasi serta menanggulangi penyebaran konten manipulatif di berbagai platform media. Di sisi lain, penggunaan citra sintetis juga memiliki potensi penyalahgunaan yang dapat menimbulkan dampak negatif. Teknologi ini dapat dimanfaatkan untuk menyebarkan informasi palsu atau berita bohong yang berpotensi menyesatkan masyarakat. Selain itu, citra sintetis dapat digunakan dalam pembuatan konten tanpa persetujuan pihak terkait, seperti manipulasi wajah seseorang dalam konten tertentu yang dapat merusak privasi, reputasi, dan keamanan digital individu. Bentuk penyalahgunaan lainnya adalah praktik penipuan, yaitu menampilkan seseorang seolah-olah melakukan atau mengatakan sesuatu yang sebenarnya tidak pernah terjadi dengan tujuan manipulasi [18].

2.3 Artificial Intelligence (AI)

Artificial Intelligence (AI) secara umum didefinisikan sebagai kumpulan sistem atau program komputer yang dirancang untuk meniru kecerdasan, perilaku, serta kemampuan kognitif manusia. Teknologi ini dikembangkan agar mampu menjalankan berbagai tugas kompleks yang umumnya berkaitan dengan kemampuan manusia, seperti berpikir, bernalar, belajar dari pengalaman, dan menyelesaikan masalah. Salah satu karakteristik utama AI adalah kemampuannya dalam melakukan pengambilan keputusan secara mandiri. Sistem AI dapat menganalisis data, mengenali pola tertentu, serta menghasilkan keputusan tanpa memerlukan intervensi manusia secara langsung pada setiap prosesnya. Selain itu, AI juga memiliki kemampuan untuk belajar dan beradaptasi melalui pemanfaatan algoritma serta daya komputasi yang tinggi. Melalui proses pembelajaran dari data yang tersedia, sistem dapat menyesuaikan diri terhadap informasi maupun kondisi baru sehingga kinerjanya dapat terus meningkat seiring waktu [19].

Perkembangan AI dimulai dari fondasi teoritis yang kemudian berkembang menjadi berbagai pendekatan dan teknologi modern yang digunakan saat ini. Fondasi awal AI secara teoretis sering dikaitkan dengan seorang matematikawan Inggris, Alan Turing. Pada tahun 1950, ia menerbitkan makalah berjudul *Computing Machinery and Intelligence* yang membahas pertanyaan apakah mesin dapat menunjukkan tingkat kecerdasan yang setara dengan manusia. Dalam makalah tersebut, Turing juga memperkenalkan "*Tes Turing*", yaitu metode untuk mengukur kemampuan mesin dalam meniru perilaku manusia sampai seorang penilai tidak mampu membedakan apakah respons yang diterima berasal dari manusia atau mesin. Perkembangan berikutnya terjadi pada tahun 1956 melalui Konferensi Dartmouth yang diselenggarakan di Amerika Serikat. Peristiwa ini menjadi tonggak sejarah karena untuk pertama kalinya istilah "Kecerdasan Buatan" atau *Artificial Intelligence* digunakan secara resmi. Sejumlah ilmuwan pionir seperti John McCarthy, Marvin Minsky, Allen Newell, dan Herbert Simon memulai penelitian secara intensif di bidang ini. Pada masa tersebut, pendekatan yang digunakan adalah pendekatan simbolik, yaitu pandangan bahwa kecerdasan dapat dibentuk melalui manipulasi simbol-simbol yang direpresentasikan dalam sistem komputer [20].

Setelah AI diakui secara resmi sebagai bidang penelitian, perkembangannya mulai bercabang ke berbagai teknik dan metodologi penting. Pada tahun 1958, Frank Rosenblatt memperkenalkan model Perceptron sebagai dasar dari jaringan saraf tiruan yang terinspirasi oleh cara kerja otak manusia. Selanjutnya, pada tahun 1965, Lotfi A. Zadeh memperkenalkan logika fuzzy untuk menangani masalah ketidakpastian yang tidak dapat dijelaskan secara efektif oleh logika klasik yang hanya mengenal nilai benar atau salah. Kemudian pada tahun 1975, John Holland mengembangkan algoritma genetika dengan meniru proses evolusi biologis untuk menyelesaikan berbagai persoalan komputasi. Selain itu, sistem pakar (*expert systems*) juga dikembangkan untuk meniru kemampuan seorang ahli manusia dalam mengambil keputusan melalui penggunaan basis pengetahuan dan aturan yang telah ditetapkan secara eksplisit [20].

Pada beberapa dekade awal perkembangannya, AI lebih berfokus pada pengembangan sistem yang menyelesaikan masalah melalui aturan dan pengetahuan yang diprogram secara langsung. Dalam sejarah *Natural Language Processing* (NLP) atau Pemrosesan Bahasa Alami, terdapat perubahan pendekatan yang cukup signifikan. Pada era *rule-based* yang berlangsung sekitar tahun 1950-an hingga 1980-an, sistem bekerja menggunakan aturan tata bahasa dan sintaksis yang disusun secara manual oleh para ahli

bahasa. Namun, pada periode 1990-an hingga 2010-an, meningkatnya jumlah data mendorong munculnya pendekatan statistik dan *machine learning*, di mana sistem mulai menggunakan model statistik seperti *Hidden Markov Models* (HMMs) dan teknik pembelajaran mesin. Pada era modern, AI telah berkembang dari sekadar konsep dalam fiksi ilmiah menjadi teknologi nyata yang digunakan di berbagai bidang kehidupan. Sistem AI masa kini tidak lagi hanya bergantung pada aturan yang diprogram secara kaku, tetapi memanfaatkan *Machine Learning*, yaitu metode yang memungkinkan mesin belajar dari data untuk mengenali pola dan mengambil keputusan secara mandiri. Implementasi AI saat ini dapat ditemukan dalam berbagai teknologi seperti asisten virtual seperti Siri, kendaraan otonom, hingga sistem diagnosis medis yang canggih. AI telah mengalami evolusi yang sangat besar, mulai dari manipulasi simbol sederhana pada tahun 1950-an hingga menjadi sistem pembelajaran berbasis data besar (*Big Data*) yang sangat kompleks pada abad ke-21. AI memiliki beberapa cabang yang saling terhubung untuk meningkatkan kemampuan mesin dalam memahami serta merespons lingkungan sekitarnya. Berikut merupakan cabangcabang AI [20]:

1. *Machine Learning*: Merupakan sub-bidang AI yang berfokus pada pengembangan algoritma sehingga komputer mampu mempelajari data dan menghasilkan prediksi maupun keputusan tanpa harus diprogram secara langsung.
2. *Natural Language Processing* (NLP): Cabang AI yang mempelajari interaksi antara manusia dan komputer melalui bahasa alami, sehingga mesin dapat memahami, mengolah, menganalisis, serta menghasilkan teks maupun ucapan.
3. *Computer Vision*: Bidang yang memusatkan perhatian pada kemampuan komputer dalam memperoleh, memproses, serta menganalisis informasi visual, seperti gambar dan video, dengan cara yang menyerupai sistem penglihatan manusia.
4. Robotika: Cabang AI yang mengintegrasikan teknologi kecerdasan buatan untuk menciptakan mesin yang mampu melakukan berbagai tugas kompleks dan berinteraksi secara mandiri dengan lingkungan fisik.
5. *Expert Systems*: Sistem yang dikembangkan untuk meniru kemampuan seorang ahli dalam mengambil keputusan pada bidang tertentu dengan memanfaatkan basis pengetahuan dan aturan yang telah ditentukan.
6. *Artificial Neural Network* (ANN): Model komputasi yang terinspirasi dari mekanisme kerja otak manusia, yang terdiri atas sejumlah lapisan neuron buatan untuk mengolah data yang kompleks.

7. *Fuzzy Logic*: Pendekatan yang dikembangkan untuk menangani ketidakpastian yang tidak dapat diselesaikan dengan logika klasik benar atau salah, sehingga dapat merepresentasikan nilai di antara keduanya.
8. *Algoritma Genetika*: Metode penyelesaian masalah yang mengadopsi konsep evolusi biologis, seperti seleksi alam, mutasi, dan *cross-over*, guna memperoleh solusi yang paling optimal.
9. *Knowledge Representation*: Cabang yang berfokus pada penyajian fakta dan informasi dalam bentuk simbolik agar komputer dapat mengolah pengetahuan serta menarik kesimpulan dengan lebih efektif.
10. *Search Algorithms*: Komponen dasar dalam AI yang digunakan untuk membantu sistem menemukan solusi terbaik melalui proses eksplorasi berbagai kemungkinan dalam ruang pencarian yang kompleks.
11. *Intelligent Agent*: Program atau sistem yang memiliki kemampuan memahami lingkungan, mengambil keputusan, dan melakukan tindakan secara mandiri untuk mencapai tujuan tertentu.

2.4 Generative AI

Generative AI merupakan cabang dari AI yang berfokus pada kemampuan sistem untuk menghasilkan konten baru, bukan hanya menganalisis, mengklasifikasi, atau memproses data yang telah tersedia [21]. Istilah “*generative*” mengacu pada kemampuan teknologi ini dalam menciptakan atau memproduksi sesuatu yang baru. Konten yang dihasilkan oleh *Generative AI* mencakup berbagai bentuk, seperti teks, gambar, musik, audio, video, kode pemrograman, hingga simulasi digital. Secara umum, *Generative AI* bekerja dengan memanfaatkan algoritma *deep learning* yang kompleks, khususnya model *transformer*, sehingga sistem mampu mempelajari pola dari kumpulan data dalam jumlah besar melalui pendekatan probabilistik dan pemahaman konteks. Dengan kemampuan tersebut, *Generative AI* dapat mengenali pola-pola tersembunyi dalam data pelatihan dan menghasilkan output baru yang tetap memiliki karakteristik serupa dengan data aslinya [22].

Berbeda dengan AI tradisional yang umumnya dirancang untuk tugas tertentu, seperti klasifikasi atau prediksi, *Generative AI* memiliki tingkat fleksibilitas yang lebih tinggi dalam berbagai bidang. Teknologi ini tidak hanya berfungsi untuk mengidentifikasi informasi, tetapi juga mampu mentransformasikan input menjadi bentuk konten baru yang bersifat

kreatif dan dinamis. Dalam pengembangannya, terdapat beberapa model arsitektur yang mendukung kinerja *Generative AI* sebagai berikut:

1. *Generative Adversarial Networks* (GAN): Sebagai salah satu terobosan besar dalam pembelajaran generatif yang diperkenalkan pada tahun 2014. Model tersebut mengadopsi struktur dua komponen utama: jaringan *generator* dan jaringan *discriminator*. *Generator* berfungsi mensintesis data artifisial dari input stokastik (*noise*), sementara *discriminator* melakukan evaluasi komparatif antara data autentik dan sintetis. Proses klasifikasi pada *discriminator* diimplementasikan melalui fungsi aktivasi sigmoid dengan optimasi berbasis *binary cross-entropy loss*. Mengingat *generator* tidak memiliki akses langsung ke citra asli, pembelajaran dilakukan secara iteratif melalui umpan balik dari *discriminator*. Seluruh sistem dioptimalkan menggunakan metode *backpropagation* hingga konvergensi tercapai, yakni saat perbedaan antara data asli dan palsu menjadi minimal [23].
2. *Variational Auto-Encoder* (VAE): Diperkenalkan setahun sebelumnya pada tahun 2013, VAE berfokus pada pembelajaran distribusi probabilitas dari data tanpa label. Sistem ini bekerja dengan mengodekan (*encoding*) data masukan ke dalam ruang laten (*latent space*) berdimensi lebih rendah, kemudian mendekodekan (*decoding*) kembali ke wujud aslinya melalui metode pengambilan sampel agar representasi latennya mengikuti distribusi probabilitas standar. Karena fungsi densitas matematisnya tidak dapat dihitung secara langsung (*intractable*), VAE mengandalkan pendekatan inferensi variasional dan variabel laten untuk mempelajari distribusi data tersebut [24].
3. Model Difusi: Saat ini, model difusi diakui sebagai standar mutakhir (*state-of-the-art*) untuk menghasilkan citra yang beragam dan berkualitas tinggi. Berbeda dengan arsitektur sebelumnya, model ini membangkitkan citra melalui tahapan *denoising* (penghilangan derau). Prosesnya diawali dengan proses maju (*forward process*) yang secara berulang menambahkan derau *gaussian* pada citra masukan, dilanjutkan dengan proses balik (*reverse process/sampling*) untuk membuang derau tersebut demi merekonstruksi gambar baru. Sebagai inovasi lebih lanjut, diciptakanlah *Latent Diffusion Model* (LDM). Jika model difusi standar beroperasi langsung pada ruang piksel yang berdimensi tinggi, LDM bekerja jauh lebih efisien dengan memproyeksikan citra masukan ke dalam ruang laten berdimensi rendah sebelum mengaplikasikan algoritma difusi [24].

Jenis-jenis hasil yang dihasilkan oleh Generative AI dalam proses pembuatan dapat dikelompokkan ke dalam beberapa kategori utama. Setiap kategori memiliki fungsi dan karakteristik teknis yang berbeda pada tahapan produksi, yaitu sebagai berikut [25]:

1. Hasil Berbasis Teks (*Text-Based*)

Pada tahap awal atau tahap konsep, AI dimanfaatkan untuk menghasilkan gagasan dan menyusun struktur cerita. Hasil yang diperoleh mencakup ide cerita dan sinopsis yang berfungsi sebagai referensi tambahan dalam menyusun konflik, klimaks, penyelesaian, hingga bagian penutup cerita. Selain itu, AI juga dapat menghasilkan naskah *screenplay* dan *storyboard* berupa narasi detail setiap adegan yang nantinya digunakan sebagai pedoman dalam proses visualisasi. Beberapa platform yang dapat digunakan pada tahap ini antara lain ChatGPT, GrokAI, dan DeepSeek.

2. Hasil Gambar dan Sketsa (*Text-to-Image*)

Pada tahap desain, AI digunakan untuk mengubah perintah berbasis teks menjadi bentuk visual statis. Hasil yang dihasilkan berupa gambar sketsa yang menampilkan visualisasi awal suatu adegan atau potongan cerita dalam bentuk sketsa sederhana maupun hitam-putih. Sketsa ini digunakan untuk memastikan konsistensi konsep sebelum masuk ke tahap berikutnya. Selain itu, AI juga dapat menghasilkan visualisasi karakter dan latar yang menggambarkan tokoh maupun lingkungan cerita berdasarkan deskripsi naratif yang telah dibuat sebelumnya.

