Sistem Deteksi Gerakan Dasar Bela Diri Taekwondo Menggunakan Arsitektur Yowo Dengan Rgb

1st Muchlis Aryomukti
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
muchlisaryo@student.telkomuniversity.ac.id

2nd Muhammad Nasrun
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
nasrun@telkomuniversity.ac.id

3rd Meta Kallista
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
metakallista@telkomuniversity.ac.id

***Abstrak*— Taekwondo merupakan salah satu olahraga cabang seni bela diri yang populer di Indonesia. Bela diri Taekwondo ini terdapat berbagai teknik gerakan dan jurus yang bisa dipelajari dengan cara dilatih oleh sabeum di dojang terkait yang selanjutnya diteruskan latihan mandiri. Akan tetapi, terdapat kendala untuk orang awam yang mempelajari gerakan ini karena tidak tahu nama teknik gerakan dan jurus dalam bela diri Taekwondo ketika mereka melihat orang yang berlatih atau mengikuti lomba Taekwondo yang menyebabkan mengalami kesulitan berlatih dengan sabeum. Arsitektur YOWO merupakan salah satu metode dalam deep learning yang digunakan untuk lokalisasi jenis gerakan manusia. YOWO menggunakan penggabungan 3D-CNN dengan 2D-CNN. RGB merupakan ekstraksi fitur yang bertujuan untuk membagi warna menjadi tiga (3) channel, yaitu Red, Green, dan Blue. Arsitektur YOWO cocok digunakan untuk mendeteksi gerakan berupa input video dan frame. Hasil yang didapat setelah melakukan pengujian average precision gerakan bela diri Taekwondo yaitu momtong jireugi sebesar 97.92% dengan nilai akurasi 99.70%, precision 99.18%, recall 93.90%, dan f1-score terbaik adalah 96.31%, dengan parameter batch size: 16, learning rate: 0.0001, num frames: 8, 3D-CNN dimension: 2, 2D-CNN dimension: 1, epochs: 10, num workers: 5, dan rasio dataset 60%:40%.**

Kata kunci*— *bela diri taekwondo, sistem deteksi gerakan dasar bela diri, arsitektur YOWO, RGB

I. PENDAHULUAN

Taekwondo merupakan salah satu olahraga cabang seni bela diri yang berasal dari Korea yang digemari oleh kalangan masyarakat di Indonesia. Ketika orang awam ingin mempelajari gerakan Taekwondo, tidak semua orang itu mengetahui nama gerakan dalam Taekwondo apalagi sikap dalam gerakan Taekwondo. Bahkan bila sudah *searching* informasi gerakan Taekwondo di internet tidak terlalu detil sehingga orang awam tersebut tidak terbayang bagaimana melakukan gerakan Taekwondo dengan benar. Untuk menangani permasalahan tersebut, diperlukan sebuah sistem yang dapat dapat bekerja secara *real-time* dengan cara menyorotkan kamera secara langsung kepada pengguna yang melakukan gerakan Taekwondo atau meng*upload* video ke dalam sistem agar mengetahui nama gerakan Taekwondo tersebut.

Penelitian ini membuat sistem deteksi gerakan dasar bela diri Taekwondo menggunakan arsitektur YOWO dengan RGB untuk melakukan deteksi gerakan yang diharapkan dapat membaca gerakan-gerakan Taekwondo beserta akurasi geraka tersebut agar dapat diterapkan langsung kepada masyarakat.

II. KAJIAN TEORI

A. Taekwondo

Taekwondo merupakan olahraga cabang seni bela diri yang berasal dari Korea. Taekwondo dalam bahasa Korea, *Hanja* untuk *Tae* artinya "menendang dengan kaki", *Kwon* artinya "meninju dengan tangan", dan *Do* artinya "jalan" atau "seni" atau "aliran". Dalam aturan PBTI [1] terdapat sepuluh (10) tingkatan peringkat dalam seni bela diri Taekwondo. Adapun untuk orang awam atau belum pernah berlatih Taekwondo di *dojang* dimulai dari tingkat dasar yaitu *geup* 10 dengan sabuk berwarna putih. Gerakan dalam *geup* 10 yaitu *momtong jireugi* (pukulan mengarah ke tengah atau ke ulu hati), *eolgol jireugi* (pukulan mengarah ke atas/ke kepala), *ap koobi seogi* (kuda-kuda sikap jalan panjang), *arae makki* (tangkisan ke arah bawah untuk menangkis tendangan lawan), *eolgol makki* (tangkisan ke arah kepala), *ap chagi* (tendangan depan ke arah perut memakai kaki depan), dan *dollyo chagi* (tendangan dari arah samping). Adapun jurus gerakan dalam *geup* 10 yaitu *Basic 1* dan *Basic 2*.

