

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Pengumpulan data adalah tahap awal yang sangat krusial dalam pengembangan model. Data yang digunakan dalam penelitian ini berasal dari dataset yang dibuat oleh roboflow yang dapat diakses melalui link berikut <https://universe.roboflow.com/roboflow-jvuqo/football-players-detection-3zvbc/dataset/2>. Dataset yang digunakan dalam penelitian ini terdiri dari 312 klip gambar pertandingan sepak bola yang telah memiliki anotasi. Anotasi mencakup empat kelas utama, yaitu pemain (*players*), penjaga gawang (*goalkeeper*), wasit (*referees*), dan bola (*ball*). Pilihan kelas ini sejalan dengan tujuan penelitian yang berfokus pada perhitungan penguasaan bola (*ball possession*) pada pertandingan sepak bola. Untuk mendukung tujuan tersebut, dataset telah mengalami proses augmentasi data yang akan dijelaskan pada tahap pelatihan Model YOLOv8. Pembagian proporsi antar data *train*, *valid*, dan *test* sebesar 80%, 14%, 6%. Besaran persentase pembagian dataset didasari akan sedikitnya gambar, sehingga diperlukannya data latih yang sangat dominan besarnya yakni 80%. Selanjutnya sampel gambar pada dataset ditunjukkan pada Gambar 4.1 di bawah ini.



Gambar 4.1 Contoh Gambar Pada Dataset

4.2 *Preprocessing* & Pelatihan Model

Setelah pengumpulan data selesai, tahap berikutnya adalah pelatihan model YOLOv8. Pada penelitian yang dilakukan kali ini penulis menggunakan model YOLOv8m atau medium. Model YOLOv8m digunakan karena kemampuannya

dalam mendeteksi objek dengan akurasi tinggi dan kecepatan yang efisien. Proses pelatihan dimulai dengan pembangunan arsitektur model, yang kemudian dilatih menggunakan data latih yang telah dikumpulkan. Model dilatih menggunakan jumlah epoch 100, *batch size* 8 dan *learning rate* sebesar 0.0001. Terdapat beberapa eksperimen pemodelan yang dilakukan pada tahap pemodelan, antara lain preprocessing data, dan augmentasi data. Beberapa eksperimen yang disebutkan diperuntukan untuk meningkatkan kualitas dari model yang dibuat.

4.2.1 Data Preprocessing

Preprocessing dilakukan untuk mengkonsistenkan bentuk data yang akan dilakukan pelatihan. Proses ini penting untuk memastikan bahwa data yang masuk ke model memiliki format yang konsisten dan sesuai dengan kebutuhan model. Pada penelitian ini dilakukan penyesuaian ukuran gambar atau resizing dengan ukuran dimensi 640x640 pixel. Besaran ukuran dimensi ini merujuk pada penelitian sebelumnya tentang. Hal ini dilakukan untuk memastikan semua gambar memiliki ukuran yang sama sebelum masuk ke model, sehingga model dapat memprosesnya secara efisien, kemudian hasil *preprocessing* ditunjukkan pada Gambar 4.2 dan 4.3.



Gambar 4.2 Gambar Sebelum *Preprocessing*



Gambar 4.3 Gambar Setelah *Preprocessing*

4.2.2 Data Augmentasi

Hal ini perlu dilakukan karena dataset yang dipakai cenderung tidak banyak sehingga dengan adanya augmentasi membuat data itu semakin bertambah. Penelitian yang dilakukan kali ini augmentasi datanya berupa melakukan pembalikan gambar secara horizontal, sehingga akan meningkatkan generalisasi model dan mencegah *overfitting*. Proses data augmentasi hanya digunakan pada gambar training, karena tujuan utama dari data validasi dan pengujian adalah untuk mengevaluasi kinerja model pada data yang tidak terlihat selama pelatihan. Dengan melakukan augmentasi pada data validasi atau pengujian akan mengubah distribusi asli data tersebut. Hal ini dapat mengakibatkan evaluasi yang tidak akurat tentang kinerja model. Oleh karena itu, sangat penting untuk menjaga data validasi dan pengujian tetap representatif dari data yang akan dihadapi model. Hasil augmentasi dapat dilihat pada Gambar 4.4 dan 4.5.