3. Hasil Visual Bergaya (*Image-to-Image*)

Tahap ini bertujuan menghasilkan visual yang lebih matang dengan gaya artistik tertentu. AI dapat mengubah sketsa awal menjadi ilustrasi kartun tiga dimensi yang lebih detail dengan tambahan warna, pencahayaan, dan efek visual yang lebih realistis. Dalam penelitian yang dijadikan acuan, hasil gambar yang dihasilkan memiliki spesifikasi teknis berupa resolusi 1536×1024 piksel, kedalaman bit 96 dpi, serta menggunakan format file *.PNG*. Salah satu platform yang digunakan dalam proses ini adalah MidjourneyAI.

4. Hasil Gerakan Animasi (*Image-to-Video* dan *Text-to-Video*)

Pada tahap ini, AI mampu mengubah gambar statis menjadi rangkaian adegan bergerak. Hasil yang dihasilkan berupa video animasi pendek yang menampilkan pergerakan karakter, perpindahan kamera visual, serta pengaturan pencahayaan yang disesuaikan dengan perintah atau *prompt* yang diberikan. Secara teknis, video yang dihasilkan memiliki resolusi 1152×768 piksel, bitrate sebesar 971 kbps, *frame rate* 30 fps, dan menggunakan format file *.Mp4*. Hasil gerakan tersebut juga memenuhi sedikitnya dua dari dua belas prinsip animasi,

seperti *Staging*, *Timing*, atau *Squash and Stretch*, sehingga gerakan yang dihasilkan memiliki makna dan struktur, bukan hanya gerakan acak.

5. Hasil Audio (*Text-to-Audio*)

AI juga berperan dalam menghasilkan berbagai elemen suara yang dibutuhkan dalam multimedia dan animasi. Hasil yang dihasilkan meliputi suara dialog antar karakter, musik latar (*background*), narasi (*voice over*), serta efek suara (*sound effect* atau SFX) yang berfungsi untuk memperkuat suasana dan mendukung jalannya cerita animasi.

Penerapan Generative AI dilakukan melalui integrasi yang terstruktur antara teknologi kecerdasan buatan dan kreativitas manusia. Tujuan penerapan ini adalah meningkatkan efisiensi dan produktivitas kerja tanpa menghilangkan peran manusia sebagai pengarah utama dalam pengembangan konsep. Berikut penjelasan penerapan *Generative AI* berdasarkan beberapa aspek [26]:

1. Tahapan Alur Kerja (*Workflow*)

Penerapan AI dalam proses desain dilakukan melalui tahapan yang sistematis agar hasil yang diperoleh sesuai dengan kebutuhan pengguna. Tahapan pertama adalah identifikasi kebutuhan, yaitu proses memahami *brief* atau kebutuhan klien untuk menentukan visi visual yang ingin dicapai. Selanjutnya, desainer menyusun *prompt* atau instruksi yang bersifat deskriptif dengan mencantumkan elemen-elemen penting seperti warna, gaya visual, serta komposisi desain. Ketepatan dalam menyusun *prompt* menjadi faktor yang sangat penting karena kualitas hasil yang dihasilkan AI sangat dipengaruhi oleh kualitas masukan yang diberikan. Setelah itu, *prompt* dimasukkan ke platform AI seperti Leonardo AI atau Midjourney untuk menghasilkan konsep awal secara cepat. Hasil yang diperoleh kemudian melalui tahap seleksi dan revisi manual agar aspek estetika serta detail desain tetap sesuai kebutuhan. Tahap terakhir adalah finalisasi, yaitu evaluasi kembali berdasarkan masukan dari klien sebelum hasil akhir dikirimkan.

2. Platform dan Alat yang Digunakan

Berbagai platform digunakan untuk mendukung implementasi *Generative AI* dalam proses desain. Midjourney banyak dimanfaatkan untuk menghasilkan ilustrasi digital yang lebih rinci maupun mengubah gambar potret menjadi karya seni dengan gaya tertentu. Selain itu, Canva AI dan Shutterstock AI digunakan untuk mempercepat pembuatan berbagai kebutuhan desain konten umum. Platform lain seperti Hotpot.ai

dimanfaatkan untuk mengubah deskripsi teks menjadi visual menarik sehingga mempermudah proses penerjemahan ide menjadi bentuk gambar.

3. Bidang Implementasi Praktis

Penerapan *Generative AI* telah digunakan dalam berbagai aktivitas operasional dan proses kreatif. Teknologi ini membantu mengotomatisasi tugas-tugas teknis yang bersifat berulang, seperti menghapus latar belakang gambar, memisahkan elemen visual, melakukan koreksi warna, serta meningkatkan kualitas dan resolusi gambar (*upscaling*) secara otomatis. Selain itu, AI juga mendukung eksperimen kreatif melalui pembuatan *moodboard* otomatis dan penyusunan desain awal yang cepat untuk kebutuhan presentasi atau *pitching* kepada klien. Pada bidang visualisasi karya sastra, AI dapat mengubah teks seperti narasi atau puisi menjadi visual yang menarik, sehingga menciptakan peluang baru dalam pengembangan *storytelling* berbasis visual.

Keberhasilan penerapan *Generative AI* dapat diukur melalui beberapa indikator penting. Dari segi waktu, pekerjaan yang sebelumnya memerlukan proses berjam-jam kini dapat diselesaikan dengan lebih cepat, bahkan hanya dalam beberapa menit. Selain meningkatkan kecepatan kerja, AI juga mampu menjaga kualitas hasil secara konsisten melalui penggunaan parameter algoritma yang stabil. Penerapan teknologi ini turut meningkatkan efisiensi penggunaan sumber daya dengan mengurangi beban pekerjaan yang dilakukan secara manual. Dengan demikian, pengguna dapat memusatkan perhatian pada pengembangan ide kreatif dan penyusunan strategi visual yang lebih kompleks.

Meskipun menawarkan berbagai manfaat, penerapan *Generative AI* tetap menghadapi beberapa tantangan yang perlu diperhatikan. Salah satu tantangan yang cukup penting berkaitan dengan aspek etika dan orisinalitas karya, karena terdapat risiko terjadinya plagiarisme serta berkurangnya nilai keaslian akibat dominasi hasil yang dihasilkan algoritma. Selain itu, penggunaan AI juga memunculkan perdebatan mengenai hak cipta dan kepemilikan terhadap karya yang dibuat oleh sistem kecerdasan buatan. Tantangan lain yang muncul adalah adanya kesenjangan keterampilan dalam penggunaan teknologi tersebut. Keberhasilan pemanfaatan AI sangat dipengaruhi oleh kemampuan pengguna atau desainer dalam memahami teknologi, khususnya pada aspek *prompt engineering* [26].

Interaksi antara manusia dan *Generative AI* umumnya dilakukan melalui prompt atau instruksi berbasis teks. Mekanisme ini memungkinkan terciptanya kolaborasi kreatif antara manusia dan AI dalam menghasilkan ide maupun konten baru, yaitu perpaduan antara

keaktivitas manusia dengan kemampuan eksplorasi data oleh sistem AI. Meskipun demikian, Generative AI tetap memiliki keterbatasan. Teknologi ini pada dasarnya hanya mensimulasikan pola berpikir manusia melalui pendekatan matematis sehingga tidak memiliki kesadaran, emosi, maupun imajinasi yang sebenarnya. Selain itu, Generative AI juga berpotensi mengalami halusinasi, yaitu kondisi ketika sistem menghasilkan informasi yang tampak meyakinkan tetapi tidak sesuai dengan fakta atau tidak memiliki dasar data yang valid. Secara keseluruhan, Generative AI merepresentasikan perkembangan penting dalam bidang kecerdasan buatan karena mengubah peran AI dari sekadar alat bantu pasif menjadi mitra kolaboratif yang mampu mendukung dan meningkatkan produktivitas kreatif manusia. Beberapa contoh generative AI modern meliputi Imagen, DALL-E 3, Midjourney, dan Grok-2 [27].

1. Imagen

Tahun 2022 menyaksikan kemajuan signifikan dalam teknologi pembuatan *text-to-image*. Google memperkenalkan Imagen, sebuah model berbasis difusi yang menghasilkan gambar berkualitas tinggi dari perintah tekstual. Proses pembuatan gambar oleh Imagen menggunakan sistem bertahap. Tahap dimulai dengan memproses teks (*Teks Encoder*) saat memasukkan *teks prompt*, Imagen menggunakan *text encoder* besar bersama T5-XXL yang statusnya dibekukan. Tugas *encoder* adalah memetakan dan menerjemahkan teks input tersebut menjadi serangkaian *embedding*, yakni representasi data yang dapat dipahami oleh mesin. Selanjutnya, tahap pembuatan gambar awal (*Base Diffusion Model*) di mana *text embedding* tersebut dimasukkan ke dalam model difusi dasar berbasis arsitektur U-Net. Model ini bertugas mengubah titik-titik acak menjadi gambar awal yang masih beresolusi sangat rendah, yaitu 64x64 piksel, dengan menyesuaikan bentuknya agar sesuai dengan instruksi teks. Untuk membuat gambar menjadi tajam dan beresolusi tinggi, Imagen kemudian meningkatkan resolusi secara bertahap (*Super-Resolution Diffusion Models*) menggunakan dua model difusi tambahan secara berurutan. Tahap pertama memperbesar gambar dari resolusi 64x64 menjadi 256x256 piksel, lalu tahap kedua memperbesar gambar dari 256x256 menjadi resolusi akhir 1024x1024 piksel. Keseluruhan proses ini menggunakan arsitektur baru yang disebut Efficient U-Net, yang dirancang agar lebih cepat dan efisien dalam penggunaan memori.

Selain alur utama tersebut, terdapat teknik khusus yang digunakan untuk memastikan gambar yang dihasilkan berkualitas tinggi dan tidak rusak. Imagen mengandalkan beberapa teknik penting, di antaranya adalah *Classifier-free Guidance* yang memastikan gambar

sangat patuh dan selaras dengan deskripsi teks pengguna. Kemudian terdapat *Dynamic Thresholding*, sebuah teknik untuk mencegah warna gambar menjadi terlalu *oversaturated* atau tidak natural saat sistem dipaksa patuh pada teks, sehingga gambar tetap terlihat realistis dan detail. Terakhir, diterapkan pula *Noise Conditioning Augmentation* pada model penambah resolusi untuk membuatnya lebih kuat terhadap cacat visual yang mungkin dihasilkan oleh model beresolusi lebih rendah sebelumnya [28].



Gambar 2.1 Imagen [6].

Gambar 2.1 merupakan hasil menggunakan model Imagen berdasarkan teks prompt: *'A photograph of two scuba divers observing a sea turtle swimming over a coral reef underwater'*. Secara visual, model berhasil merepresentasikan hubungan spasial antar objek dengan akurat. Objek utama berupa penyu ditempatkan di bagian tengah dengan detail tekstur tempurung yang tajam, sementara dua penyelam diletakkan di latar belakang untuk memberikan kedalaman ruang. Pencahayaan bawah air dan pemilihan palet warna biru-hijau menunjukkan kemampuan model dalam memahami konteks lingkungan secara fotorealistik.

2. DALL-E 3

OpenAI meluncurkan DALL-E 3, alat pembuat gambar yang memungkinkan realisasi visual dari konsep-konsep kreatif secara instan. Para penulis dapat “menghidupkan” karakter, latar, dan adegan-adegan dalam cerita mereka melalui gambar-gambar yang dihasilkan oleh AI. Visualisasi ini dapat membantu menemukan arah baru dalam proses kreatif, karena detail-detail yang sebelumnya tidak terlihat jelas ketika hanya digambarkan secara tekstual, kini dapat dilihat dengan jelas melalui gambar-gambar tersebut. Bagi para pencipta konten grafis, seperti produser video, kemampuan DALL-E untuk menghasilkan gambar-gambar asli dari deskripsi teks yang singkat membuat proses pembuatan elemenelemen visual yang menarik menjadi lebih mudah. Selain itu, alat ini juga dapat digunakan untuk merancang prototipe, membuat konsep iklan, dan menciptakan ikon-ikon, semuanya berdasarkan deskripsi teks saja. [29].

DALL-E 3 memanfaatkan ChatGPT untuk melakukan ekspansi *prompt*, yaitu proses mengubah instruksi singkat dari pengguna menjadi deskripsi teks yang jauh lebih panjang dan rinci. Teknik perluasan teks ini tidak hanya eksklusif untuk DALL-E 3, tetapi juga terbukti mampu meningkatkan kualitas hasil dari model *Text-to-Image* (T2I) lainnya, seperti SDXL dan Midjourney. Dengan menambahkan instruksi spesifik ke dalam ChatGPT, pengguna dapat menciptakan variasi gambar berkualitas tinggi. Secara umum, *prompt* yang lebih panjang dan mendetail akan menghasilkan gambar yang jauh lebih baik, seperti ketajaman detail objek dan akurasi teks yang lebih pas. Pendekatan ini sangat krusial, terutama saat membuat komposisi visual yang rumit seperti poster, buku, atau iklan yang hampir mustahil dieksekusi dengan baik tanpa instruksi yang mendetail.

Meskipun sederhana, memastikan model mematuhi instruksi secara presisi merupakan salah satu tantangan terbesar dalam sistem T2I. Model-model terdahulu sering kali gagal mengikuti instruksi teks dengan ketat, sehingga kerap menghasilkan objek atau atribut yang keliru. Namun, belajar dari kasus-kasus kegagalan tersebut, DALL-E 3 kini mampu menghasilkan gambar dengan jumlah objek, proporsi ukuran, serta atribut global maupun lokal yang tepat. Selain itu, model ini juga berhasil meminimalisir halusinasi atau kemunculan objek yang tidak diminta. Kemampuan tingkat tinggi ini sangat esensial dalam dunia desain, karena memberikan kebebasan dan rasa percaya diri bagi para desainer untuk menggunakan berbagai variasi diksi dalam mengontrol hasil gambar.

Lebih jauh lagi, model T2I juga dituntut mampu merealisasikan adegan yang di luar nalar atau surealis, seperti "seekor kuda menunggangi astronot." Fleksibilitas ini sangat penting dalam ranah desain yang kerap melibatkan kreasi imajinatif dengan kombinasi objek yang tak lazim. Berdasarkan pengujian menggunakan *prompt* sulit dari berbagai komunitas, DALL-E 3 terbukti unggul dalam menangani tata letak spasial, bentuk, serta atribut objek yang tidak biasa. Tantangan spesifik lainnya adalah menuliskan teks dengan benar dan proporsional di dalam gambar. Hal ini merupakan kebutuhan mendasar untuk keperluan desain, namun sering menjadi titik lemah bagi banyak model T2I pendahulu. Berdasarkan uji coba, DALL-E 3 sudah jauh lebih dapat diandalkan dalam merender teks, meskipun belum sepenuhnya sempurna untuk kalimat atau tata letak yang sangat kompleks. Evaluasi terhadap kendala umum yang sering ditemui pada model T2I terdahulu seperti kecacatan anatomi tangan dan wajah, replikasi gaya seni spesifik, serta pembuatan objek rumit juga menunjukkan kemajuan pesat. Saat ini, DALL-E 3 terbukti bekerja jauh lebih stabil dan andal dalam mengatasi kasus-kasus menantang tersebut [30].