B. Pengenalan Aktivitas Manusia

Pengenalan aktivitas manusia (*human action recognition*) adalah bagian dari pengenalan gestur (*gesture recognition*) dan visual komputer (*computer vision*) pada bidang kecerdasan buatan (*artificial intelligence*). Pengenalan aktivitas manusia bisa digolongkan ke dalam pembelajaran mesin (*machine learning*) dan pembelajaran mendalam (*deep learning*). Hal tersebut karena dalam pembelajaran mesin terdapat cabang *supervised learning* yang di dalamnya terdapat jenis klasifikasi, dan secara rumus matematika sulit dipahami oleh manusia akan tetapi mudah dipahami oleh komputer. Sedangkan *deep learning* menggunakan konsep jaringan saraf (*neural network*) yang merupakan kombinasi dari tahap ekstraksi fitur dan tahap klasifikasi ketika tahap tersebut terdapat dalam pembelajaran mesin.

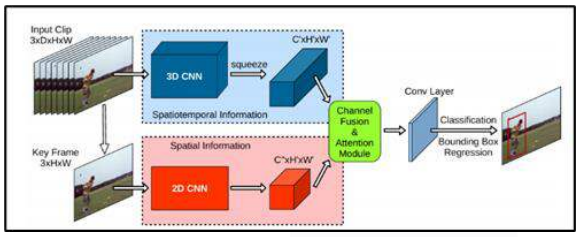
Hal yang membuat pengenalan aktivitas manusia terkesan istimewa dibandingkan cabang kecerdasan buatan yang lain adalah terdapat data koordinat rangka dan sendi [2].

C. Arsitektur YOWO (*You Only Watch Once*)

Arsitektur YOWO (*You Only Watch Once*) merupakan sebuah arsitektur *deep learning* yang bertujuan untuk melokalisasi jenis gerakan atau tindakan secara *spasiotemporal* (spasial dan temporal) di dalam video.

Ide konsep YOWO bermula dari kemampuan visual pada manusia. Cara memahami konsep YOWO, anggaplah kita sedang antusias menonton film "Guru Bangsa Tjokroaminoto" di dalam studio bioskop. Setiap detik yang kita tonton, mata kita menangkap satu *frame*. Kita melihat pemeran di film tersebut berbentuk 2D (dua dimensi) karena bentuk layar di bioskop, sedangkan kita memahami dalam film tersebut sebenarnya berbentuk 3D (tiga dimensi) sesuai realitas di kehidupan nyata, lalu hal tersebut disimpan di dalam memori otak kita. Setelah itu, kedua jenis dimensi tersebut digabungkan agar kita dapat menyimpulkan adegan di film itu dengan masuk akal [3][4].

Pada gambar 1, arsitektur YOWO terdiri dari empat bagian utama, yaitu: 3D-CNN, 2D-CNN, CFAM (*Channel Fusion Attention Module*), dan *bounding box regression*. Pada dasarnya, konsep YOWO mirip dengan YOLO (*You Only Look Once*), tetapi perbedaannya adalah di dalam YOLO versi manapun tidak menggunakan 3D-CNN.