Gambar 4.4 Gambar Sebelum Augmentasi



Gambar 4.5 Gambar Setelah Augmentasi

4.3 Evaluasi Model I

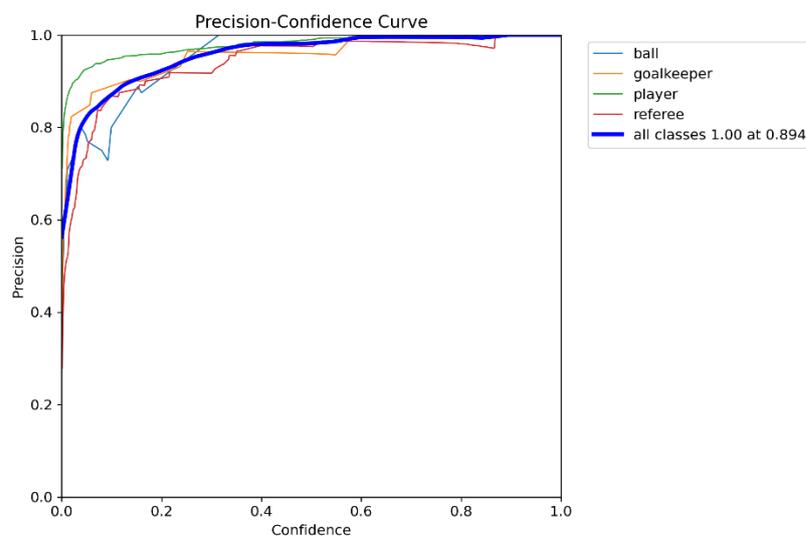
Evaluasi model dilakukan untuk mengevaluasi performa model deteksi objek dan pelacakan yang telah dikembangkan. Data validasi yang telah disiapkan

digunakan untuk menguji performa *object detection*. Metode evaluasi yang digunakan antara lain adalah perhitungan matrix *precision*, *recall*, dan *mAP*. evaluasi dilakukan dengan menguji model dengan data validasi. Berikut adalah hasil dari model yang dibangun ditunjukkan pada Tabel 4.1.

Tabel 4.1
Hasil Pelatihan Model 100 Epoch

Model	Precision	Recall	mAP
YOLOv8m (100 epoch)	88%	73.3%	82.3%

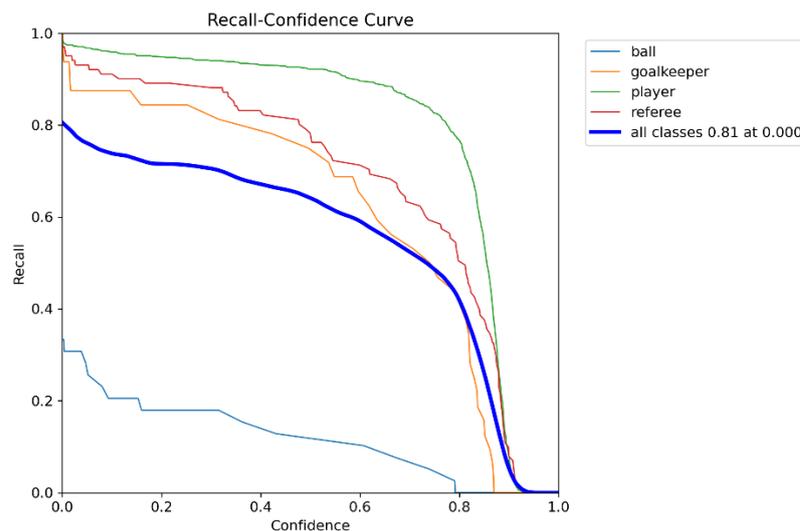
Berdasarkan Tabel 4.1 menghasilkan nilai *precision*, *recall*, dan *mAP* seperti yang dicantumkan pada tabel diatas. Nilai *precision* yang diraih model cukup tinggi yang mengindikasikan bahwa model baik dalam menghindari deteksi yang salah (*false positive*) atau model sangat baik dalam mendeteksi objek. Nilai *recall* yang diraih mengidikasikan bahwa model masih terdapat kekurangan yakni terdapat objek yang terlewat dan tidak terdeteksi. Hal ini terjadi karena pada pertandingan sepak bola mungkin ada beberapa objek yang saling menghalangi atau tumpang tindih. Nilai *mAP* yang diraih cukup baik sehingga model mampu medeteksi berbagai jenis objek dengan akurasi yang cukup tinggi. Grafik hasil pelatihan *precision* dan *recall* ditunjukkan pada Gambar 4.6 dan 4.7.



Gambar 4.6 Grafik *Precision* Model Epoch 100

Berdasarkan Gambar 4.6, menunjukkan hasil *precision* model berdasarkan performa per Kelas:

- *Ball*: Kurva *ball* cenderung lebih tinggi dan stabil dibandingkan kelas lainnya, menunjukkan bahwa model sangat baik dalam mengenali bola.
- *Referee*: Kurva pada *referee* juga memiliki kecenderungan yang tinggi walaupun tidak setinggi pada *ball* yang membuat model ini mampu mengenali referee dengan baik.
- *Goalkeeper* dan *Player*: kurva untuk kedua kelas ini memiliki bentuk yang mirip, menunjukkan bahwa model mengalami kesulitan membedakan antara keduanya, terutama pada tingkat kepercayaan yang lebih rendah.



