BAB 2
TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 *Thumbnail*

Thumbnail adalah gambar miniatur yang mewakili isi dari sebuah video. Fungsi utama dari *thumbnail* adalah untuk menarik perhatian audiens agar tertarik mengklik video tersebut. Selain itu, *thumbnail* memiliki peran untuk memberi kesan pertama bagi audiens. *Thumbnail* yang menarik harus jelas, menarik visualnya, dan mencerminkan isi dari video dengan cara yang kreatif dan informatif. (Sendari, 2023).

Thumbnail yang menarik bukan hanya sekedar gambar, tapi juga memiliki kegunaan lain seperti membantu audiens memahami konten apa yang akan mereka tonton, sehingga dapat menambah kemungkinan mereka mengklik konten tersebut karena tampilan yang menarik dan informatif dari gambar thumbnail konten tersebut.

2.1.2 *Application Programming Interface (API)*

API adalah sistem yang memungkinkan dua komponen perangkat lunak untuk saling berkomunikasi dan berbagi data antar aplikasi yang berbeda. API menyediakan cara terstandarisasi dan aman bagi aplikasi untuk mengakses dan menggunakan data atau layanan yang disediakan oleh aplikasi atau platform lain. Konsep ini memudahkan integrasi antara sistem yang berbeda tanpa harus membangun kembali fitur yang sudah ada. API bekerja sebagai jembatan yang menghubungkan komunikasi antara klien dan server melalui pertukaran permintaan dan respons (Goodwin, 2024). Misalnya, ketika seorang pengguna melakukan pembelian di situs *e-commerce* dan memilih opsi pembayaran melalui pihak ketiga seperti PayPal, situs tersebut mengirimkan permintaan ke server menggunakan API. Permintaan ini kemudian dikirim ke *server web*. Setelah menerima permintaan yang valid, API memanggil program eksternal sistem pembayaran pihak ketiga, dan kemudian memproses permintaan dan mengirimkan respons kembali ke API. Data tersebut diteruskan kembali ke aplikasi awal sehingga pengguna dapat melihat hasil transaksi secara langsung, dengan proses yang berlangsung secara transparan di balik layar.