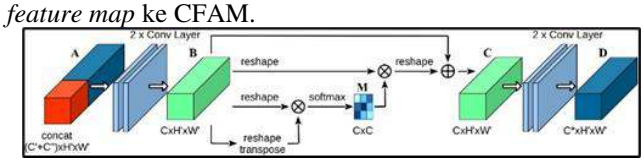


GAMBAR 1
Alur arsitektur YOWO

2D-CNN pada penelitian ini digunakan untuk memproses informasi spasial terkhusus *dataset* gambar berbentuk dua (2) dimensi. Arsitektur dasar yang digunakan adalah Darknet-19 karena arsitektur dasar tersebut memiliki keseimbangan yang baik antara akurasi dan efisiensi [5]. *Input* yang digunakan dalam 2D *network* adalah *frame* yang memiliki ukuran *input* $[C \times H \times W]$, ketika $C = 3$ yang merupakan perpaduan dari *channel* RGB, H dan W merupakan ukuran *height* dan *width* dari *frame*. Sedangkan *output* dari 2D *network* adalah *frame* yang memiliki ukuran *output* $[C' \times H' \times W']$, ketika C' adalah jumlah *output channel*, $H' = \frac{H}{32}$, dan $W' = \frac{W}{32}$.

3D-CNN pada penelitian ini digunakan untuk memproses informasi *spatiotemporal* (ruang dan waktu) terkhusus *dataset* video yang memiliki dimensi waktu selain memiliki dimensi ruang (*frame*). Arsitektur dasar yang digunakan adalah 3D-ResNext-101 karena arsitektur dasar tersebut memiliki perfomansi yang tinggi berdasarkan percobaan pada *dataset* Kinetics yaitu sekitar 94,5% untuk *dataset* UCF-101 dan 70,2% *dataset* HMDB-51 [6]. *Input* yang digunakan dalah 3D *network* adalah video yang memiliki ukuran *input* $[C \times D \times H \times W]$, ketika $C = 3$ yang merupakan perpaduan dari *channel* RGB, D adalah jumlah dimensi input *frame*, H dan W merupakan ukuran *height* dan *width* dari *frame*. Sedangkan *output* dari 3D *network* adalah *frame* yang memiliki ukuran *output* $[C'' \times D' \times H' \times W']$, ketika C' adalah jumlah *output channel*, $D' = 1$, $H' = \frac{H}{32}$, dan $W' = \frac{W}{32}$. Tujuan dimensi $D' = 1$ agar dimensi dari jumlah *input frame* (karena video terdiri dari kumpulan *frame*) bisa menyesuaikan ke 2D-CNN yang selanjutnya akan digabung dengan 2D-CNN.

Feature Aggregation: CFAM (*Channel Fusion and Attention Module*), setelah 2D-CNN dan 3D-CNN memiliki bentuk dimensi yang sama yaitu $D' = 1$, CFAM menggabungkan kedua hal tersebut. Hasil dari penggabungan tersebut mengkodekan informasi gerakan dan tampillan dari *feature map* ke CFAM.



GAMBAR 2
Ilustrasi penggabungan 2D-CNN dan 3D-CNN menggunakan modul CFAM (Channel Fusion and Attention Mechanism)

Gambar 2 mengilustrasikan kegunaan modul CFAM. *Concat* atau gabungan $A \in R^{(C'+C'')\times H\times W}$ merupakan informasi dari 2D-CNN dan 3D-CNN yang mengabaikan keterkaitan di antara keduanya. Lalu peneliti memasukkan A ke dalam dua (2) *convolutional layers* untuk menghasilkan *feature map* baru yaitu $B \in R^{C\times H'\times W'}$. Setelah itu, beberapa operasi dilakukan pada *feature map* B.

Misalkan $F \in R^{C\times N}$ adalah tensor yang dibentuk ulang dari *feature map* B, ketika $N = H \times W$, yang berarti bahwa fitur di setiap *channel* tunggal divektorisasi ke satu dimensi:

$$B \in R^{C\times H\times W} \xrightarrow{vektorisasi} F \in R^{C\times N} \quad (1)$$

Kemudian produk matriks antara $F \in R^{C\times N}$ dan transposnya $F^T \in R^{N\times C}$ dilakukan untuk menghasilkan matriks Gram $G \in R^{C\times C}$, yang menunjukkan korelasi fitur di seluruh *channel*:

$$G = F \cdot F^T \; dengan \; G_{ij} = \sum_{k=1}^{N} F_{ik} \cdot F_{jk} \quad (2)$$