Gambar 4.7 Grafik *Recall* Model *Epoch* 100

Berdasarkan Gambar 4.7 menunjukkan hasil *recall* model berdasarkan performa per Kelas:

- *Ball*: Kurva *ball* cenderung sangat rendah dibandingkan dengan kelas lainnya, hal ini mengakibatkan model sering kali terlewat dalam melakukan deteksi pada kelas ini.
- *Referee* dan *Goalkeeper*: Kurva pada *referee* dan *goalkeeper* hampir sama yang mana model mampu mengenali objek dengan baik sehingga sedikit kemungkinan model terlewat dalam melakukan deteksi.
- *Player*: kurva untuk *player* sangatlah tinggi sehingga dapat disimpulkan bahwa model sangat baik dalam mengenali kelas ini.

4.4 Hyperparameter Tuning

Hyperparameter tuning dilakukan untuk mencari performa model terbaik dalam melakukan deteksi pada pertandingan sepak bola. Berikut hasil evaluasi dari *hyperparameter tuning* yang telah dilakukan ditunjukkan pada Tabel 4.2.

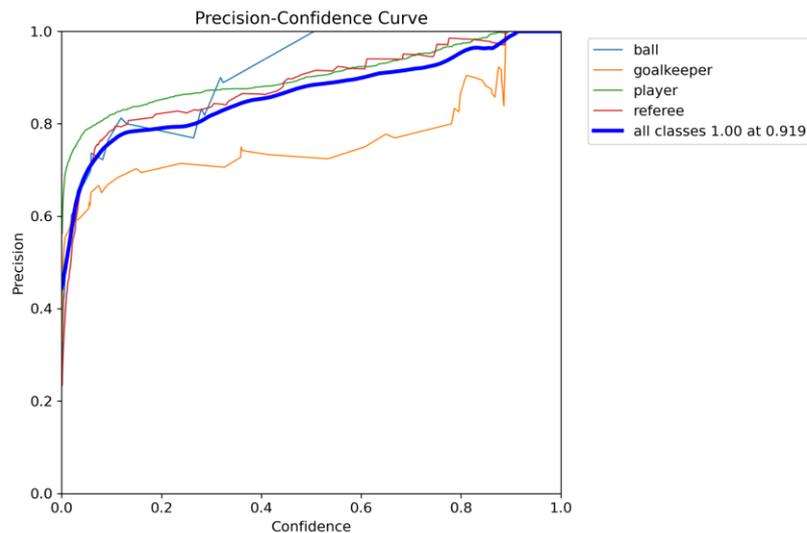
Tabel 4.2

Hasil Evaluasi Model Dengan *Hypaeparametr Tuning*

Model	Precision	Recall	mAP
YOLOv8m (100 epoch)	88%	73.3%	82.3%
YOLOv8m (150 epoch)	91.50%	73.60%	82.70%
Yolov8m(300 epoch)	90.10%	74.10%	81.40%

Pada tahap ini dilakukan 6 kali pengujian yang mana menggunakan 3 batasan *epoch* berbeda, sehingga untuk tiap *epoch*-nya dilakukan 2 kali pengujian untuk benar-benar memastikan hasil yang didapat. Berdasarkan Tabel 4.2, dari hasil evaluasi setelah dilakukan *hyperparameter tuning* dapat disimpulkan bahwa model deteksi objek YOLOv8m menunjukkan peningkatan kinerja seiring bertambahnya jumlah *epoch* pelatihan, namun mengalami penurunan ketika dilakukan pada epoch ke 300, hal terindikasi mengalami overfitting.

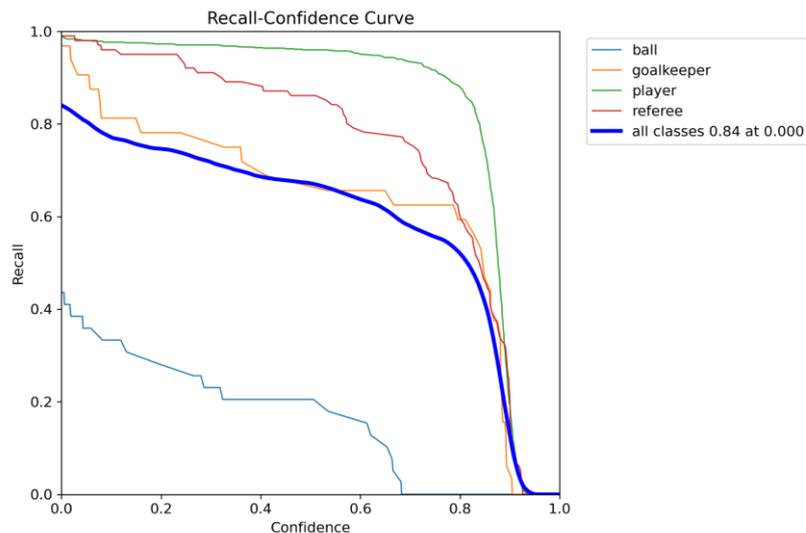
Peningkatan paling signifikan terlihat pada metrik *precision*, yang mengindikasikan bahwa model menjadi semakin baik dalam mendeteksi objek secara akurat dan lengkap. Pilihan model dengan 150 *epoch* terindikasi sebagai pilihan yang paling optimal berdasarkan hasil evaluasi. Hal ini disebabkan oleh nilai *mAP* dan *precision* yang tertinggi dari pada *epoch* lainnya. Artinya, model dengan 150 *epoch* mampu mencapai keseimbangan terbaik antara kemampuan mendeteksi objek yang benar dan kemampuan menemukan semua objek yang ada. Serta terdapat pertimbangan bahwa akan adanya diminishing return atau overfitting jika dilakukan pelatihan yang dilakukan lebih lama. Berikut adalah gambaran grafik yang dibandingkan dengan nilai *confidence curve* mengenai hasil evaluasi yang dilakukan menggunakan YOLOv8m 150 *epoch* yang ditunjukkan pada Gambar 4.8.