Gambar 2.2 DALL-E 3 [6].

Gambar 2.2 merupakan hasil visualisasi dari model *generative* DALL-E 3 berdasarkan prompt spesifik. Dalam gambar tersebut, terlihat figur seseorang menggunakan busana samurai dengan memegang sebilah pedang katana sekaligus sebuah pizza, dikelilingi oleh empat orang dengan ekspresi terkejut dramatis. Hal ini menunjukkan bagaimana kecerdasan buatan mampu mengonstruksi narasi visual yang menggabungkan dua objek yang tidak lazim dalam satu bingkai secara koheren.

3. Midjourney

Midjourney adalah generator seni AI berbasis teks-ke-gambar yang mampu menciptakan karya visual utuh dalam hitungan menit berdasarkan instruksi singkat pengguna tanpa sekadar menggabungkan elemen foto yang sudah ada. Platform ini beroperasi secara unik melalui server Discord yang memungkinkan setiap orang menghasilkan gambar sekaligus berinteraksi dan mendiskusikan teknik perintah di berbagai kanal tematik secara langsung. Dibandingkan dengan generator lainnya, Midjourney sering menjadi pilihan utama karena memiliki karakter artistik yang kuat serta komposisi warna harmonis yang memberikan kesan estetika layaknya lukisan tangan yang nyata. Keunggulan tersebut terbukti saat para ahli menilai bahwa hasil generasinya mampu melampaui DALL-E 2 dan Stable Diffusion, khususnya dalam menciptakan tekstur serupa lukisan cat minyak dan menyampaikan nada emosional yang mendalam. Pengembangan platform ini terus berlanjut melalui pembaruan versi terbaru seperti v6 yang menyempurnakan parameter gaya serta

meningkatkan kemampuan sistem dalam memahami instruksi bahasa alami secara lebih presisi [31].

Midjourney merupakan bagian dari keluarga model *text-to-image* berbasis difusi yang bekerja dengan memulai dari gangguan acak (*random noise*) dan secara berulang melakukan proses "*denoising*" hingga membentuk gambar yang sesuai dengan perintah teks [32]. Model ini mempelajari pemetaan deskripsi bahasa ke dalam pola visual guna menyempurnakan struktur serta gaya gambar secara bertahap melalui berbagai tahapan proses teknis. Mengingat sistemnya sangat sensitif terhadap teks, penerapan *prompt engineering* seperti merinci subjek, pencahayaan, komposisi, hingga parameter seperti *aspect ratio* atau kode perintah *--style raw*, *--s*, *--ar*, dan versi model *-v* menjadi kunci utama dalam mengendalikan hasil akhirnya. Berbagai studi mengenai alat perintah menunjukkan bahwa eksplorasi dan penyempurnaan variasi *prompt* secara signifikan meningkatkan keselarasan antara keinginan pengguna dengan gambar yang dihasilkan sekaligus membantu penemuan gaya baru dengan lebih efektif. Kanal-kanal Discord milik Midjourney pun secara eksplisit berfungsi sebagai ruang kolaboratif yang digunakan pengguna untuk mempelajari serta saling bertukar strategi penyusunan instruksi teks tersebut [33].

Sistem *text-to-image* termasuk Midjourney telah meningkatkan produktivitas kreatif hingga sekitar 25% serta memungkinkan komunitas non-ahli untuk menghasilkan karya seni konsep, ilustrasi, dan desain secara instan. Teknologi ini kini dimanfaatkan secara luas dalam industri periklanan maupun berbagai solusi desain, bahkan tercatat pernah memenangkan beberapa kompetisi seni bergengsi [34]. Di sisi lain, proses pelatihan AI pada koleksi seni daring tanpa izin telah memicu kekhawatiran mendalam mengenai hak cipta dan peniruan gaya yang merugikan para seniman secara ekonomi maupun reputasi. Hal ini mendorong para praktisi kreatif untuk menyerukan regulasi yang lebih ketat, penggunaan alat perlindungan seperti Glaze, serta kejelasan aturan mengenai kepemilikan dan kepengarangan karya seni AI [35]. Fenomena tersebut menempatkan Midjourney sebagai medium yang sangat kuat namun kontroversial karena mengaburkan batasan antara kreativitas manusia dan hasil otomasi perangkat lunak.



Gambar 2.3 Midjourney [6].

Gambar 2.3 merupakan hasil generatif Midjourney AI, menampilkan representasi visual yang sangat realistis dan estetik, menggabungkan komposisi yang seimbang antara subjek dengan elemen dekoratif botani. Keunggulan AI dalam gambar ini terlihat pada manipulasi pencahayaan alami yang dramatis, di mana kontras antara cahaya matahari dan bayangan siluet serta tekstur yang mendetail pada bulu hewan maupun permukaan material di sekitarnya. Penggunaan palet warna yang menciptakan atmosfer domestik yang tenang, membuktikan kemampuan model generatif dalam menerjemahkan *prompt* tekstual menjadi sebuah karya visual yang memiliki narasi artistik yang kuat dan kualitas fotorealistik yang meyakinkan.

4. Grok-2

Grok-2 menandai fase evolusi krusial dalam pengembangan model bahasa besar (*large language models*) sebagai penerus dari Grok-1.5 yang terintegrasi secara mendalam dengan platform X. Dikembangkan oleh xAI, model ini dirancang untuk mengoptimalkan produktivitas dan interaksi pengguna di media sosial melalui pemanfaatan data *platform* secara *real-time* serta karakter kepribadian yang unik dan komunikatif [36]. Kehadirannya menggeser paradigma interaksi AI dari ruang privat ke ruang publik sosial, yang sekaligus memposisikan Grok-2 sebagai kompetitor utama bagi model global seperti GPT-4 dan Claude. Dari sisi teknis, Grok-2 membawa kemajuan pada aspek penalaran serta menyertakan fitur multimodal, termasuk pemahaman dan pembuatan gambar, sebuah tren utama dalam pengembangan LLM mutakhir saat ini. Aksesibilitas model ini ditentukan oleh ketersediaannya bagi pengguna Premium di platform X, serta identitas karakternya yang khas yang digambarkan lebih humoris dan "pemberontak" dibandingkan AI standar yang cenderung netral atau memiliki filter yang ketat. Perkembangan ini mencerminkan

pergeseran yang lebih luas dalam adopsi kecerdasan buatan, tingkat kepercayaan pengguna, serta dinamika kompetisi model fondasi di tingkat global [37].

Grok-2 diperkenalkan oleh xAI sebagai iterasi lanjutan dari Grok-1.5 yang dirancang khusus untuk platform X guna menyediakan asisten virtual cerdas dengan pemanfaatan data media sosial secara *real-time*. Implementasi publiknya di platform X menandai pergeseran besar dari penggunaan bot percakapan privat tradisional menuju ruang interaksi sosial yang lebih terbuka. Evolusi model ini menempatkan posisinya sejajar dengan kompetitor global utama seperti GPT-4o dan Claude 3.5 dalam peta persaingan kecerdasan buatan. Secara teknis, Grok-2 menunjukkan peningkatan signifikan dalam penalaran logis serta respons yang lebih peka terhadap konteks dibandingkan dengan versi pendahulunya. Model ini mengintegrasikan kemampuan multimodal yang memungkinkan pemrosesan teks sekaligus gambar, serta mendukung fitur pembuatan gambar langsung di dalam antarmuka platform [37].

Fitur-fitur tersebut selaras dengan tren pengembangan LLM global yang menuju pada model visi-bahasa terpadu untuk menyelesaikan tugas penalaran yang kompleks. Berbeda dengan banyak LLM yang mengutamakan netralitas, Grok-2 memiliki karakteristik persona yang humoris dan "pemberontak" untuk menciptakan interaksi yang lebih menarik bagi pengguna. Integrasinya dengan platform X memberikan akses instan ke aliran informasi dinamis, sebuah fitur yang dianggap unik di antara implementasi LLM besar saat ini. Meskipun memiliki keunggulan tersebut, akses penuh terhadap model ini terbatas bagi pengguna Premium, sementara pengguna gratis menghadapi batasan penggunaan yang lebih ketat. Peluncuran Grok-2 tidak hanya memposisikannya di jajaran pemimpin pasar, tetapi juga menyoroti tantangan terkait kepercayaan pengguna, privasi, serta urgensi strategi moderasi konten yang transparan pada platform berskala besar [38].



Gambar 2.4 Grok-2 [6].

Gambar 2.4 merupakan citra sintetis hasil Grok 2, menampilkan seorang wisatawan pria muda yang berdiri dengan pose profil, terlihat fokus mempelajari peta fisik yang dipegangnya. Latar belakang gambar menampilkan landmark ikonik Menara *Eiffel* yang buram, meskipun detail arsitekturalnya tidak tajam. Pencahayaan alami dari matahari terbenam menyinari pemandangan, menciptakan efek pada latar belakang dan bayangan lembut pada subjek. Komposisi ini berhasil menyeimbangkan subjek di latar depan dengan latar belakang yang dikenali, menciptakan narasi visual tentang penjelajahan di kota yang terkenal.

2.5 Deteksi Citra Sintetik

Deteksi citra sintetis merupakan salah satu bidang dalam visi komputer yang berfokus pada proses mengidentifikasi serta membedakan antara gambar buatan AI dan gambar asli. Kebutuhan terhadap sistem deteksi ini semakin meningkat seiring dengan perkembangan pesat metode sintetis gambar, khususnya yang berbasis *Generative Adversarial Networks* (GAN) dan *Diffusion Models*, karena kedua metode tersebut mampu menghasilkan citra dengan tingkat fotorealisme yang sangat tinggi. Secara umum, konsep deteksi citra sintetis melibatkan beberapa pendekatan utama yang saling melengkapi, yaitu [39]:

1. Identifikasi “sidik jari” buatan

Pendekatan dalam deteksi citra sintetis yang berasumsi bahwa setiap arsitektur model generatif meninggalkan jejak halus yang bersifat unik pada gambar yang dihasilkannya, layaknya tanda khas pada kamera digital. Jejak tersebut dapat berupa pola atau karakteristik kecil yang tidak kasat mata, namun konsisten muncul pada output dari model tertentu. Dalam proses deteksi, informasi ini dimanfaatkan untuk mengungkap apakah sebuah gambar merupakan hasil buatan AI. Lebih jauh lagi, teknik ini juga dapat digunakan untuk menelusuri atau mengidentifikasi arsitektur model generatif yang digunakan dalam pembuatan citra tersebut.

2. Analisis artefak visual tingkat tinggi

Salah satu pendekatan dalam deteksi citra sintetis yang berfokus pada keterbatasan model generatif dalam mereproduksi fitur semantik secara sempurna. Pendekatan ini mengamati adanya ketidaksempurnaan visual yang muncul pada gambar hasil generasi AI. Bentuk ketidaksempurnaan tersebut dapat berupa anomali pada warna,

kurangnya kesimetrisan alami seperti bentuk pupil mata yang tidak proporsional, serta inkonsistensi pada bayangan dan pantulan cahaya dalam suatu adegan.

Melalui indikator-indikator tersebut, sistem deteksi dapat mengidentifikasi apakah suatu gambar memiliki karakteristik sebagai citra sintetis.

3. Analisis fitur tingkat rendah dalam forensik digital

Pendekatan deteksi citra sintetis yang digunakan ketika artefak visual tingkat tinggi semakin sulit ditemukan akibat perkembangan metode generasi gambar. Pendekatan ini memfokuskan pada karakteristik teknis yang lebih mendasar yang berkaitan dengan arsitektur model pembuat citra. Salah satu aspek yang dianalisis adalah artefak yang muncul sebagai dampak dari proses *up-sampling* dalam tahap sintesis gambar. Artefak tersebut biasanya bersifat halus namun tetap dapat dideteksi melalui analisis forensik digital yang tepat. Dengan demikian, metode ini membantu mengidentifikasi citra sintetis meskipun secara visual tampak semakin realistis.

4. Ketidaksesuaian spektra

Merujuk pada perbedaan yang signifikan antara citra sintetis dan citra asli dalam kandungan sinyal pada frekuensi menengah hingga tinggi. Perbedaan ini dapat dianalisis melalui distribusi daya spektral baik secara radial maupun angular. Pada citra sintetis, pola distribusi frekuensi sering menunjukkan karakteristik yang tidak alami dibandingkan dengan citra asli. Hal ini menyebabkan adanya perbedaan yang dapat terdeteksi secara jelas pada domain frekuensi. Dengan demikian, ketidaksesuaian spektra menjadi salah satu indikator penting dalam proses identifikasi citra buatan.

Karakteristik dalam pendeteksian citra sintetis berfokus pada ketidakkonsistenan elemen visual tertentu. Berikut adalah rincian karakteristik tersebut [40]:

5. Ketidakselarasan refleksi pupil: Citra yang dihasilkan oleh AI seringkali memiliki kejanggalan pada area pupil mata, di mana bentuk refleksi cahaya antara mata kiri dan mata kanan tidak menunjukkan keselarasan atau simetri. Sebaliknya, pada mata manusia normal, pantulan cahaya akan membentuk kilauan yang konsisten di sekitar area pupil kedua mata.
6. Perbedaan bentuk kilauan cahaya: Pada citra asli hasil kamera, bentuk kilauan cahaya pada kedua mata cenderung sama. Namun, pada citra manipulasi AI, bentuk kilauan cahaya tersebut berbeda antara satu mata dengan yang lainnya.

7. Kualitas pencahayaan yang kurang sempurna: Meskipun AI mampu menghasilkan citra yang sangat mirip dengan aslinya, AI masih memiliki kekurangan dalam hal akurasi pencahayaan pada pupil mata, yang menjadi titik lemah utama untuk proses identifikasi.

2.6 Machine Learning

Machine Learning adalah kemampuan sebuah mesin yang dapat belajar secara mandiri berdasarkan input yang diberikan tanpa harus berulang kali diprogram secara eksplisit. Input tersebut merupakan data yang akan dilatih di dalam model *machine learning* dan berguna dalam prediksi pola yang dihasilkan berdasarkan data tersebut [41]. Manfaat menggunakan *machine learning* dalam peninjauan sistematis adalah dapat mengurangi jumlah pekerjaan manual yang diperlukan dalam proses tersebut. *Machine Learning* menggunakan algoritma yang mampu mempelajari pola dan hubungan yang kompleks dalam data, daripada mengandalkan pendekatan berbasis aturan. Ini memungkinkan pengguna untuk melakukan keputusan dengan tingkat akurasi yang lebih tinggi [42].

