BAB 2
LANDASAN TEORI

2.0.1 Penelitian Terdahulu

Penelitian mengenai prediksi pemenang pada permainan *esports* secara garis besar terbagi menjadi dua pendekatan berdasarkan sumber data: berbasis telemetri internal *game* dan berbasis analisis visual dari rekaman pertandingan.

Pada ranah *Multiplayer Online Battle Arena* (MOBA), Yang et al. mengusulkan *Two-Stage Spatial-Temporal Network* (TSSTN) untuk memprediksi kemenangan secara *real-time* pada *game Honor of Kings*, memanfaatkan data telemetri internal yang kaya, termasuk fitur spasial, yang diperoleh langsung dari server permainan [11]. Pendekatan berbasis telemetri semacam ini menghasilkan fitur yang bersih dan terstruktur, namun bergantung pada akses langsung ke data internal *game* yang umumnya tidak tersedia untuk *game* pihak ketiga.

Pada ranah *fighting game* tradisional, Bannai dan Kajinami mengusulkan RIHBE (*Real-time Interpretable Health-Based Win Prediction Model*), yang mengekstraksi nilai *health point* (HP) dari HUD melalui analisis citra dan memprediksi pemenang menggunakan model *ensemble* yang menekankan interpretabilitas [12]. Pendekatan serupa juga diterapkan oleh Chulajata et al., yang menggunakan arsitektur LSTM dan *Transformer Encoder* untuk memprediksi hasil pertandingan *Street Fighter II Turbo* berdasarkan deret waktu persentase HP kedua pemain [13]. Kedua penelitian ini menjadikan HP sebagai sinyal utama progres pertandingan, karena *HP bar* merupakan indikator kondisi pertandingan yang secara eksplisit tersedia pada *fighting game* bergenre tradisional.

Penelitian lanjutan dari kelompok riset yang sama ArcadeViT oleh Chulajata et al. secara eksplisit mengidentifikasi keterbatasan sinyal HP tunggal, yaitu tidak merepresentasikan informasi posisi maupun pergerakan karakter, dan mengusulkan ekstraksi fitur visual langsung dari *frame* video menggunakan *Vision Transformer* [14]. Pergeseran pendekatan ini menunjukkan tren literatur yang bergerak dari sinyal skalar (HP) menuju fitur spasial/visual yang lebih kaya sebuah tren yang mendukung premis penelitian ini.

Tabel 2.1 merangkum perbandingan penelitian terdahulu tersebut dengan penelitian ini berdasarkan domain permainan, sinyal fitur yang digunakan, arsitektur model, serta sumber data.

Tabel 2.1. Perbandingan Penelitian Terdahulu

No	Penelitian	Domain	Sinyal Fitur	Arsitektur	Sumber Data
1	Bannai & Kajinami, RIHBE [12]	*Fighting game* tradisional	Nilai HP (skalar)	*Ensemble* (mengutamakan interpretabilitas)	Video : analisis citra
2	Chulajata dkk., LSTM/Transformer [13]	*Street Fighter II Turbo*	Deret waktu persentase HP (skalar)	LSTM & *Transformer Encoder*	Video : pembacaan *HP bar*
3	Chulajata dkk., ArcadeViT [14]	*Street Fighter II*	Piksel mentah *frame* (fitur spasial implisit)	*Vision Transformer*	*Frame* video mentah
4	Yang dkk., TSSTN [11]	*Honor of Kings* (MOBA)	Telemetri *real-time*, termasuk fitur spasial	*Two-stage spatial-temporal network*	Telemetri/API internal *game*
5	**Penelitian ini**	***Brawlhalla* (*platform fighter*)**	**Posisi 2D pemain relatif terhadap *stage*, terkoreksi *zoom/pan* kamera**	**YOLOv8 + CenterNet (ekstraksi) → GRU (prediksi)**	**Video : ekstraksi koordinat eksplisit**