Gambar 4.8 Grafik *Precision* Setelah *Hyperparameter Tuning*

Berdasarkan Gambar 4.8 menunjukkan hasil grafik *precision-confidence curve* yang komprehensif tentang kekuatan dan kelemahan model deteksi objek. Model ini sangat baik dalam mendeteksi bola, tetapi mungkin memerlukan perbaikan lebih lanjut untuk meningkatkan akurasi deteksi pada kelas penjaga gawang, pemain, dan wasit, terutama pada *confidence threshold* yang lebih rendah. Kemudian jika dilihat performa per Kelas berdasarkan *precision* adalah:

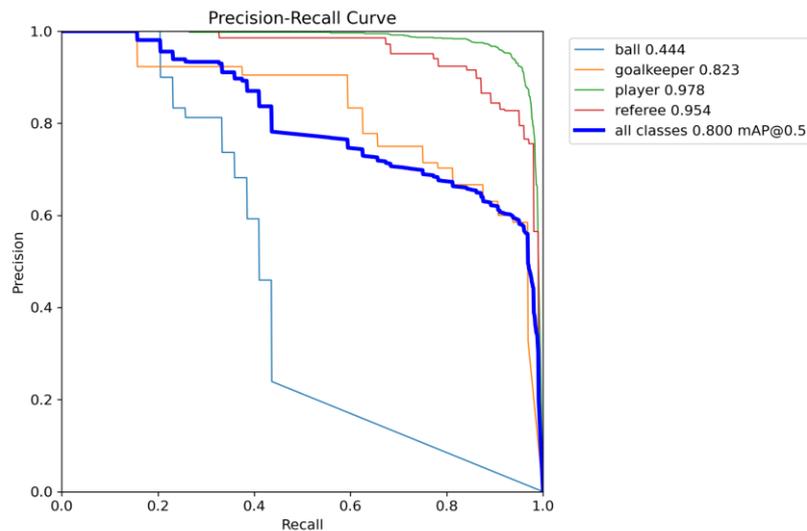
- **Ball:** Kurva *precision* untuk kelas ini terlihat cukup stabil dan menunjukkan nilai *precision* yang tinggi sepanjang interval *confidence*. Ini menunjukkan bahwa model dapat dengan baik mengidentifikasi bola.
- **Goalkeeper:** Kurva untuk kelas ini lebih berfluktuasi, terutama di nilai *confidence* yang lebih tinggi. Fluktuasi ini mungkin menunjukkan adanya ketidakstabilan model dalam mengidentifikasi kiper dengan konsisten.
- **Player:** Kurva ini juga cukup stabil dan hampir mencapai *precision* maksimum, menunjukkan kinerja yang baik untuk deteksi pemain.
- **Referee:** Kurva ini memiliki nilai *precision* yang lebih rendah dibandingkan kelas lain, terutama di *confidence* yang lebih tinggi. Hal ini menunjukkan bahwa model mengalami kesulitan dalam mendeteksi wasit dengan akurasi tinggi.



Gambar 4.9 Grafik *Recall* Setelah *Hyperparameter Tuning*

Berdasarkan Gambar 4.9 menunjukkan menunjukkan bahwa semua kelas memiliki nilai *recall* yang cukup baik kecuali bola dengan perbedaan nilai yang sangat signifikan. Kemudian jika dilihat performa per Kelas berdasarkan *recall* adalah:

- **Ball:** Kurva *recall* untuk kelas bola cukup rendah di sebagian besar tingkat kepercayaan. Ini menunjukkan bahwa model mungkin tidak mendeteksi banyak bola, atau bahwa model perlu menurunkan threshold kepercayaan untuk meningkatkan *recall*.
- **Goalkeeper:** Kurva untuk kelas ini cukup stabil tetapi menunjukkan penurunan *recall* yang cukup signifikan saat kepercayaan meningkat. Ini menunjukkan bahwa ketika model sangat yakin, seringkali gagal mendeteksi penjaga gawang yang benar.
- **Player:** Kurva untuk pemain menunjukkan nilai *recall* yang tinggi dan cukup stabil, yang menunjukkan bahwa model bekerja dengan baik dalam mendeteksi pemain.
- **Referee:** Kurva *recall* untuk wasit juga cukup tinggi, tetapi sedikit lebih rendah dari pemain, menunjukkan deteksi yang cukup baik tetapi mungkin bisa ditingkatkan lebih lanjut.