Ketika setiap elemen G_{ij} dalam matriks Gram G mewakili produk dalam antara peta fitur vektor i dan j. Setelah menghitung matriks Gram, *softmax layer* diterapkan untuk menghasilkan *channel attention map* $M \in R^{C\times C}$:

$$M_{ij} = \frac{exp(G_{ij})}{\sum_{j=1}^{C} exp(G_{ij})} \quad (3)$$

ketika M_{ij} adalah skor yang mengukur dampak *channel* j^{th} pada *channel* i^{th}. Oleh karena itu M menjumlahkan dependensi fitur antar *channel* yang diberikan *feature map*. Untuk melakukan dampak *attention map* ke fitur asli, perkalian matriks lebih lanjut antara M dan F dilakukan dan hasilnya dibentuk kembali menjadi ruang tiga (3) dimensi $R^{C\times H\times W}$, yang memiliki bentuk yang sama dengan tensor input:

$$F' = M \cdot F \tag{4}$$

$$F' \in R^{C\times N} \xrightarrow{reshape} F'' \in R^{C\times H\times W} \tag{5}$$

Keluaran *channel attention modul* $C \in R^{C\times H\times W}$ menggabungkan hasil ini dengan *input feature map* asli B dengan parameter skalar yang dapat dilatih α menggunakan operasi penjumlahan berdasarkan elemen, dan α secara bertahap mempelajari bobot dari 0:

$$C = \alpha \cdot F'' + B \tag{6}$$

Persamaan (6) menunjukkan bahwa fitur akhir dari setiap *channel* adalah penjumlahan *weights* dari fitur semua *channel* dan fitur asli, yang memodelkan dependensi semantik jarak jauh antara *feature map*. Pada akhirnya, peta fitur $C \in R^{C\times H\prime\times W\prime}$ dimasukkan ke dalam dua *convolutional layer* lagi untuk menghasilkan *output feature map* $D \in R^{C*\times H\prime\times W\prime}$ dari modul CFAM. Dua *convolutional layer* di awal dan akhir modul CFAM sangat penting karena mereka membantu menggabungkan fitur yang berasal dari *backbone* yang berbeda dan memiliki kemungkinan distribusi yang berbeda. Tanpa *convolutional layer* ini, CFAM sedikit meningkatkan performa.

Bounding Box Regression pada kali ini mengikuti dari dokumentasi YOLO [5]. *Convolutional layer* akhir dengan *kernel* 1×1 diterapkan untuk menghasilkan jumlah *output channel* yang diinginkan. Untuk setiap *grid cell* dalam $H' \times W'$, lima (5) *anchor* sebelumnya dipilih dengan teknik k-means pada *dataset* yang sesuai dengan *NumCls class conditional action scores*, empat (4) koordinat dan *confidence score* membuat ukuran *output* akhir YOWO $[(5 \times (NumCls + 5) \times H' \times W']$. Regresi *bounding box* kemudian disempurnakan berdasarkan *anchor* ini.

Loss function didefinisikan serupa dengan *network* YOLOv2 asli [5] kecuali bahwa peneliti menerapkan *smooth* L_1 *loss* dengan beta = 1 untuk lokalisasi seperti pada [7], yang diberikan sebagai berikut:

$$L_{1,smooth}(x,y) = \begin{cases} 0{,}5(x-y)^2\,, & jika\ |x-y| < 1 \\ |x-y| - 0{,}5\,, & sebaliknya \end{cases} \tag{7}$$

ketika x dan y masing-masing mengacu pada prediksi *network* dan *groundtruth*. *Smooth* L_1 *loss* kurang sensitif terhadap *outlier* daripada *MSE loss* dan mencegah ledakan gradien dalam beberapa kasus. Peneliti masih menerapkan *MSE loss* untuk *confidence score*, yang diberikan sebagai berikut:

$$L_{MSE}(x,y) = (x-y)^2 \tag{8}$$

Akhir *detection loss* menjadi penjumlahan dari koordinat individual *loss* untuk x, y, *width*, dan *height*; dan *confidence score loss*, yang diberikan sebagai berikut:

$$L_D = L_x + L_y + L_w + L_h + L_{conf} \tag{9}$$

Peneliti telah menerapkan *focal loss* [8] untuk klasifikasi, yang diberikan sebagai berikut:

$$L_{focal}(x,y) = y(1-x)^\gamma \log(x) + (1-y)x^\gamma \log(1-x) \tag{10}$$

Ketika x adalah prediksi *network* dengan *softmax* dan $y \in \{0,1\}$ adalah *groundtruth class label*. γ adalah faktor modulasi, yang mengurangi sampel *loss* dengan *confidence* tinggi dan meningkatkan sampel *loss* dengan *confidence* rendah.