Berdasarkan perbandingan pada Tabel 2.1, *Brawlhalla* sebagai *platform fighter* tidak memiliki *HP bar* sama sekali, kondisi kemenangan ditentukan oleh eliminasi *stock* melalui *knockback* ke luar arena, sehingga tidak tersedia sinyal skalar tunggal seperti HP yang dapat diekstraksi langsung. Selain itu, tidak tersedia akses telemetri internal *game* sebagaimana yang dimanfaatkan oleh TSSTN, sehingga satu-satunya sumber data yang dapat diakses adalah rekaman video pertandingan itu sendiri. Berbeda dengan ArcadeViT yang mempelajari representasi spasial secara implisit dari piksel mentah, penelitian ini melakukan ekstraksi fitur secara eksplisit dan terstruktur, posisi (x, y) pemain relatif terhadap *stage*, terkoreksi terhadap *zoom* dan *pan* kamera menggunakan YOLOv8 dan CenterNet, sebelum dimodelkan secara temporal menggunakan GRU. Pendekatan ini menghasilkan fitur masukan yang tetap dapat diinterpretasikan secara langsung, sekaligus menjangkau domain *platform fighter* yang belum banyak dieksplorasi dalam literatur prediksi kemenangan *esports* berbasis visual.

2.1 Esports

Berdasarkan tinjauan sistematis yang dilakukan oleh Formosa et al. (2022), esports didefinisikan sebagai aktivitas permainan video yang bersifat kompetitif dan terorganisir. Definisi ini mencakup ekosistem yang melibatkan pemain profesional maupun amatir, tim, penyelenggara turnamen, dan penonton. Perkembangan esports ditandai dengan profesionalisasi yang meliputi pelatihan terstruktur, sistem

liga, dan hadiah kompetisi yang signifikan[15].

Ekosistem esports dibangun atas interaksi antara pengembang game, penyelenggara turnamen, tim, pemain, dan penonton. Kompetisi dalam esports terbagi ke dalam berbagai genre, termasuk First-Person Shooter (FPS), Multiplayer Online Battle Arena (MOBA), dan fighting game [16]. Masing-masing genre memiliki karakteristik mekanik yang berbeda, sehingga pendekatan analisis yang diterapkan pun disesuaikan dengan kebutuhan spesifik setiap genre. Klasifikasi genre tersebut dilakukan berdasarkan elemen permainan yang dominan, seperti strategi tim, ketepatan aim, atau penguasaan ruang dan waktu dalam pertarungan individu [16].

Genre fighting game dipilih sebagai objek penelitian dalam tugas akhir ini karena karakteristik pertandingannya yang dinamis dan ketergantungan yang tinggi pada aspek spasial-temporal. Kemenangan dalam fighting game sering kali ditentukan oleh kemampuan pemain dalam mengontrol posisi, membaca pergerakan lawan, dan memanfaatkan ruang panggung secara efektif—faktor-faktor yang menjadikan genre ini kaya akan data spasial-temporal yang dapat diekstraksi dan dianalisis [16]. Penelitian ini bertujuan untuk mengekstraksi data pergerakan pemain dari rekaman pertandingan, yang kemudian dianalisis untuk mengidentifikasi pola-pola spasial-temporal yang berkontribusi terhadap hasil akhir pertandingan.

2.2 Fitur Spasial-Temporal dalam Pertandingan

Data spasial-temporal merupakan jenis data yang mengandung informasi lokasi (spasial) yang berubah terhadap waktu (temporal) [17]. Dalam konteks penelitian ini, fitur spasial merujuk pada posisi absolut pemain dalam bidang dua dimensi (koordinat x dan y) pada setiap frame, sedangkan aspek temporal merujuk pada urutan frame yang merepresentasikan lintasan pergerakan pemain dari waktu ke waktu.

Analisis spasial-temporal sangat krusial dalam game pertarungan karena keputusan strategis tidak dapat dipahami dari satu momen statis saja. Posisi pemain di suatu titik tertentu (misalnya di pojok kiri panggung) tidak memberikan informasi apakah ia sedang menyerang atau bertahan. Informasi tersebut baru terungkap ketika posisi tersebut diamati sebagai bagian dari lintasan yang kontinu, apakah pemain bergerak mendekati lawan (agresif) atau menjauh (defensif) [13]. Dengan demikian, representasi pergerakan pemain sebagai sekuens spasial-

temporal memungkinkan model untuk menangkap pola-pola seperti momentum, tekanan sudut (*corner pressure*), dan perubahan tempo pertarungan.