Gambar 4.10 Grafik *Precision-Recall* Setelah *Hyperparameter Tuning*

Berdasarkan Gambar 4.10 diatas menunjukkan bahwa kurva PR memiliki hubungan antara *precision* dan *recall* pada berbagai ambang kepercayaan. Setiap titik pada kurva ini menunjukkan pasangan nilai *precision* dan *recall* yang dihasilkan oleh model pada tingkat confidence tertentu. Kemudian jika dilihat performa per Kelas berdasarkan kurva *precision-recall* adalah:

- **Player:** Kurva ini sangat baik, dengan *precision* dan *recall* yang mendekati 1 di sebagian besar titik. Ini menunjukkan bahwa model sangat baik dalam mendeteksi objek.
- **Referee:** Kurva ini juga menunjukkan kinerja yang baik, dengan *precision* dan *recall* yang tinggi hampir di seluruh rentang. Ini berarti model juga sangat andal dalam mendeteksi wasit.
- **Goalkeeper:** Kurva ini memiliki *precision* dan *recall* yang lebih rendah dibandingkan player dan referee, tetapi masih cukup baik. Ini menunjukkan bahwa model masih cukup baik dalam mendeteksi kiper, meskipun ada beberapa prediksi yang salah atau tidak terdeteksi.
- **Ball:** Kurva ini menunjukkan bahwa model kesulitan dalam mendeteksi bola dengan baik. *Precision* turun drastis ketika *recall* meningkat, menunjukkan bahwa model sering kali mendeteksi bola secara salah atau gagal mendeteksinya sama sekali.

Setelah melakukan evaluasi model yang sudah dilakukan hyperparameter tuning dengan data validasi maka selanjutnya untuk memastikan model benar-benar

baik adalah dengan melakukan pengujian pada data tes. Hal ini dilakukan karena data tes sama sekali belum pernah dilihat oleh model sehingga tidak akan terjadi adanya bias pada model. Berikut merupakan hasil pengujian model dengan data tes.

Tabel 4.3
Hasil Evaluasi Akhir Model

Model	Precision	Recall	mAP
YOLOv8m (150 epoch)	92.8%	75.3%	81.6%

Dengan melihat dari Tabel 4.3 diatas dapat disimpulkan bahwa model mampu berjalan dengan baik sebab model cenderung sedikit membuat kesalahan dalam mengklasifikasikan suatu objek dan mampu mengenal objek dengan cukup baik. Kemudian, *output* model terhadap gambar adalah pengelompokkan pada objek dengan menampilkan label deteksi yang ditunjukkan pada Gambar 4.11.

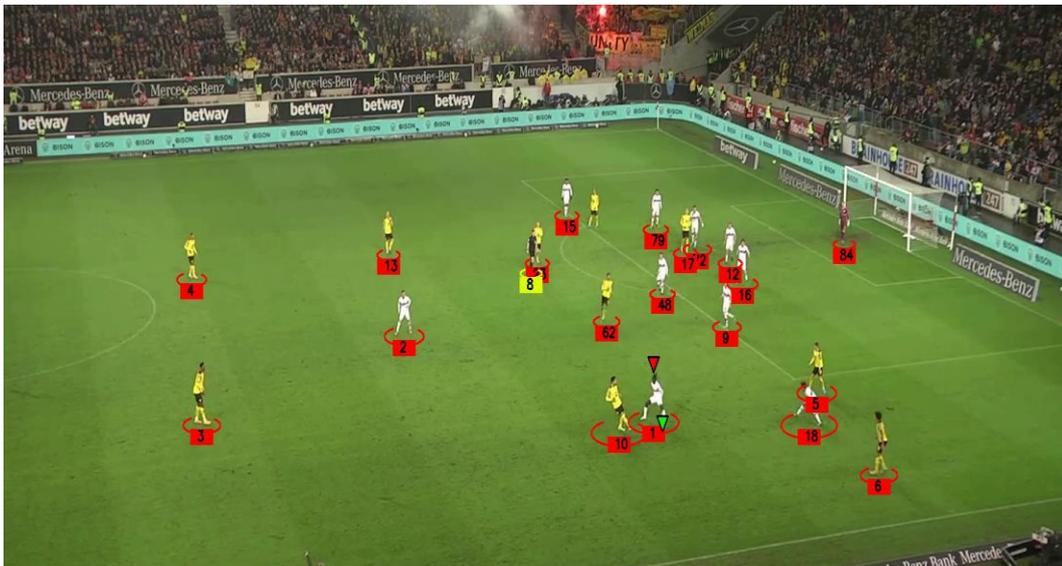


Gambar 4.11 Output Model

4.5 Implementasi Object Tracking

Setelah mendapatkan model terbaik selanjutnya dilakukan penerapan *object tracking* menggunakan *ByteTrack* yang mana bertujuan untuk melacak pergerakan objek, khususnya pemain sepak bola, dari satu *frame* ke *frame* berikutnya. Dengan melacak objek secara kontinu, maka akan memperoleh informasi yang lebih akurat mengenai posisi pemain dan sesuai dengan penelitian ini yaitu menghitung *ball*