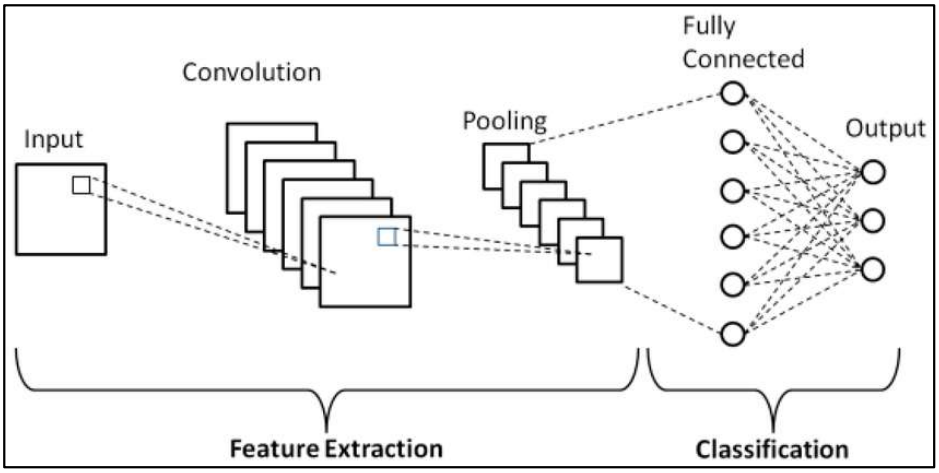
2.1.3 *Deep Learning*

Deep learning merupakan cabang dari machine learning yang memanfaatkan jaringan saraf tiruan (*artificial neural networks*) dengan banyak lapisan (*deep neural networks*) untuk mempelajari representasi data secara hierarkis. Setiap lapisan dalam jaringan ini mengekstraksi fitur dari data mentah, memungkinkan sistem untuk memahami data dengan tingkat abstraksi yang lebih tinggi tanpa intervensi manusia. Hal ini menjadikan deep learning sangat efektif dalam tugas-tugas seperti pengenalan suara, pengolahan citra, dan pemrosesan bahasa alami (Salim, 2023). Berbeda dengan pendekatan *machine learning* tradisional yang sering memerlukan rekayasa fitur secara manual, *deep learning* mampu secara otomatis mengekstraksi fitur dari data mentah. Kemampuan ini memungkinkan penerapan deep learning dalam berbagai bidang, termasuk pengenalan wajah, deteksi objek, dan analisis sentimen. Secara teori, *deep learning* dapat dipahami melalui pendekatan sistem dinamis dan kontrol optimal, di mana jaringan saraf dipandang sebagai sistem dinamis nonlinier yang memproses informasi secara berlapis-lapis (Salim, 2023). Pendekatan ini membantu dalam memahami bagaimana informasi disebar melalui lapisan-lapisan jaringan dan bagaimana pembelajaran terjadi dalam sistem tersebut (Liu, 2019).

2.1.4 *Convolutional Neural Network (CNN)*

Algoritma CNN adalah jenis jaringan saraf tiruan yang dirancang khusus untuk mengolah data berbentuk citra. CNN mampu mengekstraksi pola visual seperti warna, bentuk, dan tekstur melalui proses konvolusi dan *pooling*, sehingga sangat efektif digunakan dalam tugas klasifikasi gambar, termasuk pengenalan objek dan analisis desain visual. CNN terus menjadi arsitektur unggul dalam pengolahan citra dan pengenalan pola, berkat kemampuannya secara otomatis mengekstraksi fitur visual tanpa rekayasa fitur manual. Sebagai contoh, Gaur & Kumar (2025) mengembangkan CNN ringan untuk mendeteksi bencana dari citra udara, menekankan pentingnya desain arsitektur sederhana namun efektif dalam mengidentifikasi pola spasial . Selain itu, Khalifa & Keti (2025) dalam kajian mereka menyoroti tren penggunaan deep learning berbasis CNN untuk aplikasi IoT, termasuk monitoring visual dan deteksi objek real-time, dan menegaskan bahwa blok konvolusi–pooling–dense masih menjadi struktur dasar yang dominan .

Gambar arsitektur pada awal bab, seperti yang ditampilkan di atas (*feature extraction → downsampling → fully connected*), menggambarkan mekanisme inti CNN dalam mengurangi dimensi input sambil mempertahankan fitur penting.



Gambar 2.1 Arsitektur CNN

Berdasarkan Gambar 2.1, Arsitektur dasar CNN terdiri dari beberapa blok utama: pertama, lapisan konvolusi menerapkan filter untuk mendeteksi fitur seperti tepi dan pola; setelah itu, dilakukan *pooling* (misalnya *max pooling*) untuk mereduksi dimensi dan meningkatkan ketahanan model terhadap translasi. Selanjutnya, lapisan *fully connected* akan mengagregasi fitur-fitur ini untuk mengambil keputusan klasifikasi. Desain modal CNN yang terfokus memungkinkan penggunaan sumber daya komputasi secara lebih baik, sesuai kebutuhan dalam konteks integrasi sistem AI dengan platform seperti YouTube Thumbnail Analyzer.

2.1.5 *Confusion Matrix*

Confusion Matrix adalah tolak ukur untuk mengevaluasi kinerja suatu model dan secara visual ditampilkan sebagai matriks. Dalam matriks ini, terdapat empat buah kategori, yaitu:

1) *True Positive* (TP): model memprediksi suatu kejadian dan prediksinya tepat karena kejadian tersebut memang terjadi.
2) *True Negative* (TN): model memprediksi suatu kejadian, padahal sebenarnya kejadian itu tidak terjadi.