Loss akhir yang digunakan untuk optimalisasi arsitektur YOWO adalah penjumlahan *loss* deteksi dan klasifikasi, yang diberikan sebagai berikut:

$$L_{final} = \lambda L_D + L_{Cls} \tag{11}$$

ketika λ = 0,5 merupakan performa dalam penelitian ini

D. *Evaluation Metrics*

Evaluation metrics merupakan teknik evaluasi pada model dengan beberapa parameter untuk mengukur performa pada sebuah sistem. Parameter tersebut yaitu *accuracy*, *recall*, *precision*, F1-*score*, AP (*average precision*), dan mAP (*mean Average Precision*) [9].

1. Accuracy

Accuracy merupakan perbandingan antara nilai jawaban benar dengan total data [9]. Berikut persamaan *accuracy*:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

TP : *True Positive*
TN : *True Negative*
FP : *False Positive*
FN : *False Negative*

2. Recall

Recall merupakan perbandingan antara TP (*True Positive*) dengan jumlah TP dengan FN (*False Negative*). *Recall* digunakan jika terdapat *groundtruth* data yang bernilai benar akan tetapi terdeteksi salah atau jika nilai FN nya besar [9]. Berikut persamaan *recall*:

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

TP : *True Positive*
FN : *False Negative*

Jika semakin besar nilai FN nya, maka semakin kecil nilai *recall*nya.

3. *Precision*

Precision merupakan perbandingan antara TP dengan jumlah TP dengan FP (*False Positive*). *Precision* digunakan jika terdapat *groundtruth* data yang bernilai salah akan tetapi terdeteksi benar atau jika nilai FP nya besar [9]. Berikut persamaan *precision*:

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

TP : *True Positive*
FP : *False Positive*

Jika semakin besar nilai FP nya, maka semakin kecil nilai *precision*nya.

4. F1-*score*

F1-*score* merupakan gabungan antara *recall* dan *precision* [9]. Berikut persamaan f1-*score*:

$$F1 - score = \frac{2 * recall * precision}{recall + precision} \quad (15)$$

atau

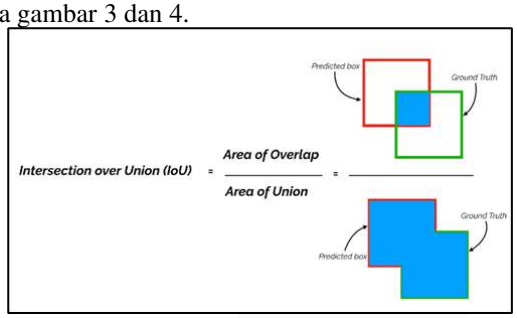
$$F1 - score = \frac{2TP}{2TP + FP + FN} \quad (16)$$

TP : *True Positive*
FP : *False Positive*
FN : *False Negative*

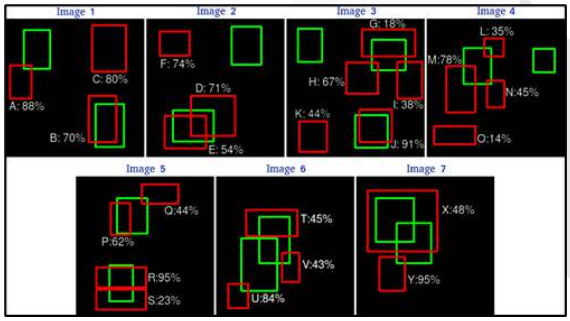
Untuk mendapatkan f1-*score* yang tinggi, diperlukan nilai *recall* dan *precision* yang tinggi.

5. AP (*Average Precision*)

AP (*Average Precision*) merupakan perbandingan antara nilai *recall* dan *precision* yang sudah diplot dalam grafik. Biasanya ditentukan oleh nilai IoU (*Intersection over Union*) yang merupakan perbandingan antara irisan area *groundtruth box* dan area *detection box* dengan gabungan area *groundtruth box* dan *detection box* [10][11]. Berikut ilustrasi IoU pada gambar 3 dan 4.