Dalam *Machine Learning*, sebuah program komputer diberikan serangkaian tugas yang perlu diselesaikan. Program tersebut dikatakan telah “belajar” dari pengalamannya jika kinerjanya dalam menyelesaikan tugas tersebut membaik seiring banyak latihan yang dilakukan program tersebut. Sebagai contoh, perangkat lunak yang mampu mendiagnosis kanker berdasarkan catatan medis pasien, perangkat lunak akan menganalisis data medis dari sejumlah besar pasien, kinerjanya akan meningkat seiring bertambahnya pengetahuan. Jumlah kasus kanker yang dapat diprediksi dan dideteksi dengan benar akan dijadikan sebagai tolok ukur kinerja algoritma tersebut. *Machine learning* telah digunakan berbagai bidang, seperti, robotika, asisten pribadi virtual, permainan video, pengenalan pola, pemrosesan *natural language*, *data mining*, prediksi lalu lintas, rekomendasi produk, prediksi pergerakan pasar saham, diagnosis medis, prediksi tindakan penipuan, serta penyempurnaan hasil pencarian di mesin pencari (seperti Google). Dalam AI, *machine learning* merupakan kemampuan untuk beradaptasi secara otomatis, dengan sedikit atau tanpa campur tangan manusia. *Machine learning* memungkinkan proses pelatihan menggunakan dataset yang lebih kecil. Namun, diperlukan campur tangan manusia yang lebih banyak untuk membantu proses pembelajaran dan memperbaiki kesalahan-kesalahan yang terjadi. *Machine learning* mengandalkan campur tangan manusia dalam mengklasifikasikan data dan menentukan atribut-atributnya [43].

Ada tiga paradigma utama yang digunakan untuk menggali pola dan pengetahuan dari data: *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*. Dalam *Supervised Learning*, model pembelajaran menggunakan data yang sudah diberi label sebagai dasar untuk prediksi atau klasifikasi. Sedangkan dalam *Unsupervised Learning* model harus menemukan pola dari data tanpa bantuan label eksternal. Meskipun tujuan dari kedua paradigma ini adalah untuk menghasilkan model yang mampu mengenali pola dalam data, perbedaan mendasar dalam pendekatan mereka memiliki dampak yang signifikan dalam aplikasi praktis dan hasil yang diharapkan. Sedangkan dalam *Reinforcement Learning* adalah sebuah algoritma yang mengakumulasi informasi (dalam bentuk sinyal penguatan) untuk memilih langkah-langkah yang menuju hasil yang diinginkan dengan tingkat optimal. Meskipun tujuan dari ketiga paradigma ini adalah untuk menghasilkan model yang mampu mengenali pola dalam data, perbedaan mendasar dalam pendekatan mereka memiliki dampak yang signifikan dalam hasil yang diharapkan [43].

2.6.1 *Supervised Learning*

Pendekatan *supervised machine learning* beroperasi dengan melibatkan peran "pengawas" atau pelatih. Konsep pengawasan ini merujuk pada proses pelatihan mesin menggunakan himpunan data yang telah diberi label, sehingga variabel hasil atau target yang ingin dicapai sudah diketahui secara pasti sejak awal [44]. Algoritma *supervised learning* memiliki dataset training yang digunakan untuk memandu komputer menghasilkan output yang sesuai dengan harapan. Algoritma *supervised learning* dilatih dengan *data training* yang sudah dilabeli (*labeled*), sehingga model dapat mempelajari pemetaan antara input dan output yang diinginkan. Algoritma di dalam *supervised learning* diklasifikasikan ke dalam dua kategori utama, yaitu: klasifikasi merupakan serangkaian algoritma yang dirancang untuk menentukan label atau kategori spesifik dari suatu data masukan. Metode ini diimplementasikan apabila variabel target yang dicari bersifat kategorikal. Sedangkan regresi merupakan sekumpulan teknik analisis yang digunakan untuk memprediksi hasil akhir atau memetakan hubungan antara satu maupun lebih variabel independen dan dependen. Pendekatan ini umumnya diterapkan ketika variabel target yang digunakan bernilai kontinu [45].

2.6.2 Unsupervised Learning

Unsupervised learning beroperasi tanpa adanya pengawasan secara eksplisit. Berbeda dengan *supervised learning*, pendekatan *unsupervised learning* melatih algoritma menggunakan dataset yang tidak memiliki label atau informasi mengenai output yang diharapkan. Dalam prosesnya, model tidak diberikan acuan jawaban benar atau salah, sehingga pembelajaran berlangsung tanpa intervensi manusia untuk melakukan koreksi atau penyesuaian parameter secara langsung. Oleh karena itu, *unsupervised learning* cenderung menghasilkan temuan yang lebih subjektif karena sistem berfokus pada penemuan struktur atau pola tersembunyi berdasarkan karakteristik dari data itu sendiri. Algoritma dalam *unsupervised learning* terbagi menjadi dua kategori utama, yakni: pengelompokan (*Clustering*) merupakan serangkaian algoritma yang bertugas memisahkan data tanpa label ke dalam berbagai kluster atau kategori spesifik, murni berdasarkan tingkat kesamaan maupun perbedaan karakteristik data. Asosiasi (*Association*) adalah algoritma berbasis aturan yang dirancang untuk melacak dan menemukan keterkaitan atau relasi antar berbagai elemen data dalam suatu dataset [46].

2.6.3 Reinforcement Learning

Reinforcement learning merupakan metode pembelajaran yang dilakukan melalui interaksi langsung dengan lingkungan di sekitar masalah yang dihadapi. Berbeda dengan pendekatan lain, dalam metode ini tidak diberikan instruksi secara eksplisit mengenai tindakan yang harus diambil, melainkan belajar secara mandiri melalui serangkaian tindakan yang dilakukan sendiri. Proses ini pada dasarnya merupakan pembelajaran berbasis percobaan dan kesalahandi mana model memilih langkah berdasarkan akumulasi pengalaman sebelumnya serta opsi yang tersedia saat ini. Setiap tindakan yang diambil tidak hanya memengaruhi situasi saat ini, tetapi juga berdampak pada kondisi lingkungan dan nilai imbalan di masa depan. Dalam proses pembelajaran *reinforcement learning* memanfaatkan mekanisme sinyal imbalan (*reward*) dan hukuman (*punishment*) sebagai pemandu. Model bertujuan untuk memahami kondisi lingkungan dan mengubahnya demi mencapai tujuan tertentu yang telah ditetapkan. Secara teknis, model akan menerima imbalan atas perilaku yang tepat dan mendapatkan hukuman atas tindakan yang salah, sehingga fokus utama algoritma adalah mengembangkan kemampuan untuk memilih tindakan yang dapat memaksimalkan total nilai imbalan yang diterima [47].

2.7 Deep Learning

Deep learning merupakan bentuk evolusi mutakhir dari *machine learning* di dalam AI yang dirancang untuk mempelajari berbagai hierarki konsep kompleks mulai dari elemen yang paling sederhana. *Deep learning* memanfaatkan jaringan saraf tiruan untuk mereplikasi mekanisme pembelajaran yang terjadi pada otak manusia. Sistem ini mampu beroperasi secara otonom untuk memperoleh informasi tanpa memerlukan aturan-aturan eksplisit atau intervensi manusia dalam perancangannya. Arsitektur *deep learning* dibangun di atas lapisan-lapisan algoritma yang memberikan interpretasi unik terhadap data input guna membangun hubungan antara input tersebut dengan label tertentu secara mandiri. Hal ini memungkinkan sistem untuk mengenali pola-pola yang kompleks dan menghasilkan prediksi yang akurat tanpa bergantung pada instruksi manual [48].

Secara struktural, algoritma komputasi *deep learning* dibangun di atas arsitektur jaringan berlapis yang dikenal sebagai *artificial neural networks* (ANN), berupaya meniru cara kerja jaringan saraf biologis pada otak makhluk hidup. Istilah "*deep*" pada penamaan teknologi ini secara khusus merujuk pada banyaknya jumlah lapisan pemrosesan bertumpuk yang harus dilalui oleh suatu data di dalam jaringan tersebut. Lapisan pemrosesan pertama pada model ini disebut sebagai lapisan input, yang bertugas menerima dan menyalurkan nilai data mentah menuju setiap neuron di lapisan berikutnya. Setelah lapisan input, terdapat sekumpulan lapisan *hidden layers* yang berfungsi menerapkan pembobotan matematika pada data masukan tersebut dan meneruskannya melalui sebuah fungsi aktivasi. Penggunaan fungsi aktivasi di dalam lapisan *hidden layers* krusial untuk membantu jaringan menambahkan sifat non-linearitas, sehingga mesin sanggup mempelajari berbagai hubungan kompleks di dalam data [48].

Pemrosesan data tersebut kemudian bermuara pada lapisan *output layer* yang menghasilkan hasil prediksi akhir, di mana jumlah unit pada lapisan ini selalu disesuaikan dengan jumlah target kelas keluaran yang diharapkan. Penyematan berbagai tingkatan abstraksi dan lapisan pemrosesan yang sistematis ini menghasilkan sebuah proses representasi pembelajaran yang jauh lebih terstruktur dan efisien. Untuk memastikan keakuratan prediksi, model *deep learning* memanfaatkan algoritma *backpropagation* yang bertugas menghitung tingkat kesalahan antara hasil keluaran sistem dengan nilai target yang sebenarnya. Algoritma ini kemudian akan bergerak mundur menuju lapisan tersembunyi guna meminimalkan nilai kerugian tersebut melalui mekanisme penyesuaian bobot matematis secara terukur. Seluruh proses evaluasi dan penyesuaian berulang tersebut akan

terus dieksekusi oleh mesin hingga hasil prediksinya mencapai tingkat akurasi yang valid untuk digunakan. Meskipun sangat canggih dan berkinerja unggul di banyak domain, fondasi ANN pada *deep learning* tetap memiliki sejumlah kelemahan, seperti rentan mengalami *overfitting* pada data pelatihan serta tidak adanya jaminan sistem akan selalu mencapai konvergensi solusi yang optimal [48].

2.8 Transfer Learning

Transfer learning merupakan teknik dalam pengembangan model *machine learning* di mana model yang sudah dilatih sebelumnya (*pre-trained model*) pada dataset besar digunakan sebagai dasar untuk menyelesaikan tugas atau masalah yang baru [49]. *Transfer learning* memiliki beberapa keuntungan dalam pengembangan model deteksi objek, antara lain sebagai berikut [50]:

1. *Transfer learning* memungkinkan model yang telah dilatih sebelumnya (*pre-trained model*) untuk lebih cepat disesuaikan dengan tugas baru tanpa harus memulai proses pelatihan dari awal.
2. Pendekatan *transfer learning* tetap mampu menghasilkan model dengan performa yang baik meskipun jumlah data pelatihan yang tersedia relatif sedikit, karena model telah memiliki pengetahuan dasar yang diperoleh dari proses pelatihan sebelumnya.
3. Model dapat menggunakan kemampuan ekstraksi fitur yang diperoleh dari dataset berskala besar untuk mengenali karakteristik visual umum, kemudian menyesuaikannya dengan karakteristik objek pada dataset target melalui proses pelatihan lanjutan.

Strategi *transfer learning* umumnya diterapkan melalui beberapa tahapan sebagai berikut [51]:

1. Pemanfaatan model *pre-trained*

Proses *transfer learning* diawali dengan menggunakan model yang telah dilatih sebelumnya (*pre-trained model*), misalnya pada dataset ImageNet. Model tersebut telah memiliki kemampuan dalam mengekstraksi fitur-fitur visual dasar sehingga dapat dimanfaatkan sebagai dasar untuk menyelesaikan tugas baru.

2. Tahap *feature extraction*

Pada tahap awal pelatihan, sebagian besar lapisan dasar (*backbone*) dibekukan (*freezing*) sehingga bobot hasil pelatihan sebelumnya tetap dipertahankan. Selanjutnya, lapisan klasifikasi diganti atau disesuaikan dengan jumlah kelas pada

dataset target. Strategi ini bertujuan agar model dapat beradaptasi dengan tugas baru tanpa kehilangan kemampuan ekstraksi fitur umum yang telah dipelajari.

3. Tahap *fine-tuning*

Setelah model mencapai performa yang stabil, sebagian lapisan yang sebelumnya dibekukan dibuka kembali (*unfreezing*) untuk dilatih ulang. Proses ini memungkinkan model mempelajari karakteristik yang lebih spesifik dari dataset target sehingga performa klasifikasi dapat ditingkatkan.

4. Penggunaan *learning rate* yang rendah

Pada tahap *fine-tuning*, proses pembaruan bobot dilakukan menggunakan *learning rate* yang relatif kecil. Pendekatan ini bertujuan untuk menyesuaikan parameter model secara bertahap tanpa menghilangkan pengetahuan umum yang telah diperoleh dari proses pelatihan sebelumnya.

5. Optimasi parameter model

Selama proses pelatihan, algoritma optimasi digunakan untuk memperbarui bobot model berdasarkan hasil *backpropagation*. Optimizer seperti Adam atau AdamW banyak digunakan karena mampu menyesuaikan laju pembelajaran secara adaptif sehingga proses konvergensi menjadi lebih stabil dan efisien.

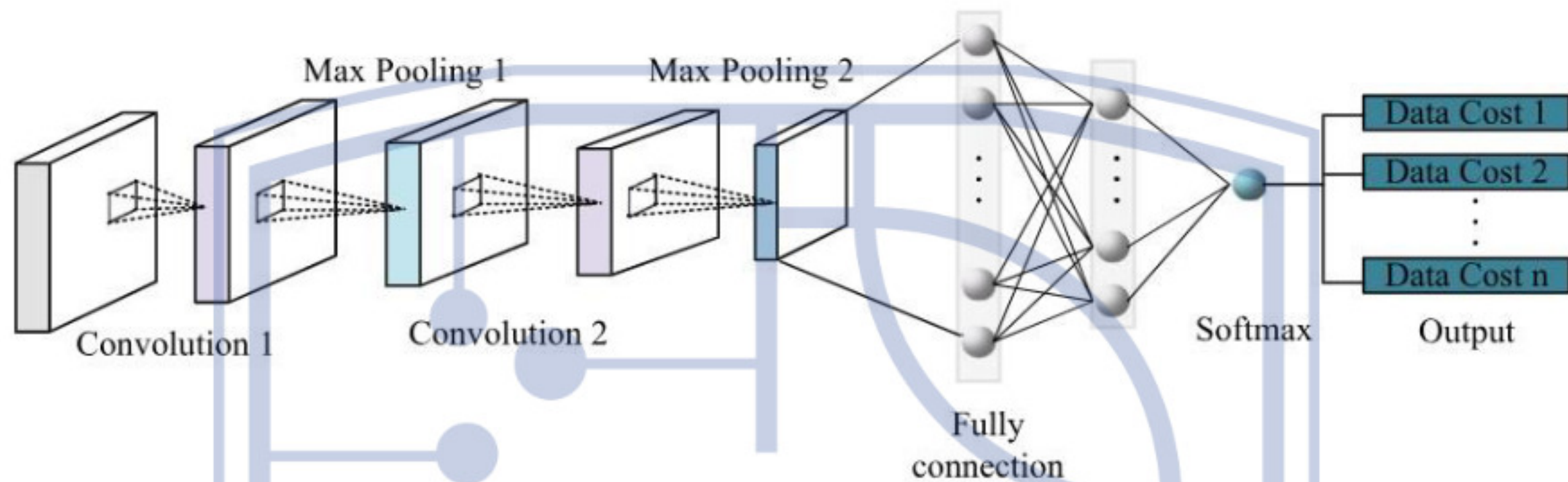
6. Integrasi modul tambahan

Model pre-trained dapat dipadukan dengan modul tambahan, seperti Convolutional Block Attention Module (CBAM), untuk meningkatkan kemampuan model dalam memfokuskan perhatian pada fitur-fitur yang paling relevan sebelum dilakukan proses fine-tuning. Hal ini memungkinkan model menghasilkan representasi fitur yang lebih diskriminatif pada tugas klasifikasi citra [52].