Pada penelitian ini, fitur spasial-temporal diekstraksi dari rekaman video pertandingan Brawlhalla melalui dua tahap. Tahap pertama adalah ekstraksi fitur spasial per frame menggunakan YOLOv8 untuk mendeteksi posisi pemain dan CenterNet untuk mendeteksi titik-titik referensi panggung. Tahap kedua adalah penyusunan fitur-fitur tersebut ke dalam jendela temporal berukuran 100 frame (setara dengan sekitar 1,5 detik). Jendela waktu ini dipilih untuk menangkap pola gerakan jangka pendek: seperti dash, lompatan, dan serangan. tanpa menambahkan kompleksitas komputasi yang tidak perlu dari sekuens yang lebih panjang. Pendekatan ini memungkinkan model untuk belajar dari dinamika pergerakan pemain secara utuh, bukan hanya dari posisi sesaat.

2.3 OCR pada Video Pertandingan

Optical Character Recognition (OCR) pada video merupakan tantangan yang signifikan karena teks dalam video hadir dengan latar belakang kompleks, variasi iluminasi, ukuran kecil, serta efek blur dan motion [18, 19]. Berbeda dengan OCR pada gambar statis, video OCR juga harus mempertimbangkan konsistensi temporal, karena teks yang sama mungkin muncul dalam beberapa frame dengan kualitas yang bervariasi. Untuk mengatasi tantangan tersebut, berbagai pendekatan modern menggabungkan segmentasi, attention-based OCR, dan pemrosesan region untuk meningkatkan ketahanan terhadap teks kecil dan buram [20]. Selain arsitektur model, pra-pemrosesan seperti konversi grayscale, reduksi noise, dan thresholding adaptif terbukti signifikan dalam mengoptimalkan akurasi pengenalan teks [21]. Dari berbagai toolkit OCR yang tersedia, penelitian ini menggunakan PaddleOCR 3.0 karena dirancang untuk menjaga akurasi sekaligus efisiensi inferensi, dengan model yang memiliki kurang dari 100 juta parameter namun mencapai performa kompetitif [22]. PaddleOCR juga menyediakan API modular yang memudahkan integrasi ke dalam pipeline yang lebih besar [22] dan memiliki adopsi luas di komunitas akademik maupun industri. Dalam konteks penelitian ini, PaddleOCR digunakan untuk membaca elemen-elemen HUD pada pertandingan Brawlhalla, seperti nama pemain, angka stock, dan marker teks singkat yang muncul secara konsisten di region tertentu pada setiap frame. Pemilihan ini didasarkan pada kebutuhan akan OCR yang cepat dan konsisten pada region teks yang spesifik, bukan untuk membaca teks umum di seluruh frame [23]. Hasil pembacaan OCR

kemudian diproses lebih lanjut untuk mengekstraksi informasi posisional yang menjadi dasar analisis spasial-temporal pada penelitian ini.

2.4 YOLO dan YOLOv8

YOLO (You Only Look Once) adalah keluarga model deteksi objek satu-langkah (one-stage detector) yang terkenal karena kombinasi kecepatan dan akurasi. Berbeda dengan detektor dua-langkah seperti Faster R-CNN yang memisahkan proses proposal region dan klasifikasi, YOLO memproses seluruh gambar dalam satu kali forward pass, menghasilkan prediksi bounding box dan kelas secara simultan [24]. Pendekatan ini menjadikan YOLO sebagai pilihan utama untuk sistem yang memerlukan latensi rendah tanpa mengorbankan akurasi deteksi.

Varian terbaru, YOLOv8, banyak digunakan dalam berbagai aplikasi analitik olahraga. Pada sepak bola, YOLOv8 yang dikombinasikan dengan DeepSORT mencapai mAP hingga 96% untuk deteksi pemain dan metrik MOTA 96% untuk pelacakan multi-objek [25]. Penelitian lain menunjukkan bahwa YOLOv8 dapat digunakan untuk mendeteksi pemain, wasit, dan bola, kemudian menghitung kecepatan dan jarak tempuh pemain hanya dari video siaran [26]. Studi komparatif juga membandingkan berbagai varian YOLOv8 dan algoritma pelacakan untuk mengekstraksi lintasan pemain dari kamera tunggal resolusi tinggi [**?**]. Pada olahraga indoor seperti futsal, YOLOv8 terbukti robust dalam mendeteksi pemain dan bola meskipun terjadi okklusi yang sering [27].