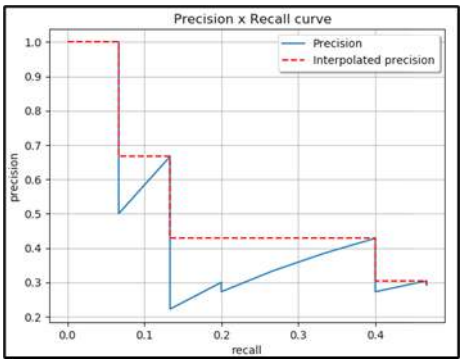
GAMBAR 3
Konsep sederhana IoU (Intersection over Union)



GAMBAR 4
Ilustrasi penerapan IoU

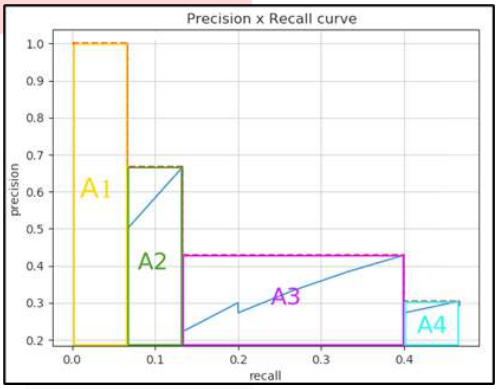
Pada gambar 4, *groundtruth box* ditandai dengan warna hijau, sedangkan *detection box* ditandai dengan warna merah.

Dalam menentukan perbandingan performa *recall* dengan *precision* diperlukan grafik *interpolated precision* untuk menghitung AUC (*Area Under the Curve*) nya, biasanya menggunakan metode PASCAL VOC. Berikut contoh cara menentukan nilai AP.



GAMBAR 5
Grafik interpolated precision menggunakan metode PASCAL VOC interpolating all points

Pada gambar 5 misal diketahui dengan IoU yang sudah ditentukan sebesar 30%, jika IoU < 30% maka bernilai FP, sedangkan IoU > 30% maka bernilai TP, dan nilai *recall* dan *precision* sudah ditentukan pada grafik. Lalu untuk AUC nya seperti pada gambar 6



GAMBAR 6
AUC (Area Under the Curve) pada grafik interpolated precision

Setelah mendapat AUC, selanjutnya dikalkulasikan dengan persamaan AP sebagai berikut:

$$AP = A1 + A2 + A3 + A4 \quad (17)$$

Dengan:
A1 : (0.0666 – 0) x 1 = 0.0666
A2 : (0.1333 – 0.0666) x 0.6666 = 0.04446222
A3: (0.4 – 0.1333) x 0.4285 = 0.11428095
A4: (0.4666 – 0.4) x 0.3043 = 0.02026638

$$AP = 0.0666 + 0.04446222 + 0.11428095 + 0.02026638$$

$$AP = 0.24560955$$
$$AP = 24.56\%$$

Jadi, nilai AP pada gambar 6 adalah 24.56%.

6. mAP (*mean Average Precision*)

Di dalam metode PASCAL VOC, mAP (*mean Average Precision*) merupakan rata-rata dari perbandingan jumlah nilai AP pada jumlah kategori objek dengan total kategori objek menggunakan IoU sebagai *treshold* [12]. Berikut persamaan mAP secara sederhana menggunakan metode PASCAL VOC:

$$mAP@IoU = \frac{\sum_{i=1}^{n} AP_i}{n}$$

$$mAP@IoU = \frac{AP_1 + AP_2 + AP_3 + \cdots + AP_n}{n} \quad (18)$$

AP : *Average Precision*
n : jumlah kategori objek
IoU : *Intersection over Union*