2.9 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) dirancang secara spesifik untuk mereduksi jumlah parameter komputasi secara signifikan pada sistem ANN, sehingga memungkinkan pengoperasian model berskala besar guna menyelesaikan tugas-tugas yang jauh lebih kompleks. Terinspirasi dari mekanisme kerja korteks visual pada hewan, arsitektur model ini beroperasi secara bertahap untuk mempelajari data mulai dari fitur tingkat rendah hingga berujung pada tingkat abstraksi yang paling tinggi. Sebagai ilustrasi dalam tugas klasifikasi citra, lapisan komputasi awal akan mendeteksi garis tepi sederhana terlebih dahulu, lalu secara berjenjang lapisan berikutnya mulai mengenali pola kompleks seperti bentuk spesifik

maupun wajah manusia. Secara arsitektural, model CNN dibangun berdasarkan tiga lapisan utama, yakni lapisan konvolusi dan *pooling* yang bertugas mengekstraksi fitur dari masukan, serta *fully connected layer* untuk memetakan hasil ekstraksi tersebut menuju prediksi keluaran akhir. Lapisan konvolusi mengawali tahapan ini dengan memproyeksikan sekumpulan susunan angka berukuran kecil, yang dikenal sebagai kernel, ke atas data masukan atau tensor guna mengambil karakteristik visual tingkat tinggi [53].

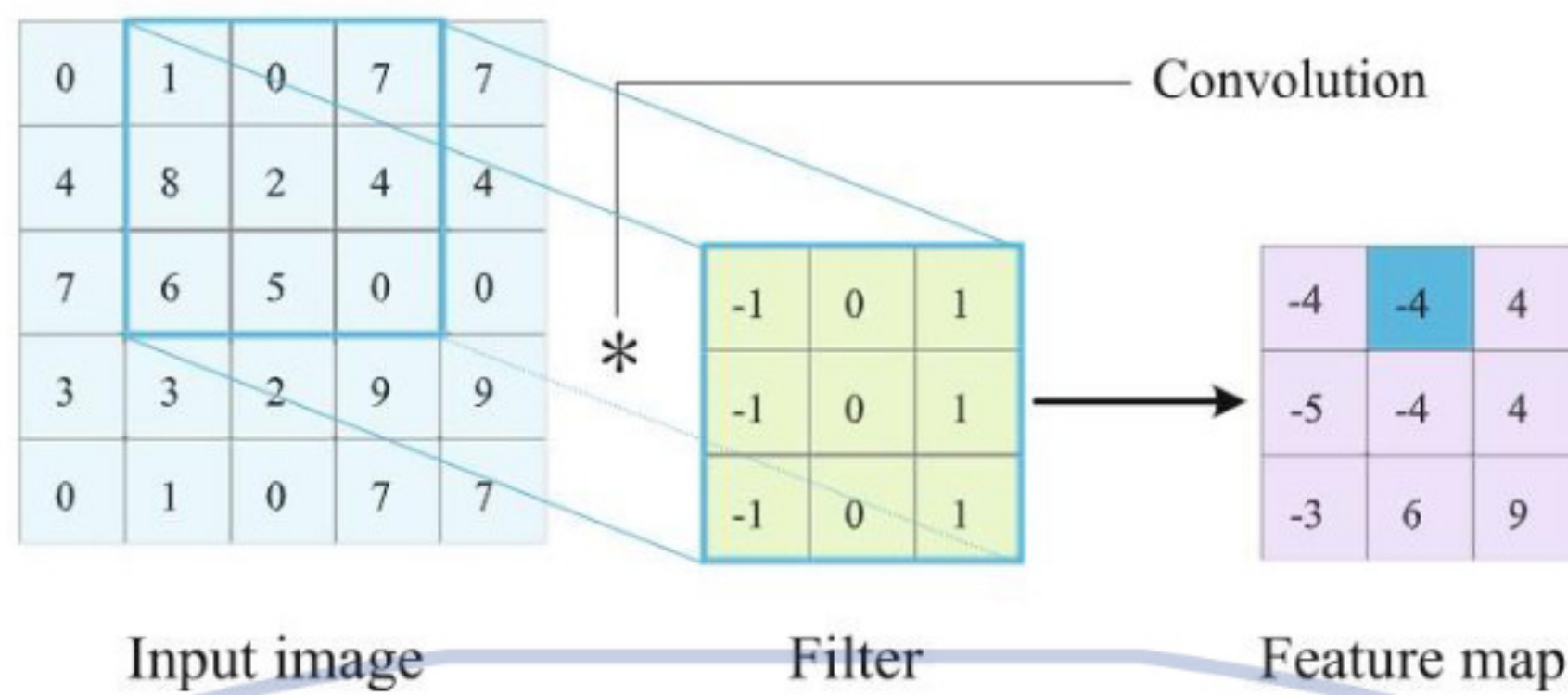


Gambar 2.5 Struktur CNN [53].

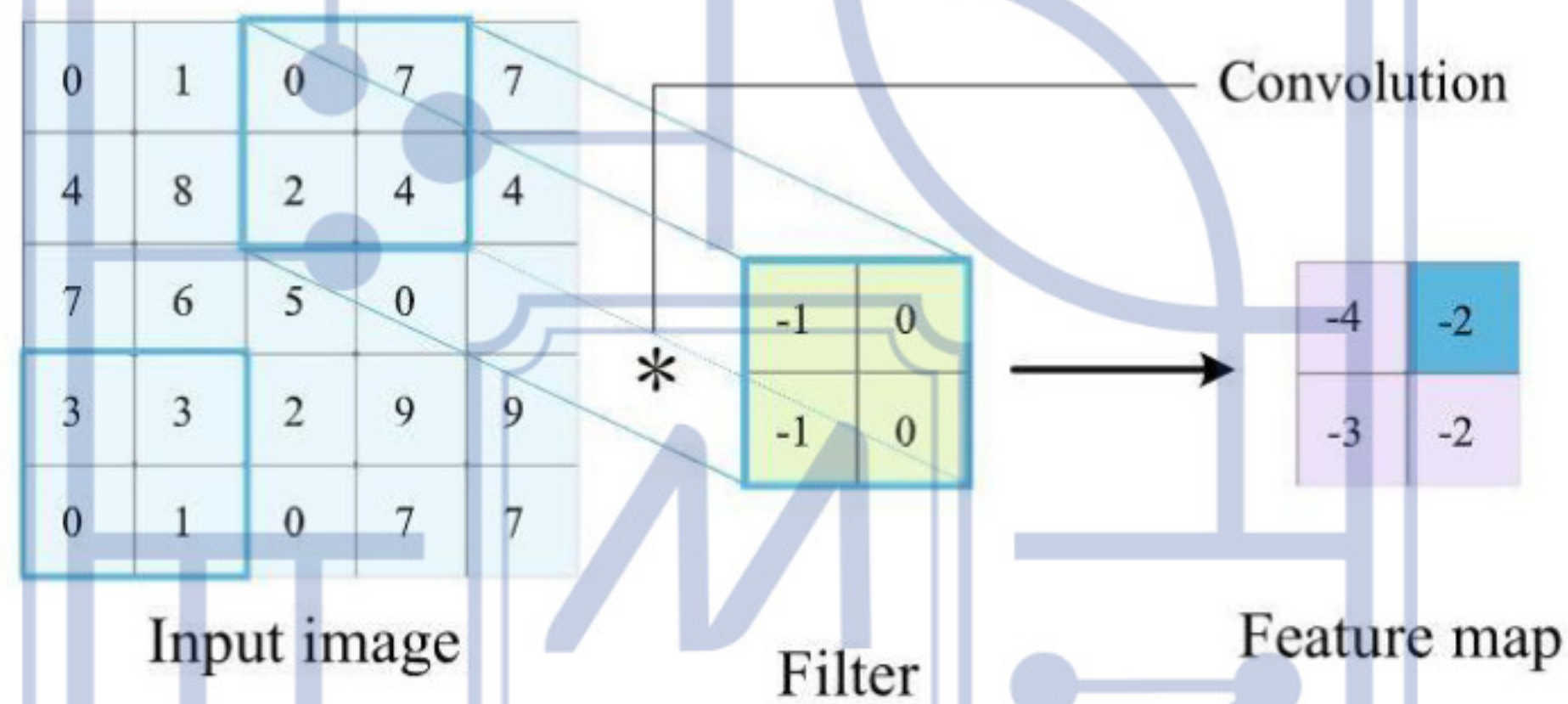
Arsitektur CNN terdiri dari beberapa lapisan utama, yaitu [53]:

1. *Convolutional Layer*

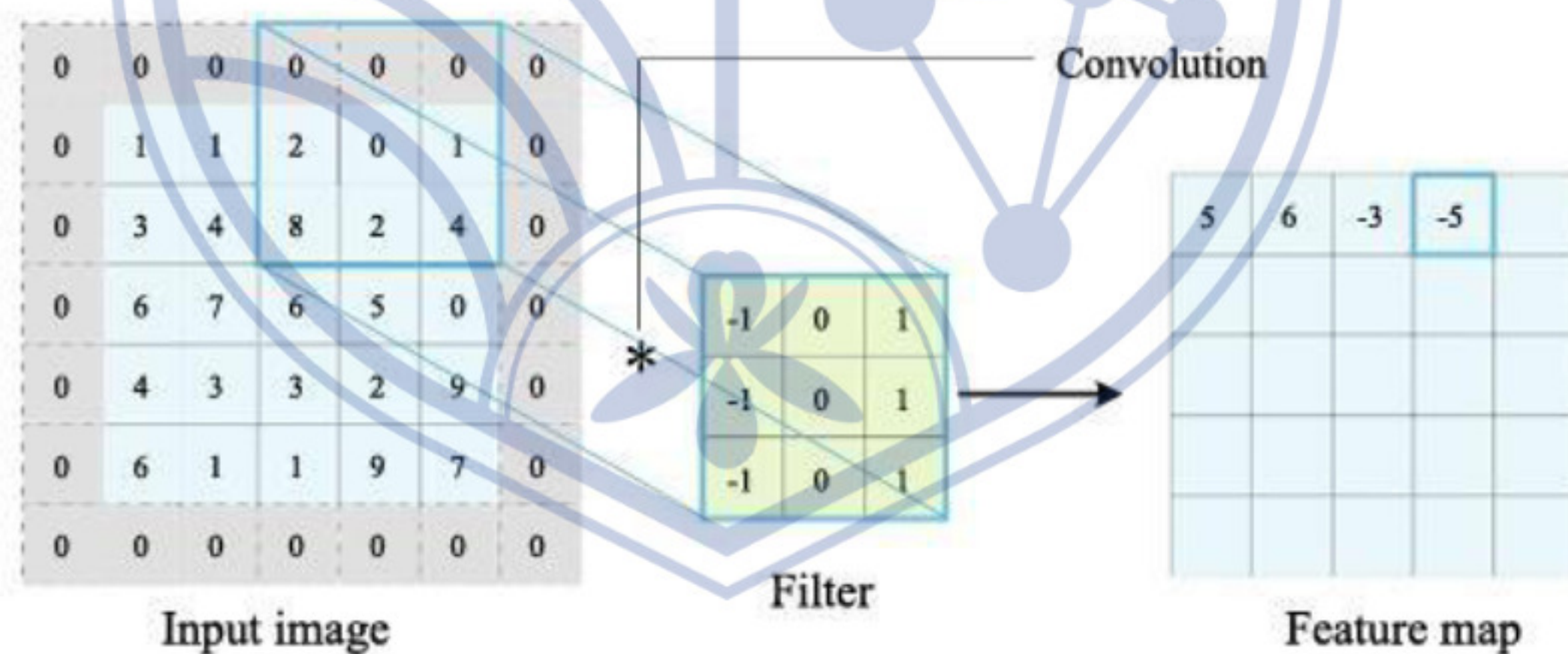
Lapisan konvolusi berfungsi secara khusus untuk mengekstraksi berbagai fitur penting yang terdapat di dalam sebuah gambar. Proses ini dimulai dengan penggunaan *Kernel* atau *Filter*, yaitu kumpulan neuron yang meluncur di atas permukaan gambar guna menghasilkan peta fitur yang informatif. Kecepatan dan ketajaman ekstraksi ini dipengaruhi oleh *Stride*, yang menentukan besarnya jumlah pergeseran filter saat menyisir area gambar tersebut. Meskipun *stride* yang besar dapat mempercepat waktu komputasi, penggunaannya perlu diwaspadai karena berpotensi mengurangi detail fitur yang tertangkap. Terakhir, teknik *padding* diterapkan dengan menambah piksel di pinggiran gambar untuk menjaga ukuran output tetap konsisten sekaligus mencegah hilangnya informasi penting pada bagian tepi [53].



Gambar 2.6 Convolution procedure [53].



Gambar 2.7 Convolution procedure (stride=(2,3)) [53].