Berdasarkan performa dan fleksibilitas tersebut, YOLOv8n dipilih dalam penelitian ini sebagai detektor posisi karakter pada frame video pertandingan Brawlhalla. Pemilihan varian YOLOv8n didasarkan pada kebutuhan akan deteksi objek yang cepat dan efisien pada video dengan resolusi standar, di mana karakter pemain tampil dalam ukuran yang relatif kecil namun memiliki kontras yang cukup jelas terhadap latar belakang panggung.

2.4.1 Time Series Multivariat

Data yang diekstraksi per frame berupa posisi pemain, status off-screen, stock, dan zoom merupakan time series multivariat [28]. Pada data semacam ini, missing value, interval waktu, dan ketidakteraturan pengamatan sering memengaruhi kualitas pemodelan [29].

Tinjauan sistematis tentang irregular time series menekankan dua

pendekatan umum: imputasi pada tahap praproses atau modifikasi algoritma agar langsung menangani missing value dalam proses belajar [28]. Ini relevan untuk pipeline video, karena frame sampling, OCR failure, atau deteksi yang hilang dapat menghasilkan fitur temporal yang tidak selalu lengkap.

2.5 Gated Recurrent Unit (GRU)

Data pergerakan pemain yang diekstraksi dari setiap frame pertandingan membentuk deret waktu (time series) yang saling bergantung secara temporal. Untuk memproses data sekuensial tersebut, diperlukan arsitektur jaringan saraf yang mampu menangkap ketergantungan antar waktu. Recurrent Neural Network (RNN) merupakan arsitektur dasar yang dirancang untuk tujuan ini, karena kemampuannya dalam mempertahankan informasi dari langkah waktu sebelumnya melalui hidden state [29]. Namun, RNN standar menghadapi masalah vanishing gradient, yang menyebabkan gradien menyusut secara eksponensial selama propagasi mundur (backpropagation) dan mengakibatkan ketidakmampuan model dalam mempelajari dependensi jangka panjang [8, 30].

Untuk mengatasi keterbatasan tersebut, dikembangkan arsitektur gated seperti Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU), yang memperkenalkan mekanisme gerbang untuk mengontrol aliran informasi. GRU dipilih dalam penelitian ini karena arsitekturnya yang lebih sederhana dibandingkan LSTM: GRU hanya memiliki dua gerbang (update gate dan reset gate), sedangkan LSTM memiliki tiga gerbang (forget gate, input gate, output gate) [8, 30]. Kesederhanaan ini menghasilkan jumlah parameter yang lebih sedikit, sehingga GRU lebih efisien secara komputasi, lebih cepat dalam pelatihan, dan memiliki risiko overfitting yang lebih rendah pada data berukuran sedang [31, 32].

Pada penelitian ini, GRU menerima masukan berupa sekuens 100 frame (kurang lebih 1,5 detik) yang berisi fitur spasial-temporal posisi pemain. GRU memproses sekuens tersebut secara berurutan dan menghasilkan hidden state akhir yang merepresentasikan ringkasan pola pergerakan dalam jendela waktu tersebut. Hidden state akhir kemudian diteruskan ke lapisan fully-connected dengan aktivasi sigmoid untuk menghasilkan probabilitas kemenangan pemain pertama. Dengan demikian, GRU berfungsi sebagai ekstraktor fitur temporal yang menangkap pola-pola gerakan, seperti serangan mendadak, tekanan sudut (corner pressure), atau perilaku bertahan yang berkontribusi terhadap hasil akhir pertandingan [9, 32].