3) *False Positive* (FP): model dengan tepat memprediksi bahwa suatu kejadian tidak terjadi karena kejadian tersebut memang tidak terjadi..
4) *False Negative* (FN): model menyatakan bahwa kejadian tidak terjadi, tetapi kenyataannya kejadian tersebut terjadi.

Confusion Matrix dihitung dengan mendefinisikan kemungkinan hasil, mengumpulkan prediksi model, dan mengklasifikasikan hasil tadi kedalam 4 buah kategori. Manfaat dari matriks ini akan terasa apabila terdapat masalah seperti dataset yang tidak seimbang, di mana terdapat satu kelas yang berjumlah jauh lebih banyak daripada yang lain, kemungkinan model gagal menangkap kelas minoritas dengan benar meskipun akurasinya tinggi.

2.1.6 *Evaluasi Model*

Evaluasi model merupakan tahap penting untuk menilai sejauh mana model klasifikasi mampu memberikan hasil yang akurat dan sesuai dengan tujuan. Setelah model dilatih dan diuji, evaluasi dilakukan dengan menggunakan *Confusion Matrix*, yaitu tabel 2×2 (untuk klasifikasi biner) atau matriks lebih besar (untuk multi-kelas) yang menunjukkan hasil prediksi model terhadap data aktual. Berdasarkan nilai *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN) yang diperoleh dari *Confusion Matrix*, beberapa metrik evaluasi dapat dihitung sebagai berikut:

1) Accuracy

Accuracy mengukur proporsi prediksi yang benar terhadap total prediksi. Metrik ini menunjukkan seberapa tepat model dalam mengklasifikasikan semua kelas.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

2) Precision

Precision mengukur ketepatan model saat memprediksi kelas positif. Artinya, dari semua prediksi positif yang dilakukan model, berapa banyak yang benar-benar positif.

Precision = TP / (TP + FP)

3) *Recall* (*Sensitivity*)

Recall mengukur kemampuan model dalam menangkap semua data yang sebenarnya positif. Ini berguna ketika kesalahan negatif (*false negative*) harus dihindari.

Recall = TP / (TP + FN)

4) F1-*Score*

F1-*Score* adalah *harmonic mean* antara *Precision* dan *Recall*. Metrik ini memberikan keseimbangan antara keduanya dan berguna saat kita menginginkan kombinasi akurasi dan sensitivitas.

F1-*Score* = 2 × (*Precision* × *Recall*) / (*Precision* + *Recall*)

2.1.7 *View Ratio (Views per Subscriber)*

View Ratio adalah metrik performa relatif yang digunakan untuk mengukur efektivitas video YouTube dalam menjangkau audiens potensial berdasarkan jumlah *subscriber* channel. Metrik ini dihitung dengan membagi jumlah views sebuah video dengan jumlah *subscriber* channel pada saat video tersebut diunggah:

View Ratio = Jumlah *Views*/Jumlah *Subscriber*

Berbeda dengan metrik pasti seperti *views* atau *likes*, *View Ratio* memberikan gambaran yang lebih adil karena mnghitung ukuran audiens yang dimiliki channel. Sebuah video dengan 100.000 views dari channel berisi 10.000 *subscriber* tentu memiliki performa yang lebih tinggi dibandingkan video dengan views serupa dari channel dengan 1 juta *subscriber*. Oleh karena itu, *View Ratio* lebih tepat untuk mengevaluasi performa dari sebuah video, khususnya dalam konteks penelitian yang berfokus pada daya tarik visual seperti *thumbnail* dan judul. Penelitian oleh Violot dkk. (2024) manemukan bahwa metrik berbasis rasio seperti *views per subscriber* atau *likes per view* lebih informatif dibandingkan hanya melihat jumlah *views* mentah. Dalam studi tersebut, *Shorts* terbukti memiliki *View Ratio* lebih tinggi, mencerminkan efektivitas desain visual dalam menarik klik. Hal ini sejalan dengan temuan Yang dkk. (2022) yang menyatakan bahwa keputusan pengguna untuk menonton video sangat dipengaruhi oleh elemen visual awal.