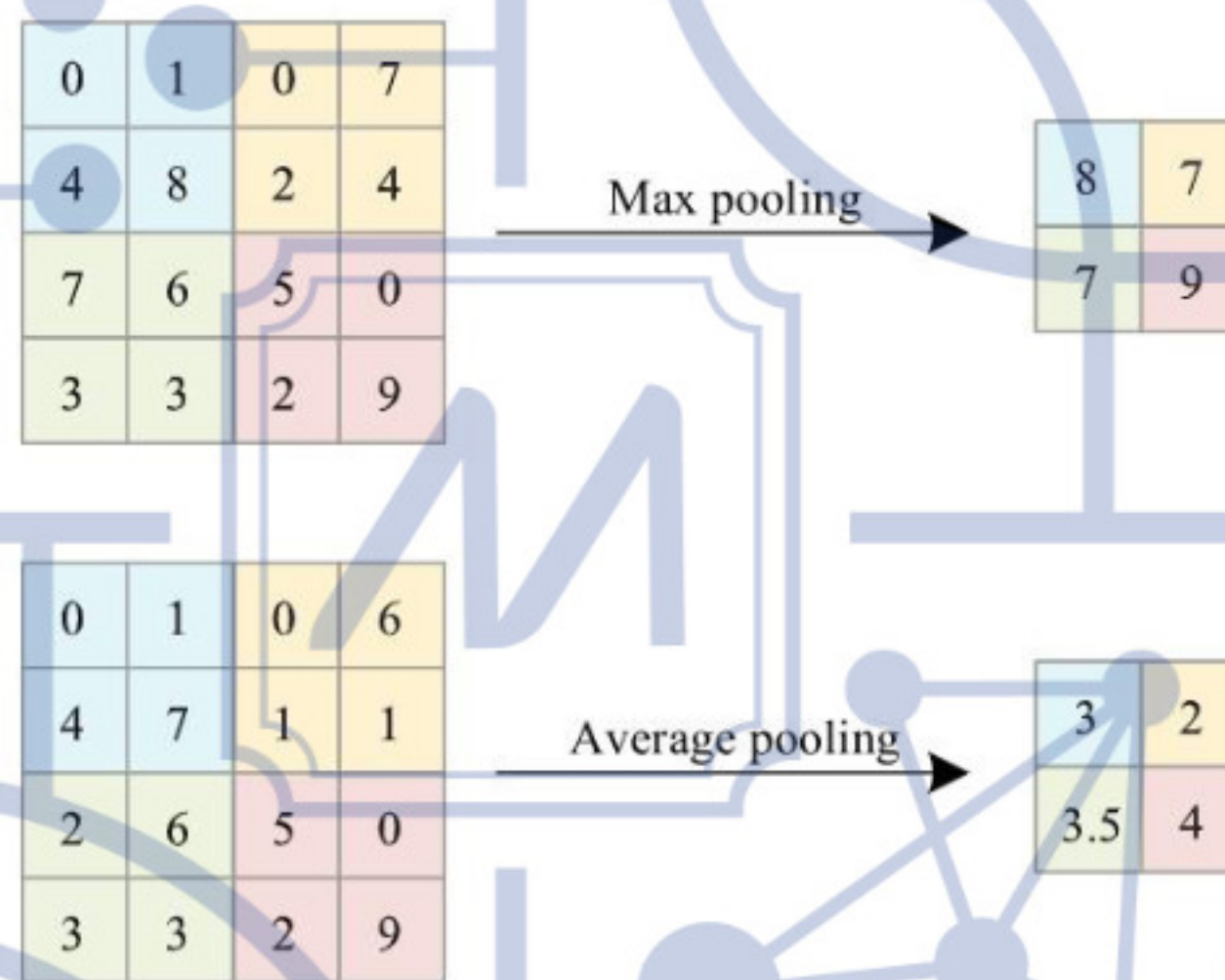


Gambar 2.8 Padding [53].

2. Pooling Layer

Pooling Layer untuk melakukan kompresi spasial dan *downsampling* guna mengecilkan dimensi matriks serta memperluas medan pandang lapisan berikutnya.

Lapisan ini bekerja secara deterministik menggunakan *sliding window* tanpa parameter yang dapat dilatih, di mana reduksi dimensi dilakukan melalui operasi statistik pada setiap saluran warna secara independen. Berbeda dengan konvolusi, standar arsitektur *pooling* umumnya meniadakan *padding* dan mengatur nilai *stride* yang sama dengan ukuran filter guna mencegah terjadinya tumpang tindih antar elemen spasial. *Max Pooling* berfungsi menyeleksi nilai maksimum *absolut* dalam jendela area untuk mempertahankan fitur kontras tajam, seperti garis tekstur yang menonjol pada gambar. Sementara itu, *Average Pooling* mengkalkulasi nilai rata-rata distribusi skalar, yang saat ini lebih sering diimplementasikan sebagai *Global Average Pooling* di akhir jaringan untuk mencegah risiko *overfitting* [53].

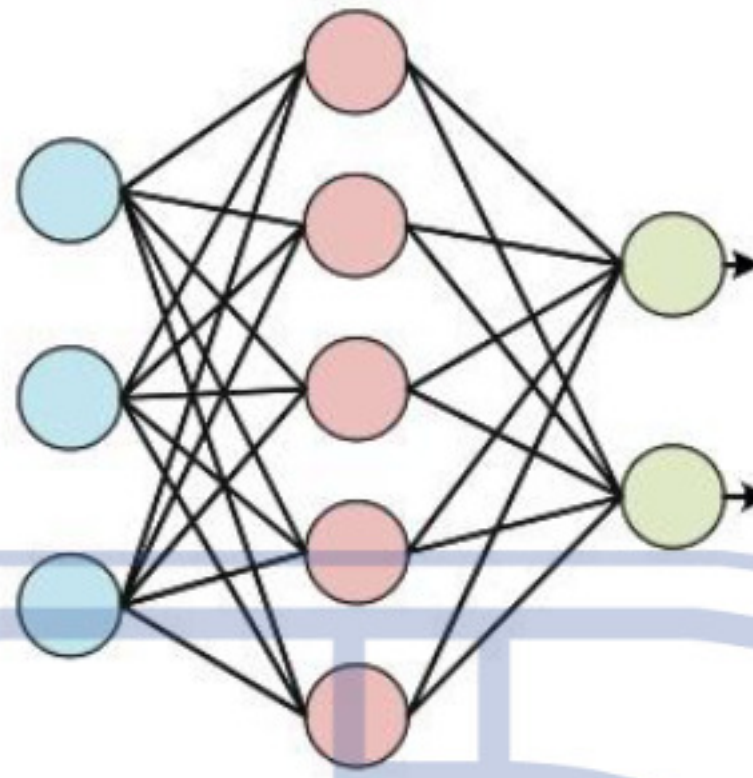


Gambar 2.9 Max Pooling dan Average Pooling [53].

3. Fully Connected Layer

Fully Connected Layer berfungsi menyerupai jaringan saraf tiruan konvensional yang menghubungkan setiap unit pemrosesan secara menyeluruh dan rapat. Tugas utamanya adalah menyatukan kembali fitur-fitur spasial yang sebelumnya terpisah menjadi satu kesatuan informasi untuk menentukan hasil klasifikasi akhir secara global. Proses pengolahan data dilakukan melalui perhitungan linear yang dipadukan dengan fungsi aktivasi guna menangkap pola hubungan antar data yang bersifat kompleks. Agar model tidak terjebak dalam kondisi menghafal data pelatihan secara berlebihan, para ahli biasanya menyisipkan teknik pelemahan saraf yang menonaktifkan sebagian koneksi secara acak. Langkah tersebut sangat penting

untuk meningkatkan kemampuan model dalam mengenali data baru yang belum pernah dilihat sebelumnya tanpa kehilangan akurasi [53].



Gambar 2.10 Fully Connected [53].

4. Flatten

Flatten bekerja dengan cara mengubah struktur data gambar yang sebelumnya berbentuk ruang tiga dimensi menjadi satu baris memanjang yang linier. Proses ini dilakukan dengan meleburkan seluruh informasi tata letak koordinat piksel agar data siap dimasukkan ke dalam lapisan saraf pusat. Melalui transformasi ini, ketergantungan posisi antar fitur dihapuskan secara permanen untuk menciptakan urutan elemen yang berurutan secara sistematis. Hasil penggabungan tersebut memungkinkan seluruh fitur yang telah diekstraksi dapat terhubung secara menyeluruh pada tahap pemrosesan berikutnya. Dengan demikian, lapisan ini berfungsi sebagai jembatan penting yang menyederhanakan data kompleks menjadi format yang lebih mudah dipahami oleh mekanisme klasifikasi akhir [53].

Selain lapisan di atas, terdapat beberapa *Activation Function* yang digunakan dalam arsitektur CNN, yaitu [53]:

1. Rectified Linear Unit (ReLU)

ReLU menjadi standar utama dalam pengembangan arsitektur jaringan saraf modern karena kesederhanaan proses perhitungannya dibandingkan fungsi lainnya. Secara mekanisme, fungsi ini bekerja dengan cara membatasi nilai input sehingga semua angka negatif akan diubah menjadi nol sementara nilai positif tetap dipertahankan. Keunggulan utama dari metode ini adalah kemampuannya dalam mengatasi kendala hilangnya gradien yang sering menghambat proses pembaruan informasi pada model. Selain itu, ketiadaan operasi matematika yang rumit membuat proses pelatihan jaringan menjadi jauh lebih cepat dan efisien secara

komputasi. Namun, penggunaan teknik ini memiliki risiko di mana sebagian unit pemrosesan bisa berhenti bekerja secara permanen jika terus-menerus menerima masukan dalam rentang negatif.

$$f(x) = \max(0, x) \dots\dots\dots (1)$$

Keterangan: x

= nilai input.

$\max(0, x)$ = fungsi yang memilih nilai terbesar antara 0 dan x .

2. Softmax

Softmax merupakan fungsi aktivasi yang dirancang khusus untuk menyelesaikan masalah klasifikasi dengan banyak kategori melalui pemetaan vektor input ke dalam rentang nilai probabilitas. Fungsi ini bekerja dengan menekan seluruh nilai output sehingga total penjumlahannya menjadi genap satu, yang kemudian merepresentasikan tingkat kepercayaan model terhadap setiap kelas. Berbeda dengan fungsi pencarian nilai maksimal standar, metode ini tetap memperhitungkan nilai-nilai kecil agar tidak terbuang begitu saja dalam perhitungan akhir. Karena penyebut dalam rumusnya menggabungkan seluruh faktor keluaran, maka setiap probabilitas yang dihasilkan memiliki keterkaitan erat satu sama lain secara matematis. Namun, penggunaan teknik ini tetap memiliki tantangan teknis seperti potensi munculnya neuron mati saat menghadapi input negatif dan kompleksitas perhitungan pada titik nol.

$$(2) \quad softmax(x) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \dots\dots\dots$$

Keterangan:

K = jumlah total kelas dalam tugas klasifikasi. x_i

= nilai input.

$\sum_{j=1}^K e^{x_j}$ = Simbol sigma berarti penjumlahan. Bagian ini menjumlahkan nilai eksponensial (e^x) dari seluruh kelas (j dari kelas ke-1 sampai kelas ke- K) untuk digunakan sebagai pembagi (normalisasi), sehingga total semua probabilitas output akhir menjadi tepat 1.

3. Sigmoid

Sigmoid digunakan dalam jaringan saraf karena kemampuannya memetakan seluruh nilai input ke dalam rentang antara nol dan satu secara halus. Karakteristik

gradiennya yang landai sangat berguna untuk mencegah terjadinya lonjakan nilai output yang drastis selama proses pengolahan informasi berlangsung. Namun, penggunaan fungsi ini memiliki kelemahan berupa pergeseran bias karena pusat distribusinya tidak berada di titik nol, sehingga memperlambat efisiensi pembaruan bobot jaringan. Selain itu, model ini rentan mengalami masalah gradien hilang karena laju perubahannya cenderung mendatar dan mendekati nol saat mencapai nilai ekstrem. Penggunaan operasi eksponensial dalam perhitungannya membuat proses komputasi model menjadi lebih lama dan memakan lebih banyak waktu dibandingkan metode lainnya.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

Keterangan: x = nilai input. e^{-x} = fungsi eksponensial dari nilai x .

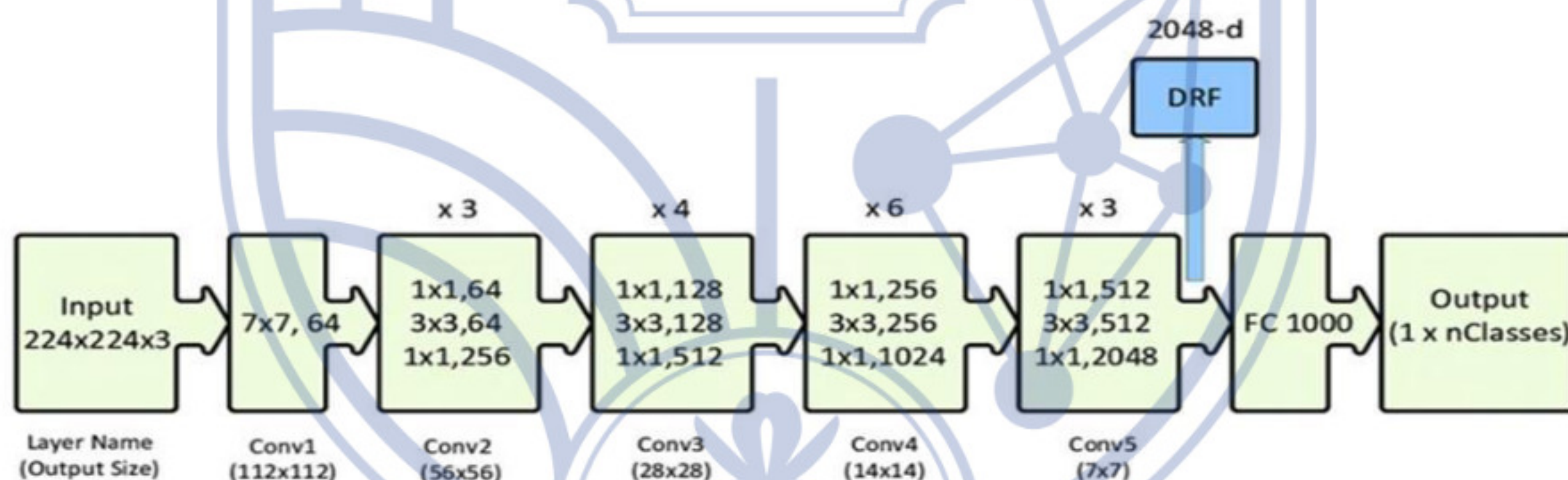
$1 + e^{-x}$ = penambahan angka 1 untuk memastikan hasil pembagian kurang dari atau sama dengan satu.

2.10 ResNet-50

ResNet-50 merupakan salah satu varian dari arsitektur *Residual Network* (ResNet), sebuah jenis Jaringan Saraf Konvensional (CNN) mendalam yang menggunakan koneksi residu atau shortcut connections (koneksi lewati). Sebelum adanya arsitektur ResNet, melatih jaringan saraf yang sangat dalam sangat sulit karena adanya masalah gradien yang menghilang (*vanishing gradient*). Dalam jaringan saraf tradisional, saat sinyal kesalahan (gradien) dikirim balik melalui banyak lapisan selama proses pelatihan, gradien tersebut bisa menjadi sangat kecil sehingga lapisan-lapisan awal tidak belajar secara efektif. Hal ini membatasi kedalaman jaringan yang bisa dibangun. Arsitektur *Residual Network* (ResNet) yang diperkenalkan oleh He Kaiming dan rekan-rekannya menghadirkan inovasi penting berupa koneksi lewati (*skip connection* atau *shortcut connection*). Mekanisme ini bekerja dengan tidak mengharuskan setiap lapisan jaringan mempelajari pemetaan baru secara penuh pada setiap proses pelatihan. Sebaliknya, koneksi tersebut memungkinkan gradien untuk mengalir secara langsung atau “melompati” satu atau beberapa lapisan dalam jaringan. Hal ini membantu memperlancar proses propagasi gradien selama pelatihan model berlangsung. Dengan adanya mekanisme tersebut, ResNet mampu mendukung pelatihan jaringan yang

lebih dalam sekaligus meningkatkan kemampuan sistem dalam mengekstraksi fitur yang lebih kompleks dan canggih [54].