2.6 CenterNet

Pendekatan *anchor-free* juga berkembang dalam deteksi objek. Salah satu representasi utamanya adalah CenterNet, yang memodelkan setiap objek sebagai satu titik pusat (*center point*) dari kotak pembatasnya, alih-alih menggunakan banyak *anchor boxes* yang telah ditentukan sebelumnya [6]. Detektor ini memanfaatkan estimasi *keypoint* untuk menemukan titik pusat tersebut, kemudian secara langsung melakukan regresi terhadap properti objek lainnya, seperti ukuran kotak pembatas, lokasi 3D, orientasi, dan bahkan pose manusia [6]. Pendekatan berbasis titik pusat ini membuat CenterNet menjadi *end-to-end differentiable*, dengan arsitektur yang lebih sederhana dan lebih cepat dibandingkan detektor berbasis kotak pembatas konvensional, karena tidak memerlukan penentuan *anchor boxes* maupun *non-maximum suppression* (NMS) yang kompleks pada saat inferensi [6]. Pada dataset MS COCO, CenterNet menunjukkan *trade-off* kecepatan dan akurasi yang kompetitif, mencapai *Average Precision* (AP) 28,1% pada kecepatan 142 FPS dan 37,4% pada 52 FPS [6].

Dalam penelitian ini, CenterNet digunakan secara spesifik untuk mendeteksi titik-titik kunci panggung (*stage keypoints*) pada pertandingan Brawlhalla, yaitu *top_left*, *top_center*, dan *top_right*. Pemilihan CenterNet untuk tugas ini didasarkan pada dua pertimbangan utama. Pertama, arsitektur *anchor-free* CenterNet sangat cocok untuk deteksi objek berukuran kecil seperti titik panggung, yang sering kali hanya mencakup beberapa piksel pada frame video. Kedua, CenterNet menggunakan *output stride* sebesar 4, yang menghasilkan resolusi spasial lebih tinggi dibandingkan detektor berbasis *anchor* seperti YOLO (yang biasanya memiliki *output stride* 16 atau 32), sehingga memungkinkan lokalisasi titik yang lebih presisi [6]. Dengan mendeteksi ketiga titik kunci tersebut, posisi dan orientasi panggung dapat direkonstruksi secara geometris, yang kemudian digunakan untuk mengoreksi efek *zoom* dan pergeseran kamera, sehingga posisi pemain yang diperoleh menjadi stabil dan independen terhadap gerakan kamera. Dengan demikian, CenterNet dan YOLOv8 berperan secara komplementer: YOLOv8 mendeteksi pemain sebagai objek bergerak, sedangkan CenterNet mendeteksi titik-titik referensi statis panggung sebagai *keypoint*.

2.6.1 Anotasi Data untuk Deteksi Objek

Anotasi data (*data annotation*) merupakan proses pemberian label atau metadata pada data mentah, dalam hal ini gambar atau video. sehingga data tersebut memiliki makna yang dapat dipelajari oleh model *machine learning* secara terawasi (*supervised learning*) [24, 6]. Pada deteksi objek berbasis CNN, anotasi berfungsi sebagai *ground truth* yang menjadi acuan bagi model untuk mempelajari pola visual yang diinginkan. Dalam penelitian ini, anotasi digunakan secara eksklusif untuk melatih dua model deteksi: YOLOv8 untuk mendeteksi pemain, dan CenterNet untuk mendeteksi titik-titik kunci panggung.

A Anotasi Bounding Box untuk YOLO

YOLO (You Only Look Once) adalah *one-stage detector* yang memprediksi *bounding box* (kotak pembatas) di sekitar objek [24]. Format anotasi yang digunakan dalam pelatihan YOLO adalah:

$$\text{class\_id} \quad c_x \quad c_y \quad w \quad h \tag{2.1}$$

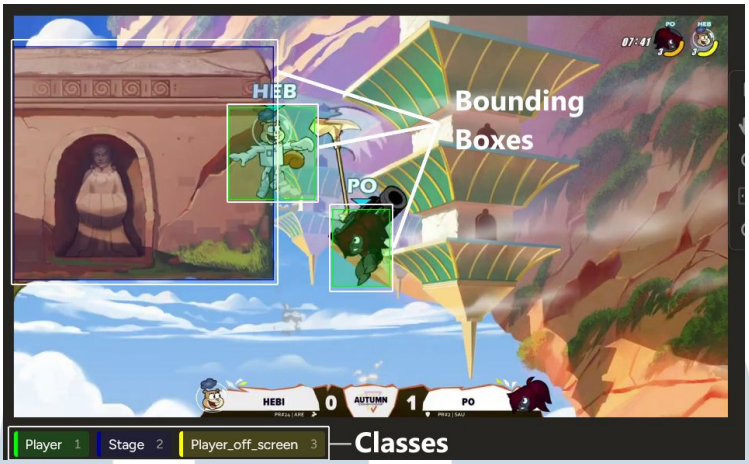
dengan:

- `class_id` : label kelas objek (misal: 0 untuk pemain 1, 1 untuk pemain 2)
- c_x, c_y : koordinat pusat *bounding box* yang dinormalisasi terhadap lebar dan tinggi gambar (rentang 0–1)
- w, h : lebar dan tinggi *bounding box* yang dinormalisasi

Sebagai contoh, anotasi untuk dua pemain pada frame Brawlhalla dapat dituliskan sebagai berikut:

```
0 0.45 0.50 0.10 0.20
1 0.60 0.48 0.10 0.20
```

Proses anotasi ini dilakukan dengan menggambar kotak di sekitar setiap pemain pada setiap frame (atau subset frame), sehingga YOLO dapat mempelajari hubungan antara fitur visual dan posisi objek. Gambar 2.1 menunjukkan contoh hasil anotasi *bounding box* tersebut pada sebuah *frame* video pertandingan.



Gambar 2.1. Contoh anotasi *bounding box* untuk deteksi pemain

B Anotasi Keypoint untuk CenterNet

CenterNet adalah arsitektur *anchor-free* yang memprediksi objek sebagai titik pusat (*center point*) pada *heatmap*, kemudian meregresi properti objek dari titik tersebut [6]. Format anotasi yang digunakan adalah:

$$\text{class\_id} \quad c_x \quad c_y \tag{2.2}$$

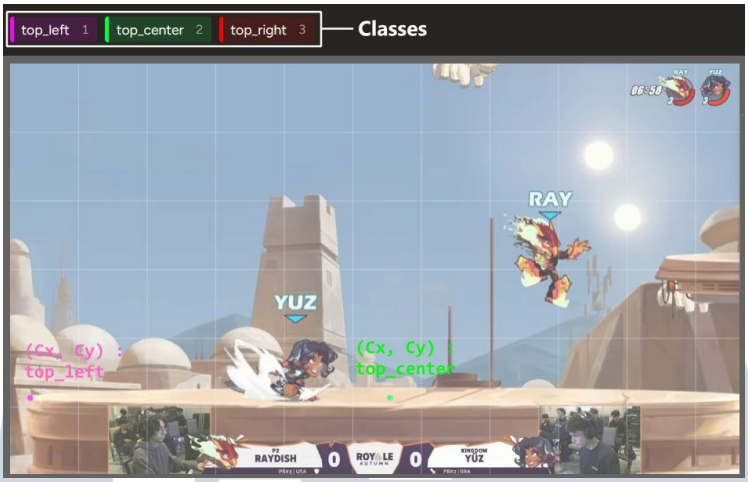
dengan:

- `class_id` : label kelas titik kunci (misal: 0 untuk *top_center*, 1 untuk *top_left*, 2 untuk *top_right*)
- c_x, c_y : koordinat titik pusat yang dinormalisasi

Contoh anotasi untuk tiga titik panggung Brawlhalla:

```
0 0.50 0.30
1 0.20 0.30
2 0.80 0.30
```

Proses anotasi CenterNet lebih ringan dibandingkan YOLO, karena anotator cukup menandai titik pusat objek, tanpa perlu menggambar kotak. Titik-titik ini kemudian dikonversi menjadi *heatmap* Gaussian yang menjadi target pembelajaran model. Gambar 2.2 menunjukkan contoh penempatan ketiga titik kunci tersebut pada *stage* dimana `classes` ada label dan `Cx` serta `Cy` adalah koordinat titik.