Dalam penelitian ini, *View Ratio* digunakan untuk mengklasifikasikan performa video ke dalam tiga kategori, yaitu: rendah (< 5%), sedang (5%–15%), dan tinggi (> 15%). Batasan ini merujuk pada praktik umum komunitas kreator dan

temuan Violot dkk. (2024) yang menunjukkan bahwa video dengan *View Ratio* di atas 15% tergolong sangat menarik. Pengelompokan ini mempermudah evaluasi performa berdasarkan daya tarik awal yang ditentukan oleh elemen visual, tanpa dipengaruhi faktor lain seperti kualitas isi atau *engagement* pasca-klik. Dengan mempertimbangkan pengaruh kuat dari tampilan awal video terhadap klik pengguna, *View Ratio* merupakan metrik yang paling relevan dan ideal dijadikan dasar pelabelan performa video dalam penelitian ini.

2.1.8 *K-Means Clustering*

K-Means *Clustering* merupakan salah satu metode *clustering* yang digunakan untuk mengelompokkan data ke dalam sejumlah *cluster* berdasarkan kedekatan jenis data terhadap pusat kelompok atau *centroid*. Tujuan utama K-Means adalah membentuk kelompok data yang memiliki karakteristik serupa dengan meminimalkan jarak antara setiap data dengan *centroid* pada *cluster* tempat data tersebut berada. Proses K-Means dilakukan dengan menentukan jumlah *cluster* (K), menentukan *centroid* awal, mengelompokkan setiap data berdasarkan *centroid* terdekat, kemudian menghitung kembali posisi *centroid* berdasarkan anggota masing-masing *cluster*. Proses tersebut dilakukan secara berulang hingga kondisi konvergensi tercapai (Jain, 2010; Macqueen, 1967).

Dalam penerapannya, nomor *cluster* yang dihasilkan oleh algoritma K-Means berfungsi sebagai identitas kelompok dan tidak secara langsung menunjukkan tingkat performa tertentu. Oleh karena itu, diperlukan proses interpretasi terhadap karakteristik setiap *cluster* untuk menentukan makna dari kelompok yang terbentuk. Interpretasi dapat dilakukan dengan membandingkan karakteristik atau nilai rata-rata variabel pada masing-masing *cluster*. Pendekatan K-Means juga telah digunakan untuk mengelompokkan video YouTube berdasarkan karakteristik metrik performanya. Widjaja dan Oetama (2020), misalnya, menggunakan algoritma K-Means untuk mengelompokkan video YouTube berdasarkan views, likes, dan dislikes dan menghasilkan tiga cluster dengan karakteristik yang berbeda.

Penelitian Asyifaa dan Muhammad (2024) juga menerapkan K-Means untuk mengelompokkan video YouTube berdasarkan *views* dan *likes*. Penelitian tersebut menggunakan tiga cluster dan menginterpretasikan hasil pengelompokan menjadi tiga kelompok *engagement*, yaitu *high*, *medium*, dan *low*. Hasil penelitian tersebut

menunjukkan bahwa hasil clustering dapat digunakan untuk mengidentifikasi kelompok video berdasarkan tingkat karakteristik *engagement*-nya. Selain pada data video YouTube, K-Means juga digunakan untuk mengelompokkan data berdasarkan tingkat performa. (Rakhmawaty dkk, 2022) menerapkan K-Means dalam analisis kinerja pegawai dan membandingkannya dengan metode *Fuzzy* C-Means (FCM). Penelitian tersebut menunjukkan penerapan *clustering* untuk membentuk kelompok berdasarkan karakteristik kinerja. Dengan demikian, hasil *cluster* dapat diinterpretasikan berdasarkan karakteristik performa yang dimiliki oleh masing-masing kelompok.