Dalam ResNet dengan kedalaman 50 lapisan ke atas, efisiensi komputasi menjadi isu krusial. Jaringan tidak menggunakan blok residual sederhana, melainkan menggunakan arsitektur *Bottleneck*. Istilah *bottleneck* merujuk pada rancangan khusus pada *residual block* yang digunakan untuk memungkinkan jaringan memiliki lapisan yang lebih dalam tanpa meningkatkan kebutuhan komputasi secara berlebihan. Konsep tersebut dikembangkan agar model dapat bekerja secara lebih efisien sekaligus mempertahankan kemampuan dalam memperkirakan fitur kompleks. Berbeda dengan model ResNet yang lebih sederhana seperti *ResNet-34*, *ResNet-50* menggunakan blok *bottleneck* yang terdiri atas tiga lapisan konvolusi dalam setiap unit residual. Lapisan pertama menggunakan konvolusi berukuran 1×1 yang berfungsi untuk mengurangi jumlah dimensi atau jumlah filter pada data masukan. Proses ini disebut *bottleneck* karena mempersempit data sehingga beban komputasi menjadi lebih ringan. Setelah itu, lapisan konvolusi 3×3 digunakan untuk melakukan pemrosesan fitur utama pada data yang dimensinya telah diperkecil. Selanjutnya, lapisan konvolusi 1×1 terakhir digunakan untuk memperluas kembali dimensi data agar sesuai dengan jalur koneksi residual (*identity connection*). Contoh Implementasi, sebagai berikut [55]:



Gambar 2.11 ResNet-50 [55].

Desain *bottleneck* memberikan berbagai keuntungan bagi model *ResNet-50*. Dari sisi efisiensi komputasi, proses pengurangan dimensi sebelum konvolusi 3×3 mampu menekan beban perhitungan sehingga jaringan dapat dibuat lebih dalam hingga mencapai 50 lapisan dengan performa yang tetap efisien. Selain itu, struktur ini juga membantu mengatasi masalah *vanishing gradient* melalui penggunaan *residual connections* yang memungkinkan informasi mengalir langsung melewati beberapa lapisan. Dengan demikian, gradien dapat tetap terjaga selama proses pelatihan berlangsung. Keuntungan lainnya adalah peningkatan

kemampuan model dalam mengekstraksi fitur visual yang lebih kompleks dan mendeteksi perbedaan yang sangat halus, seperti membedakan gambar asli dengan sintetik hasil AI.

Residual learning merupakan pendekatan inovatif dalam *deep learning* yang dirancang untuk mengatasi berbagai hambatan utama dalam pelatihan jaringan saraf yang memiliki kedalaman sangat tinggi. Pendekatan ini menjadi solusi penting agar jaringan tetap dapat dilatih secara efektif meskipun terdiri dari banyak lapisan. Salah satu permasalahan utama yang diatasi adalah *vanishing gradient*. Pada jaringan saraf yang sangat dalam, nilai gradien dapat mengecil secara drastis saat proses *backpropagation*, sehingga informasi dari lapisan awal menjadi sulit diperbarui. Kondisi ini menyebabkan proses pembelajaran tidak optimal. Dengan adanya *residual learning*, ResNet-50 mampu menjaga aliran informasi agar tetap stabil selama proses pelatihan berlangsung.

Inti dari *residual learning* adalah penggunaan *skip connections* yang memungkinkan informasi melewati satu atau lebih lapisan dalam jaringan. Alih-alih mempelajari pemetaan fungsi secara langsung, model justru mempelajari selisih atau *residual* dari input. Hal ini membuat proses optimasi menjadi lebih sederhana karena informasi asli tetap dapat dipertahankan dan diteruskan ke lapisan yang lebih dalam. Dalam implementasinya, ResNet50 yang memiliki 50 lapisan menggunakan struktur blok *residual* secara berulang. Proses dimulai dari ekstraksi fitur dasar seperti tepi dan tekstur melalui lapisan konvolusi awal. Selanjutnya, fitur tersebut diproses melalui rangkaian blok *residual* untuk menangkap pola yang semakin kompleks. Aktivasi *Rectified Linear Unit* (ReLU) diterapkan setelah operasi konvolusi untuk menambahkan non-linearitas, diikuti dengan *pooling* untuk mengurangi dimensi spasial serta meningkatkan efisiensi komputasi [56].

Dalam penyesuaian dan pelatihan model ResNet-50, terdapat *hyperparameter* yang digunakan [57]:

1. *Batch Size*

Batch size adalah jumlah data yang diproses secara bersamaan selama estimasi gradien. *Batch size* yang besar akan memperlambat konvergensi jaringan. *Batch size* yang kecil dapat mengganggu jaringan dan menghasilkan hasil yang buruk. Oleh karena itu, suatu model harus memilih *batch size* yang tepat. *Batch size* yang kecil menghasilkan hasil yang lebih baik daripada yang besar, dengan kisaran antara 2 dan 32. *Batch size* sebesar 16, 32, dan 64 umumnya digunakan bersama dengan *hyperparameter* lainnya.

2. *Learning Rate* (LR)

Learning rate merupakan *hyperparameter* yang mengatur besar kecilnya langkah pembaruan parameter model pada setiap iterasi pelatihan untuk mencapai hasil yang optimal. Penentuan nilai *learning rate* yang tepat sangat penting karena secara langsung memengaruhi kinerja dan proses pembelajaran model. Nilai *learning rate* yang terlalu tinggi dapat menyebabkan proses pelatihan menjadi tidak stabil, sedangkan nilai yang terlalu rendah dapat memperlambat konvergensi. Selain itu, peningkatan *learning rate* dapat mempercepat konvergensi ketika dua pembaruan parameter secara berturut-turut memiliki arah vektor gradien yang sama, sehingga proses optimasi dapat berlangsung lebih efisien.

3. *Optimizer*

Optimizer merupakan komponen penting dalam proses pelatihan jaringan saraf karena berfungsi memperbarui bobot model serta menyesuaikan *learning rate* untuk meminimalkan nilai fungsi kerugian (*loss function*). Dalam arsitektur CNN, tersedia berbagai jenis *optimizer* yang memiliki karakteristik dan keunggulan berbeda sehingga pemilihannya perlu disesuaikan dengan karakteristik dataset dan kebutuhan pelatihan model. Salah satu *optimizer* yang banyak digunakan adalah Adam.

4. *Loss Function*

Loss function merupakan fungsi matematis yang digunakan untuk mengukur selisih antara hasil prediksi model dengan nilai target sebenarnya. Nilai *loss* menjadi dasar dalam proses pembaruan bobot jaringan melalui perhitungan gradien selama pelatihan. Semakin kecil nilai *loss*, semakin baik kemampuan model dalam melakukan prediksi. Oleh karena itu, proses *fine-tuning* dilakukan untuk meminimalkan nilai *loss* sehingga akurasi model dapat meningkat. Beberapa jenis *loss function* yang umum digunakan meliputi Categorical Cross-Entropy, Sparse Categorical Cross-Entropy, dan Binary Cross-Entropy. *Categorical Cross-Entropy* digunakan untuk klasifikasi multikelas dengan label dalam bentuk *one-hot encoding*, sedangkan *Sparse Categorical Cross-Entropy* digunakan untuk klasifikasi multikelas dengan label berupa indeks kelas sehingga tidak memerlukan *one-hot encoding*. Sementara itu, *Binary Cross-Entropy* digunakan untuk tugas klasifikasi biner dan bekerja bersama fungsi aktivasi *sigmoid* untuk menghasilkan probabilitas prediksi antara 0 dan 1. Karena penelitian ini berfokus pada klasifikasi dua kelas, yaitu citra

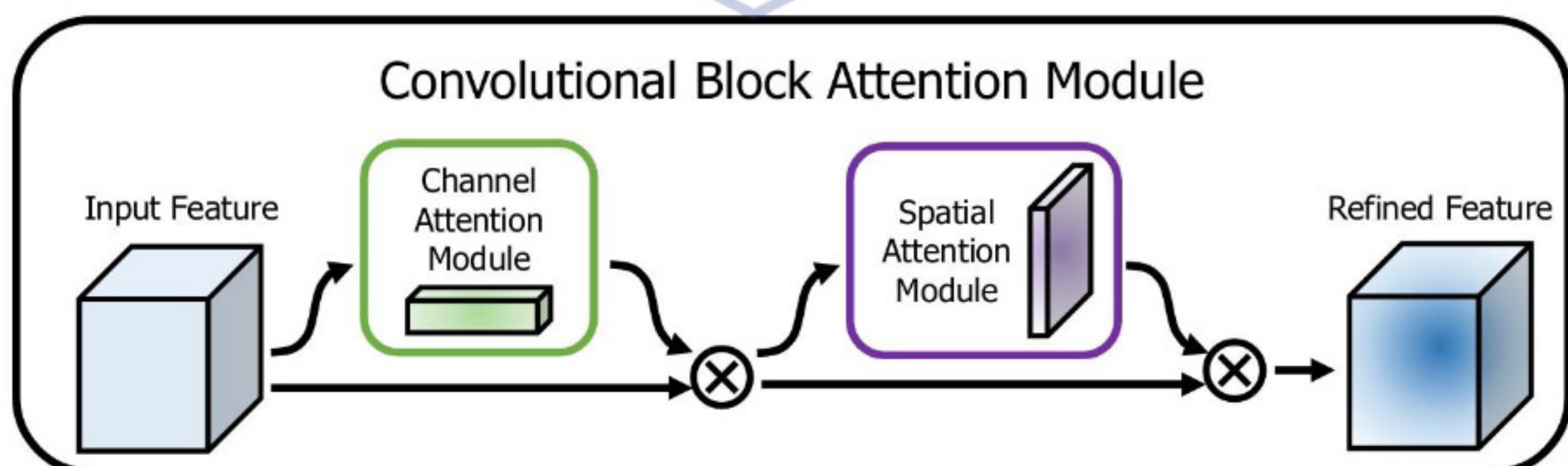
asli dan citra sintetis, maka *Binary Cross-Entropy* merupakan *loss function* yang paling sesuai digunakan.

5. *Epoch*

Epoch merupakan jumlah siklus pelatihan yang menunjukkan berapa kali seluruh data latih diproses secara lengkap melalui tahap *forward propagation* dan *backward propagation* selama proses pembelajaran jaringan saraf. Satu kali pemrosesan seluruh data latih umumnya belum cukup untuk memungkinkan model mempelajari karakteristik data secara optimal. Oleh karena itu, data latih perlu diproses berulang kali agar kemampuan generalisasi model terhadap data baru dapat meningkat. Meskipun demikian, penggunaan jumlah *epoch* yang berlebihan dapat menyebabkan *overfitting*, yaitu kondisi ketika model terlalu menyesuaikan diri dengan data latih sehingga performanya menurun pada data yang belum pernah dilihat sebelumnya. Dengan demikian, pemilihan jumlah *epoch* yang tepat menjadi faktor penting dalam proses pelatihan.

2.11 Convolutional Block Attention Module (CBAM)

Convolutional Block Attention Module (CBAM) adalah *attention module* yang serbaguna dan dapat digabungkan ke dalam berbagai arsitektur CNN. Tujuan utamanya CBAM adalah untuk secara selektif memfokuskan perhatian pada wilayah-wilayah penting dalam sebuah gambar, yang sangat bermanfaat untuk menangani gambar kompleks dengan banyak objek dan latar belakang yang rumit. CBAM terdiri dari dua komponen atau blok utama yang bekerja secara berurutan: *Channel Attention Module* (CAM) dan *Spatial Attention Module* (SAM). Modul ini bekerja dengan cara melakukan rekalisasi pada respon fitur di tingkat saluran (*channel*) dan peta fitur spasial dengan mempertimbangkan informasi konteks global maupun lokal.



Gambar 2.12 Convolutional Block Attention Module [11]

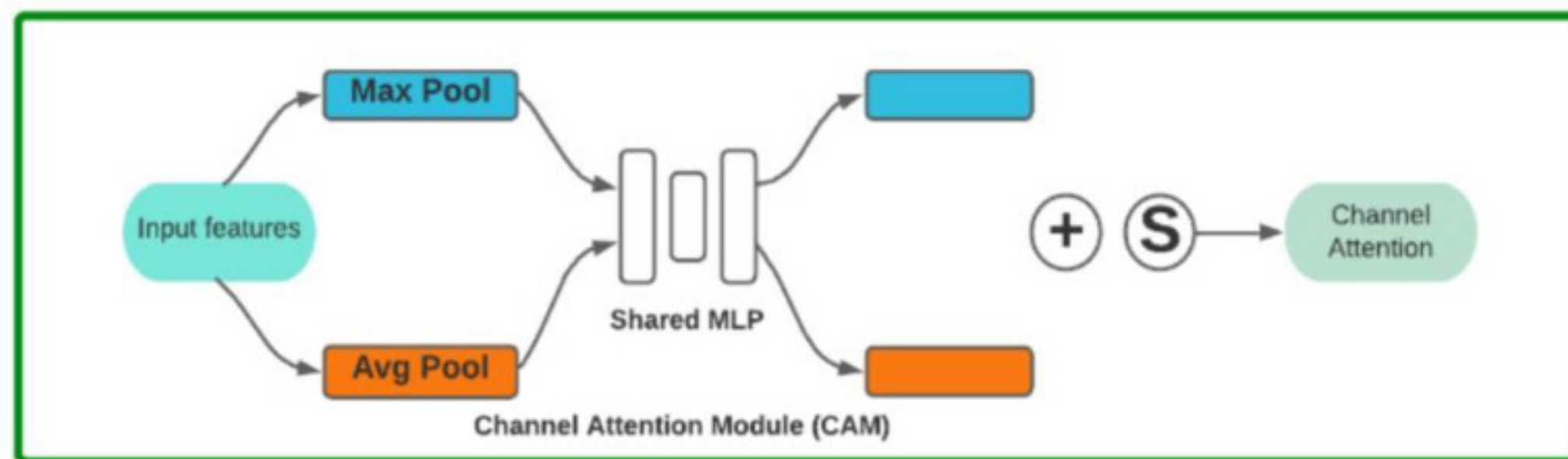
Untuk memahami mekanisme kerjanya secara lebih mendalam, modul CBAM terdiri dari dua blok pemrosesan utama yang bekerja secara sekuensial sebagai berikut [11]:

1. *Channel Attention Module (CAM)*

Channel Attention Module (CAM) difungsikan untuk melakukan kalibrasi ulang terhadap respons fitur antarsaluran secara adaptif dan dinamis. Pada tataran mekanis, kinerja modul ini sangat bergantung pada kemampuannya dalam menyaring sinyal informasi secara cerdas untuk memisahkan objek utama dari gangguan. Modul ini bekerja dengan cara memperkuat representasi fitur yang sarat akan informasi penting, sembari secara bersamaan mereduksi fitur-fitur yang dianggap kurang relevan seperti area latar belakang citra. Alur pemrosesan ini secara sistematis diawali dengan mengekstraksi peta fitur masukan (*input feature map*) untuk dikonversi menjadi vektor deskriptor pada masing-masing saluran komputasinya. Rumus untuk menghitung bobot blok perhatian saluran adalah sebagai berikut:

$$\alpha_c = \sigma(\text{MLP}(\text{AvgPool}(x_c)) + \text{MLP}(\text{MaxPool}(x_c))) \dots\dots\dots (4)$$

Bobot atensi untuk saluran c pada *output feature maps* CNN direpresentasikan oleh simbol α_c . Proses ini diawali dengan penggunaan x_c yang merupakan *feature maps* dari saluran c sebagai input utama bagi modul CAM. Untuk meringkas informasi spasial dan mengekstraksi karakteristik global dari *feature maps* tersebut, digunakan dua operasi penyusutan dimensi, yaitu *MaxPool* (pengambilan nilai maksimum) dan *AvgPool* (pengambilan nilai rata-rata). Setelah dimensi spasial direduksi, hasil dari kedua operasi *pooling* tersebut dimasukkan ke dalam lapisan MLP (*Multi-Layer Perceptron*) untuk menjalani transformasi non-linear. Output dari kedua jalur MLP ini kemudian digabungkan melalui penjumlahan elemen demi elemen (*elementwise*) dan diaktivasi menggunakan fungsi *sigmoid*, yang disimbolkan dengan σ guna menghasilkan bobot atensi final α_c . Pada tahap akhir, bobot α_c digunakan untuk memodulasi *feature maps* di setiap saluran. Mekanisme ini secara cerdas memberikan bobot yang lebih tinggi pada fitur yang dianggap informatif dan menekan fitur yang kurang relevan melalui pemberian bobot yang lebih rendah.



Gambar 2.13 Channel Attention Module [11]

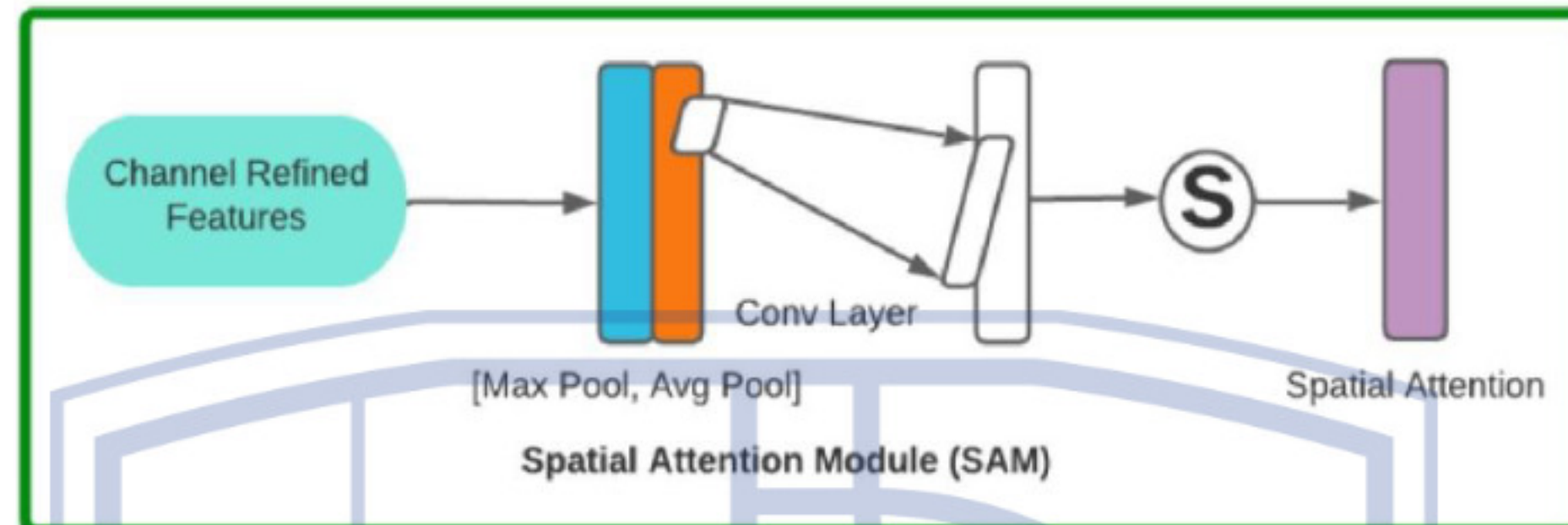
2. *Spatial Attention Module (SAM)*

Spatial Attention Module berfungsi untuk mengalibrasi ulang respons fitur secara adaptif dengan menitikberatkan fokus pada lokasi spasial yang informatif sekaligus menekan area yang tidak relevan. Proses ini dimulai dengan mengekstraksi peta fitur masukan menjadi vektor deskriptor spasial yang merangkum statistik penting, seperti nilai rata-rata dan nilai maksimum pada setiap koordinat. Selanjutnya, vektor-vektor deskriptor tersebut diproses melalui *Multi-Layer Perceptron* (MLP) untuk menghasilkan bobot atensi spasial yang akurat. Bobot yang telah dihasilkan kemudian dikalikan kembali dengan peta fitur asli guna menciptakan representasi fitur yang telah terkalibrasi secara spasial. Melalui mekanisme ini, model dapat secara otomatis mengidentifikasi bagian mana dari sebuah citra yang paling krusial untuk diproses lebih lanjut. Rumus yang digunakan untuk menghitung bobot pada blok atensi spasial adalah sebagai berikut:

$$\alpha_c = \sigma(MLP(AvgPool(x_c)) + MLP(MaxPool(x_c))) \dots\dots\dots (5)$$

Bobot atensi yang diberikan pada lokasi spasial (i, j) di dalam peta fitur input x direpresentasikan oleh simbol $\beta_{i,j}$. Untuk mengekstraksi informasi global sekaligus mereduksi dimensi spasial, digunakan dua operasi utama yaitu $AvgPool(x)$ dan $MaxPool(x)$. Pada setiap koordinat spasial (i, j) , dibentuk sebuah vektor fitur gabungan yang dituliskan sebagai $[x_{i,j}; AvgPool(x); MaxPool(x)]$. Vektor gabungan ini kemudian diproses melalui lapisan MLP untuk mendapatkan transformasi non-linear yang mendalam. Hasil dari lapisan MLP tersebut selanjutnya melewati fungsi aktivasi sigmoid σ untuk menghasilkan bobot atensi final $\beta_{i,j}$. Bobot $\beta_{i,j}$ berfungsi untuk memodulasi vektor fitur pada lokasi yang bersesuaian dalam peta fitur input. Melalui proses ini, sistem secara otomatis

memberikan bobot yang lebih tinggi pada lokasi spasial yang dianggap lebih informatif, sementara lokasi yang kurang relevan akan diberikan bobot yang lebih rendah.



Gambar 2.14 Spatial Attention Module [11]

2.12 Confusion Matrix

Dalam evaluasi performa model klasifikasi citra sintetis menggunakan akurasi saja tidak memberikan gambaran performa model. Akurasi adalah metrik yang menghitung persentase dari prediksi yang benar di antara semua prediksi. Model mungkin dapat mencapai akurasi yang tinggi dengan mengklasifikasi sebagian besar citra sebagai tanpa benar-benar mampu mendeteksi dengan baik. Akurasi tidak hanya memberi penjelasan tentang performa model. Oleh karena itu, metrik lain seperti *precision*, *recall*, dan *F1-Score* diperlukan untuk memahami kemampuan model lebih mendalam [58]. Metrik Evaluasi Kinerja model dievaluasi menggunakan kriteria evaluasi klasifikasi standar, yaitu:

1. *True Positive* (TP): Gambar yang dihasilkan AI diidentifikasi dengan benar sebagai gambar sintetis.
2. *False Positive* (FP): Gambar asli salah identifikasi sebagai sintetis.
3. *True Negatives* (TN): Klasifikasi yang benar terhadap citra asli sebagai asli.
4. *False Negatives* (FN): Gambar sintetis salah diklasifikasi sebagai asli.

Confusion matrix dapat digunakan untuk menghitung beberapa metrik yang lebih mendalam, antara lain [58]:

1. *Accuracy*: Akurasi untuk masalah terstruktur dengan mendefinisikan ulang apa yang dimaksud dengan negatif benar (TN) dan positif benar (TP) di seluruh jalur klasifikasi.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots (6)$$

2. *Precision*: Rasio antara prediksi positif yang benar (TP) dengan total seluruh prediksi positif (TP+FP). *Precision* mengukur sejauh mana yang diprediksi oleh model.

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots (7)$$

3. *Recall*: Rasio antara prediksi positif yang benar dengan jumlah data yang sebenarnya positif (TP+FN). *Recall* mengukur seberapa banyak bagian dari jalur yang seharusnya berhasil ditangkap oleh prediksi model.

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (8)$$

4. *F1-Score*: Rata-rata antara presisi dan recall yang memberikan skor tunggal untuk menyeimbangkan keduanya.

$$F1 = \frac{2TP}{2TP+FP+FN} \dots\dots\dots (9)$$

2.13 Grad-CAM

Grad-CAM (*Gradient-weighted Class Activation Mapping*) merupakan teknik *Explainable Artificial Intelligence* (XAI) yang digunakan pada model untuk menjelaskan proses pengambilan keputusan model. Teknik ini menghasilkan *heatmap* yang menunjukkan area pada gambar yang paling berkontribusi terhadap prediksi model, kemudian *heatmap* tersebut ditumpangkan pada gambar asli sehingga fokus perhatian model dapat divisualisasikan dan diinterpretasikan oleh manusia [59]. Tujuan penggunaan Grad-CAM adalah untuk menghasilkan visualisasi berupa *heatmap* yang menunjukkan area-area pada citra yang memberikan kontribusi terbesar terhadap prediksi model. *Heatmap* tersebut diperoleh dengan memanfaatkan informasi gradien dari kelas target pada lapisan konvolusi terakhir, sehingga dapat menggambarkan bagian citra yang menjadi fokus model selama proses klasifikasi. Dalam konteks deteksi *deepfake*, Grad-CAM digunakan untuk mengidentifikasi wilayah wajah yang diduga mengalami manipulasi, seperti area mata, hidung, pipi, atau mulut. Melalui visualisasi ini, pengguna dapat memahami bagaimana model mengambil keputusan, sehingga meningkatkan interpretabilitas, transparansi, dan kepercayaan terhadap hasil deteksi yang diberikan oleh model [60].

Proses pembentukan *heatmap* pada Grad-CAM dilakukan melalui tiga tahapan, yaitu sebagai berikut [61]:

1. Perhitungan Bobot Kepentingan (*Importance Weights*)

Tahap pertama adalah menghitung bobot kepentingan setiap *feature map* pada lapisan konvolusi terakhir. Bobot diperoleh dari rata-rata gradien skor kelas target terhadap setiap *feature map*. Nilai bobot ini menunjukkan seberapa besar kontribusi masing-masing *feature map* terhadap prediksi model. Semakin besar nilai bobot yang dihasilkan, semakin besar pula pengaruh *feature map* tersebut dalam menentukan kelas yang diprediksi.

2. Kombinasi Linier dan Aktivasi ReLU

Setelah bobot kepentingan diperoleh, setiap *feature map* dikalikan dengan bobot yang sesuai, kemudian seluruh hasil perkalian dijumlahkan untuk menghasilkan peta aktivasi gabungan. Selanjutnya, fungsi aktivasi Rectified Linear Unit (ReLU) diterapkan untuk mempertahankan hanya kontribusi positif terhadap prediksi, sedangkan kontribusi negatif diabaikan. Tahap ini menghasilkan peta aktivasi yang merepresentasikan wilayah-wilayah penting yang menjadi dasar pengambilan keputusan model.

3. Visualisasi Heatmap

Peta aktivasi hasil proses sebelumnya kemudian dinormalisasi dan dipetakan ke dalam gradasi warna sehingga membentuk *heatmap*. Warna dengan intensitas tinggi menunjukkan area yang memiliki kontribusi besar terhadap prediksi model, sedangkan warna dengan intensitas rendah menunjukkan area yang kurang berpengaruh. *Heatmap* tersebut kemudian ditumpangkan (*overlay*) pada gambar asli sehingga area yang menjadi fokus perhatian model dapat diamati secara visual. Pada deteksi citra sintetik, visualisasi ini umumnya menyoroti bagian yang sering mengandung artefak hasil manipulasi.

Heatmap yang dihasilkan oleh metode Grad-CAM menggunakan gradasi warna untuk merepresentasikan tingkat perhatian atau kontribusi setiap wilayah pada citra terhadap keputusan yang dihasilkan oleh model. Interpretasi setiap warna pada *heatmap* dijelaskan sebagai berikut [61]:

1. Wilayah Biru

Warna biru menunjukkan area yang memiliki tingkat kepentingan paling tinggi dalam proses pengambilan keputusan model. Wilayah ini merupakan bagian citra yang paling diperhatikan karena memberikan kontribusi terbesar terhadap hasil prediksi. Dalam deteksi citra sintetik atau *deepfake*, area berwarna biru umumnya

muncul pada fitur-fitur yang sering mengandung artefak atau ketidaksesuaian akibat proses sintesis.

2. Wilayah Hijau dan Kuning

Warna hijau dan kuning menunjukkan area dengan tingkat perhatian atau kontribusi sedang. Wilayah ini tetap memberikan pengaruh terhadap proses klasifikasi, namun kontribusinya tidak sebesar area yang ditandai dengan warna biru.

3. Wilayah Merah

Warna merah menunjukkan area yang memiliki kontribusi sangat rendah atau bahkan tidak memberikan pengaruh terhadap keputusan model. Dengan demikian, wilayah tersebut bukan merupakan fokus utama model dalam menentukan hasil klasifikasi.

Visualisasi *heatmap* menggunakan gradasi warna ini bertujuan untuk meningkatkan transparansi dan interpretabilitas model dengan memperlihatkan dasar pengambilan keputusan secara visual. Melalui visualisasi tersebut, pengguna dapat memverifikasi bahwa model melakukan klasifikasi berdasarkan karakteristik visual yang relevan dengan objek yang dianalisis, bukan berdasarkan informasi latar belakang atau fitur lain yang tidak berkaitan [61].