Gambar 2.2. Contoh anotasi *keypoint* untuk deteksi titik panggung

C Perbandingan Metode Anotasi

Perbedaan antara kedua pendekatan anotasi dirangkum dalam Tabel 2.2.

Tabel 2.2. Perbandingan anotasi YOLO dan CenterNet

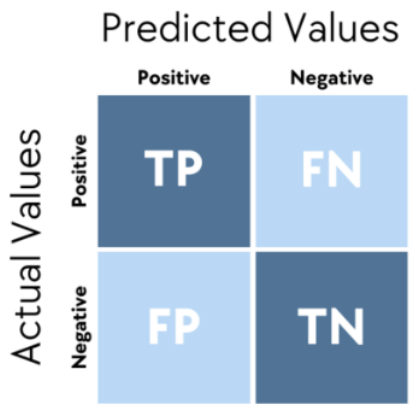
Aspek	**YOLO**	**CenterNet**
Jenis anotasi	Bounding box	Keypoint (titik)
Format anotasi	`class, cx, cy, w, h`	`class, cx, cy`
Tingkat detail	Tinggi (ukuran objek)	Rendah (hanya titik)
Kecepatan anotasi	Lambat	Cepat
Penggunaan dalam penelitian	Deteksi pemain	Deteksi titik panggung

2.7 Confusion Matrix

Confusion Matrix merupakan metode evaluasi yang digunakan untuk mengukur kinerja model klasifikasi dengan membandingkan hasil prediksi model terhadap label aktual. Matriks ini menyajikan jumlah prediksi yang benar maupun salah ke dalam empat kategori utama, yaitu **True Positive (TP)**, **True Negative (TN)**, **False Positive (FP)**, dan **False Negative (FN)**.

True Positive (TP) menunjukkan jumlah data positif yang berhasil diprediksi positif oleh model, sedangkan **True Negative (TN)** menunjukkan jumlah data negatif yang berhasil diprediksi negatif. Sebaliknya, **False Positive (FP)** merupakan kondisi ketika data negatif diprediksi sebagai positif, dan **False Negative (FN)** terjadi ketika data positif diprediksi sebagai negatif.

Penggunaan Confusion Matrix penting untuk mengevaluasi performa model, khususnya pada klasifikasi tidak seimbang (*imbalanced dataset*). Berdasarkan nilai True Positive, True Negative, False Positive, dan False Negative, metrik evaluasi model dapat dihitung sebagaimana ditunjukkan pada Gambar 4.1 berikut.



Gambar 2.3. Confusion Matrix

Berikut merupakan formulasi metrik evaluasi yang digunakan untuk mengukur performa model klasifikasi berdasarkan hasil perhitungan pada Confusion Matrix. Adapun metrik evaluasi yang digunakan meliputi **Accuracy**, **Precision**, **Recall**, dan **F1-Score** sebagai berikut:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.3}$$

Rumus 2.3 digunakan untuk menghitung tingkat akurasi keseluruhan model dalam mengklasifikasikan data secara benar, baik deteksi objek maupun prediksi *timestep*. Nilai accuracy yang tinggi menunjukkan bahwa model memiliki performa yang baik dalam melakukan klasifikasi secara umum.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2.4}$$

Precision pada rumus 2.4 digunakan untuk mengukur tingkat ketepatan model dalam mendeteksi objek dari seluruh prediksi yang diklasifikasikan sebagai fraud. Semakin tinggi nilai precision, semakin kecil kemungkinan model menghasilkan kesalahan klasifikasi positif palsu (*false positive*).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.5}$$

Recall pada rumus 2.5 digunakan untuk mengukur kemampuan model dalam mendeteksi seluruh data fraud yang sebenarnya terdapat dalam dataset. Nilai recall yang tinggi menunjukkan bahwa model mampu meminimalkan kesalahan negatif palsu (*false negative*).

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.6}$$

Nilai F1-Score pada rumus 2.6 digunakan untuk mengukur keseimbangan antara precision dan recall, terutama pada dataset yang memiliki distribusi kelas tidak seimbang. Metrik ini penting digunakan ketika diperlukan *trade-off* antara kemampuan mendeteksi fraud dan meminimalkan kesalahan prediksi.