2.2 Penelitian Terdahulu

Penelitian terdahulu merupakan landasan penting dalam memahami penerapan metode CNN dan *Natural Language Processing* (NLP). Penelitian oleh Pornpanvattana dkk. (2024) menggunakan CNN dengan arsitektur Xception untuk mengklasifikasikan *thumbnail* video dari tiga kategori (makanan, teknologi, perjalanan) di YouTube Thailand. Dataset berisi ribuan thumbnail, dan model mencapai akurasi 88%. Kontribusinya adalah sistem rekomendasi desain *thumbnail* berbasis visual lokal. Namun, penelitian ini tidak mempertimbangkan elemen teks seperti judul video. Sementara itu, penelitian oleh Zhao dkk. (2024) menggabungkan CNN untuk *thumbnail*, NLP *embedding* untuk teks (judul dan deskripsi), serta MLP untuk data numerik (durasi, *view count*) dalam pendekatan multimodal. Dataset mencakup ribuan video dari berbagai genre. Model mencapai akurasi prediksi engagement sebesar 91,2%. Penelitian ini menunjukkan keunggulan pendekatan multimodal, meskipun tidak menghasilkan rekomendasi praktis bagi kreator.

Penelitian lain oleh Wang dkk (2025) meneliti pengaruh fitur linguistik pada judul video menggunakan NLP. Dataset terdiri dari 6.512 video dari 102 kanal YouTube di Taiwan. Model menggunakan *Random Forest* dan KNN *Regression* untuk memprediksi *views*, *likes*, dan komentar, dan menghasilkan akurasi tinggi. Ini adalah bukti bahwa elemen teks dapat memengaruhi performa video, tanpa tambahan visual. Penelitian oleh (Widjaja & Oetama, 2020) menggunakan algoritma K-Means *Clustering* untuk mengelompokkan video YouTube berdasarkan karakteristik performanya. Variabel yang digunakan dalam proses

pengelompokan meliputi *views*, *likes*, dan *dislikes*. Hasil penelitian menghasilkan tiga kelompok (*cluster*) dengan karakteristik performa yang berbeda, yaitu kelompok video dengan tingkat popularitas tinggi, sedang, dan rendah. Penelitian tersebut menunjukkan bahwa K-Means dapat digunakan untuk menemukan kelompok performa video berdasarkan karakteristik data tanpa memerlukan label performa yang telah ditentukan sebelumnya. Seluruh penelitian di atas fokus pada aspek yang berbeda yaitu visual, teks, atau gabungan keduanya. Namun belum ada yang mengintegrasikan data visual dan non-visual dan menghasilkan rekomendasi praktis. Penelitian ini dilakukan untuk mengisi celah tersebut dengan membangun model klasifikasi multimodal berbasis CNN dan NLP, yang digabungkan untuk memprediksi performa video.

Tabel 2.1 Perbandingan variabel penelitian

Penulis	Metode	Hasil	Dataset	Kontribusi
Pornpanvattana dkk. (2024)	CNN (Xception) untuk klasifikasi visual thumbnail	Akurasi tertinggi: 88%	Ribuan *thumbnail* YouTube dari kategori Food, Tech, Travel	Rekomendasi desain visual berdasarkan kategori konten, fokus pada analisis visual saja.
Zhang dkk. (2023)	CNN (thumbnail) + NLP embedding (judul & deskripsi) + MLP (durasi, views) fused	Akurasi terbaik: 91.2%	Video YouTube multikategori dengan metadata lengkap	Model multimodal prediksi *engagement*, namun tidak menyediakan rekomendasi desain/penulisan judul
Wang dkk. (2025)	NLP (LIWC) + ML (Random Forest, SVM) untuk analisis judul video	Akurasi terbaik: 84.5%	6.512 video dari 102 kanal YouTube di Taiwan	Menganalisis pengaruh fitur linguistik judul terhadap *engagement* tanpa memasukkan aspek visual
Widjaja & Oetama (2020)	K-Means Clustering	Video Youtube dengan variabel *views*, *likes*, dan *dislikes*.	3 *cluster* performa	Mengelompokkan video berdasarkan karakteristik performa menggunakan K-Means