possession antar tim maka akan membantu dalam membedakan pemain antar tim. *ByteTrack* membantu mengatasi tantangan seperti oklusi (objek terhalang), sehingga tracking dapat dilakukan secara lebih lancar. *ByteTrack* berperan penting dalam pengimplementasian ini karena mampu memberikan identitas unik setiap objek yang terdeteksi dan melacak pergerakannya sepanjang jalannya video. Selain itu pada tahap ini dilakukan juga penyesuaian tampilan *bounding box* agar hasil tracking lebih mudah untuk diinterpretasikan.



Gambar 4.12 Contoh Implementasi *ByteTrack*

Berdasarkan Gambar 4.12 menunjukkan bahwa pada implementasinya tiap objek diberikan identitas berupa id unik, hal ini penting dilakukan sebab dengan pemberian id ini model akan mampu menganal pemain mana yang terakhir menyentuh bola sehingga dapat menentukan dengan lebih akurat tim mana yang sedang menguasai bola. Setelah pemebrian identitas unik pada tiap pemain model diolah lebih lanjut pada tahap selanjutnya. Pada tahap selanjutnya, pemain diklasifikasikan ke dalam tim yang berbeda berdasarkan warna seragam mereka. Selain itu, posisi bola pada beberapa *frame* yang tidak terdeteksi diperkirakan menggunakan metode interpolasi serta penentuan pemain yang menguasai bola.

4.5.1 Interpolasi Posisi Bola

Pada tahap ini, dilakukan interpolasi posisi bola untuk mengatasi permasalahan hilangnya deteksi bola pada beberapa *frame* video. Hal ini terjadi sebab kelas bola memiliki skala ukuran yang kecil dibandingkan class lainnya.

Interpolasi merupakan teknik untuk memperkirakan nilai data yang hilang berdasarkan data yang ada. Dalam konteks ini, posisi bola pada *frame* yang memiliki deteksi digunakan sebagai dasar untuk memperkirakan posisi bola pada *frame* yang tidak memiliki deteksi. Proses interpolasi dilakukan dengan menggunakan *library* Pandas untuk membentuk *Dataframe* yang berisi koordinat *bounding box* bola pada setiap *frame*. Kemudian, nilai yang hilang pada *Dataframe* tersebut diisi menggunakan metode interpolasi dan pengisian nilai terakhir (*backfill*) serta nilai awal (*forward fill*). Dengan demikian, diperoleh data posisi bola yang lebih lengkap dan kontinu, sehingga dapat digunakan untuk analisis pergerakan bola yang lebih akurat. Hasil sebelum dan sesudah dilakukan interpolasi model terhadap posisi bola ditunjukkan pada Gambar 4.13 dan 4.14.