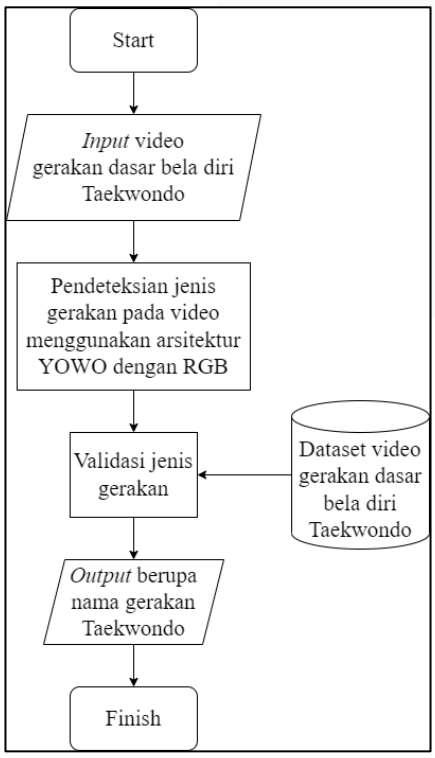
III. METODE

A. Rancangan Umum Sistem

Pada penelitian ini dirancang struktur dalam sistem secara umum. Sistem deteksi teknik gerakan dasar bela diri Taekwondo digunakan untuk mendeteksi jenis teknik gerakan dasar dalam bela diri Taekwondo berdasarkan tingkat dasar sabuk putih Taekwondo (*Geup 10*). Arsitektur yang digunakan dalam sistem deteksi teknik gerakan dasar bela diri Taekwondo yaitu arsitektur YOWO (*You Only Watch Once*). Arsitektur ini memproses masukan video dari pengguna berupa *upload file* video rekaman teknik gerakan dalam bela diri Taekwondo atau mendeteksi teknik gerakan dasar dalam bela diri Taekwondo menggunakan kamera secara langsung ke dalam sistem deteksi teknik gerakan dasar bela diri Taekwondo. Hasil dari penggunaan arsitektur YOWO pada sistem deteksi ini berupa penentuan nama teknik gerakan dasar dalam bela diri Taekwondo dan nilai akurasi seperti *accuracy*, f1-*score*, dan *average precision*.

B. Diagram Alir/*Flowchart* Sistem

Diagram alir/*flowchart* sistem merupakan diagram alir yang digunakan untuk memperlihatkan alur kerja pemrograman sistem yang bertujuan mempermudah alur pemodelan sistem. Ilustrasi *flowchart* dijelaskan dalam gambar 7 berikut.



GAMBAR 2
Sistem diagram alir deteksi gerakan dasar bela diri Taekwondo.

Diagram alir pada gambar 7 memperlihatkan alur kerja sistem dari Sistem Deteksi Gerakan Dasar Bela Diri Taekwondo Menggunakan Arsitektur YOWO dengan RGB. Sistem ini awalnya mengambil data yang dimasukkan pengguna berupa video teknik gerakan dasar bela diri Taekwondo.

Kemudian setelah mendapatkan data yang dimasukan oleh pengguna ke dalam sistem, sistem akan mendeteksi jenis teknik gerakan tersebut menggunakan arsitektur YOWO dengan RGB. Lalu akan divalidasi dengan membandingkan *input* tersebut dengan *dataset* video gerakan bela diri Taekwondo. Hasil dari validasi berupa persentase akurasi data dari *input* video dan penamaan teknik gerakan dasar dalam bela diri Taekwondo.

C. *Preprocessing Data*

Preprocessing data merupakan persiapan sebelum memproses data agar model tidak mengalami kesalahan dalam proses *training*. Berikut tahapan *preprocessing data*:

1. Penyamaan format *groundtruths custom dataset* dengan AVA *dataset* berupa penambahan kategori *timestamp* yang berdasarkan detik dimulainya di YouTube serta kustomisasi dari format YOLO Darknet.
2. Penyamaan format *label* dan *groundtruths custom dataset* dengan AVA *dataset* dikarenakan belum tersedianya penggunaan arsitektur YOWO untuk *custom dataset*, yaitu menggunakan *file* berekstensi .pbtxt.
3. Me*rename frame* agar *frame* dapat berurut sesuai gerakan Taekwondo pada video serta dapat dibaca oleh *groundtruths*.