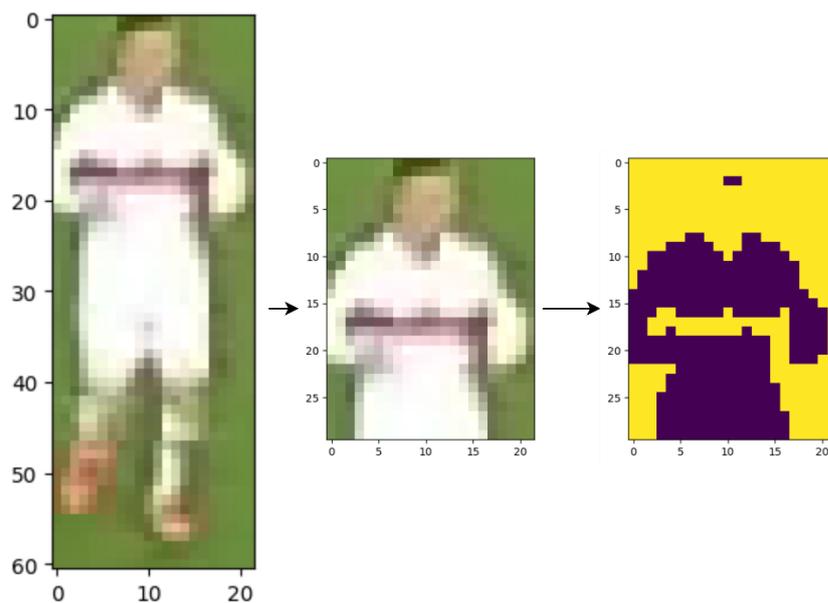
Gambar 4.13 Gambar Sebelum Interpolasi



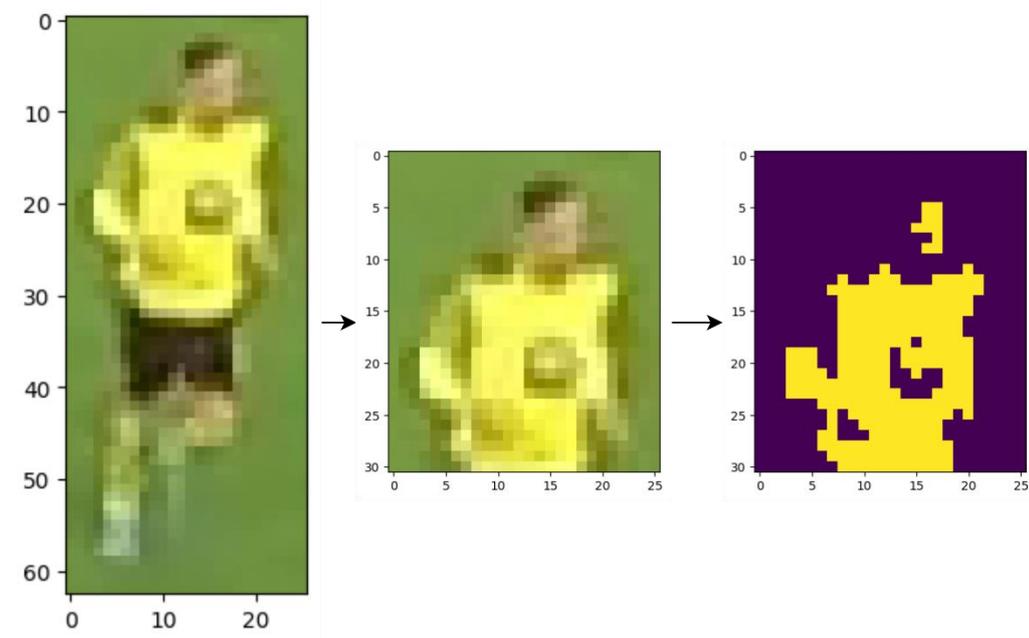
Gambar 4.14 Gambar Sebelum Interpolasi

4.5.2 Penentuan Tim Pemain

Dalam melakukan pengelompokan tim pada penelitian ini menggunakan metode *K-means clustering*. Algoritma ini mampu untuk secara otomatis mengklasifikasikan pemain ke dalam tim yang benar berdasarkan warna seragam mereka. Metode ini dipilih karena kemampuannya dalam mengelompokkan data secara efisien dan efektif. Prosesnya dimulai dengan mengambil sampel warna dari bagian atas seragam pemain. Sampel warna ini kemudian dikelompokkan menjadi dua cluster utama menggunakan algoritma *K-Means*. Selanjutnya, dilakukan analisis spasial untuk mengidentifikasi cluster yang mewakili warna latar belakang. *Cluster* yang tersisa dianggap sebagai warna tim. Dengan membandingkan warna seragam setiap pemain dengan warna tim yang telah ditentukan, maka setiap pemain dapat diklasifikasikan ke dalam tim yang sesuai. Metode ini memanfaatkan kesederhanaan *K-Means* dalam mengelompokkan data dan keefektifan analisis spasial dalam mengidentifikasi warna latar belakang. Hasil proses klasifikasi tim ditampilkan pada Gambar 4.15 dan 4.16 yang menunjukkan perbedaan warna pada proses klasifikasi oleh model.



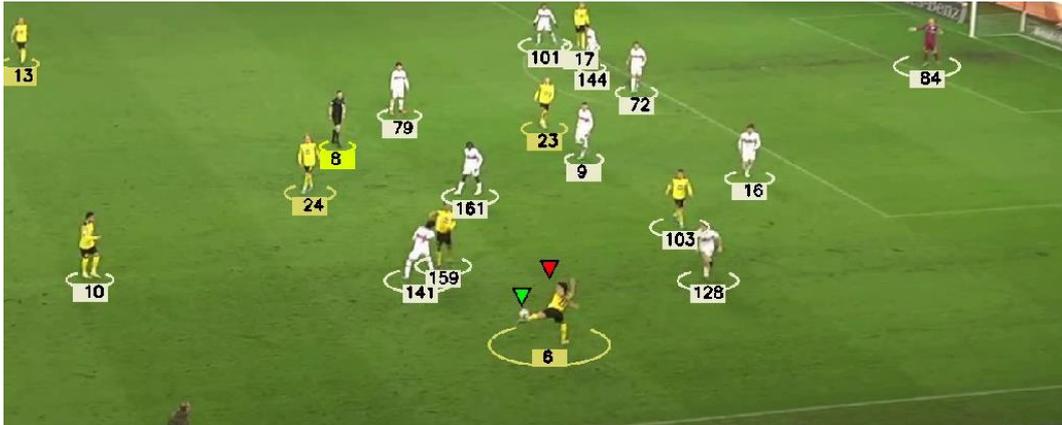
Gambar 4.15 Proses Klasifikasi Tim I



Gambar 4.16 Proses Klasifikasi Tim II

4.5.3 Penentuan *Ball Possession*

Setelah melakukan penentuan tim maka selanjutnya adalah menentukan penguasaan bola. Dalam penelitian ini untuk menentukan tim mana yang sedang memegang bola dilakukan dengan pemanfaatan rumus *euclidian distance*, yang mana rumus itu akan menghitung objek yang mana yang paling dekat dengan bola. Untuk meningkatkan akurasi dan menghindari deteksi yang salah, pada penelitian ini menetapkan ambang batas jarak maksimum sebesar 40 piksel atau setara dengan 1,05833 cm, penentuan besaran maksimum jarak ini merujuk pada penelitian sebelumnya. Dengan demikian, sistem akan menghitung jarak *Euclidean* antara pusat (*centroid*) bola dengan pusat setiap pemain. Pemain yang memiliki jarak terdekat dengan bola dan kurang dari ambang batas yang telah ditentukan akan dianggap sebagai pemegang bola. Proses ini melibatkan klasifikasi objek *real-time* menjadi dua kelas utama, yaitu bola dan pemain, sebelum perhitungan jarak dilakukan seperti yang ditampilkan pada Gambar 4.17 .



Gambar 4.17 Proses Pelabelan Bola dan Pemain

4.6 Evaluasi Model II

Setelah melalui tahap implementasi objek tracking, interpolasi posisi bola, penentuan tim pemain, dan penentuan penguasaan bola, langkah selanjutnya adalah melakukan evaluasi terhadap model yang telah dibangun. Evaluasi ini bertujuan untuk mengukur kinerja model dalam melacak objek-objek yang ada. Dalam evaluasi kali ini menggunakan matrik *precision*, *recall*, *mAP* dan *MOTA*. Hasil evaluasi diperoleh dengan membandingkan hasil prediksi model dengan data *ground truth* yang telah disiapkan sebelumnya.

Tabel 4.4
Hasil Evaluasi Model Implementasi Object Tracking

<i>Precision</i>	<i>Recall</i>	<i>mAP</i>	<i>MOTA</i>
96.47%	84.64%	87.50%	81.42%

Berdasarkan Tabel 4.4 di atas menunjukkan hasil evaluasi yang telah didapatkan. Implementasi *object tracking* pada model telah berhasil meningkatkan kinerja pelacakan objek, terutama pada nilai *recall*. Hal ini menunjukkan keberhasilan interpolasi dalam mengatasi masalah *under-detection* pada kelas tersebut. Kenaikan nilai *mAP* juga mengindikasikan peningkatan kinerja model secara keseluruhan. Nilai *MOTA* sebesar 81.42% menunjukkan bahwa model mampu melacak multi-objek dengan akurasi yang cukup baik.



Gambar 4.18 Hasil Akhir Output Model

Gambar 4.18 di atas menunjukkan hasil akhir dari penelitian yang telah dilakukan, di mana model yang dikembangkan berhasil melakukan deteksi objek dan perhitungan *ball possession* (penguasaan bola) dalam sebuah pertandingan sepak bola. Dalam gambar tersebut, dapat dilihat visualisasi dari pemain-pemain kedua tim yang sedang bertanding di lapangan, dengan masing-masing pemain diberi penanda identitas atau nomor unik. Gambar ini tidak hanya menunjukkan penguasaan bola secara keseluruhan, tetapi juga memberikan informasi mengenai posisi pemain di lapangan pada momen tertentu dalam pertandingan. Secara keseluruhan, hasil ini menunjukkan bahwa model yang dikembangkan tidak hanya efektif dalam mendeteksi objek, tetapi juga memberikan analisis yang dapat diandalkan dalam konteks penguasaan bola. Pada input video yang dimasukkan kedalam model dihasilkan bahwa tim *yellow* menguasai bola sebesar 64.13% sedangkan tim *white* sebesar 35.87%. Selain itu dilakukan juga perhitungan berapa lama tiap individu mampu menguasai bola, yang hasilnya dapat dilihat pada tabel 4.5 berikut:

Tabel 4.5
Hasil Perhitungan ball possession per individu

<i>Player ID</i>	<i>Time</i>
10	1.96 second
1	1.17 second

6	5.58 second
18	0.29 second
5	2.08 second
102	1.42 second
103	0.42 second
62	0.79 second
159	2.54 second
141	0.96 second

Dengan hasil tabel 4.5 diatas dapat disimpulkan bahwa pada vidio yang dilakukan penerapan model pemain dengan id 6 menguasai bola dengan paling tinggi. Dengan begitu model ini selain dapat melakukan perhitungan ball possession antar tim juga dapat menghitung per individu pemain.



Gambar 4.19 Objek keluar dari lapangan pertandingan

Pada gambar 4.19 dilakukan pengujian terhadap object yang keluar dari lapangan sepak bola dan hasilnya model tidak mendeteksi objek itu lagi, dengan begitu model tidak perlu membuat garis batas pada pinggir lapangan. Hal ini terjadi akibat dataset yang digunakan hanya ada objek yang berada pada dalam lapangan saja sehingga ketika berada diluar lapangan model itu tidak akan mendeteksi objek tersebut.