D. *Dataset*

Kustomisasi *dataset* merupakan *dataset* baru yang diambil secara langsung oleh peneliti berupa rekaman video sesuai ketentuan penelitian ini yaitu tingkat dasar sabuk putih (*geup* 10) dalam bela diri Taekwondo. *Dataset* tersebut terdapat seribu (1.000) *frame* dan empat puluh (40) video yang dijadikan bahan *training set* dan *test set*. Isi dari *dataset* tersebut berupa gerakan bernama *momtong jierugi*.

E. Anotasi Gambar/*Frame*

Anotasi gambar merupakan kumpulan format yang mengatur *dataset* gambar/*frame* berupa *labelmap*, *train.csv, testlist.csv,* dan *bounding box* yang memiliki koordinat sebagai *groundtruths*nya. Anotasi gambar menggunakan *website Roboflow* untuk menentukan gerakan Taekwondo dari *dataset* yang sudah dikumpulkan oleh peneliti ke dalam *website Roboflow*.

F. Anotasi Video

Anotasi video merupakan kumpulan format yang mengatur *dataset* video berupa *train list file* dan *test list file* dengan ekstensi .txt dan kumpulan *file* video dengan ekstensi .mp4.

IV. HASIL DAN PEMBAHASAN

Parameter dalam pengujian ini diinisialisasi sebagai berikut: *batch size*: 16, *learning rate*: 0.0001, *num frames*: 8, 3D-CNN *dimension*: 2, 2D-CNN *dimension*: 1, *epochs*: 10, *num workers*: 5, dan rasio *dataset* 60%:40%. Variabel penentu performa sistem yaitu nilai *accuracy*, *precision*, *recall*, dan f1-*score*.

A. Pengujian *Epochs*

1. Pengujian Epochs akan diuji sebanyak sepuluh (10) tahap. Epochs ke-1

Setelah dilakukan *training model* dengan *epochs* ke-1, didapatkan hasil sebagai berikut.

TABEL 1

Parameter	Nilai
Accuracy	99.40%
Precision	94.60%
Recall	92.70%
F1-*score*	93.35%

2. Epochs ke-2

Setelah dilakukan *training model* dengan *epochs* ke-2, didapatkan hasil sebagai berikut.

TABEL 2

Parameter	Nilai
Accuracy	99.60%
Precision	98.14%
Recall	93.40%
F1-*score*	95.51%

3. Epochs ke-3

Setelah dilakukan *training model* dengan *epochs* ke-3, didapatkan hasil sebagai berikut.

TABEL 3

Parameter	Nilai
Accuracy	99.70%
Precision	98.68%
Recall	94.10%
F1-*score*	96.18%

4. Epochs ke-4

Setelah dilakukan *training model* dengan *epochs* ke-4, didapatkan hasil sebagai berikut.

TABEL 4

Parameter	Nilai
Accuracy	99.70%
Precision	99.18%
Recall	93.90%
F1-*score*	96.31%

5. Epochs ke-5

Setelah dilakukan *training model* dengan *epochs* ke-5, didapatkan hasil sebagai berikut.

Tabel 5

Parameter	Nilai
Accuracy	99.40%
Precision	98.92%
Recall	93.40%
F1-*score*	95.74%

6. Epochs ke-6

Setelah dilakukan *training model* dengan *epochs* ke-6, didapatkan hasil sebagai berikut.

Tabel 6

Parameter	Nilai
Accuracy	99.60%
Precision	99.34%
Recall	92.60%
F1-*score*	95.66%

7. Epochs ke-7

Setelah dilakukan *training model* dengan *epochs* ke-7, didapatkan hasil sebagai berikut.

TABEL 7

Parameter	Nilai
Accuracy	99.60%
Precision	99.63%
Recall	91.70%
F1-*score*	95.28%

8. Epochs ke-8

Setelah dilakukan *training model* dengan *epochs* ke-8, didapatkan hasil sebagai berikut.

TABEL 8

Parameter	Nilai
Accuracy	99.50%
Precision	99.03%
Recall	92.90%
F1-*score*	95.62%

9. Epochs ke-9

Setelah dilakukan *training model* dengan *epochs* ke-9, didapatkan hasil sebagai berikut.

TABEL 9

Parameter	Nilai
Accuracy	99.20%
Precision	98.81%
Recall	93.10%
F1-*score*	95.49%

10. Epochs ke-10

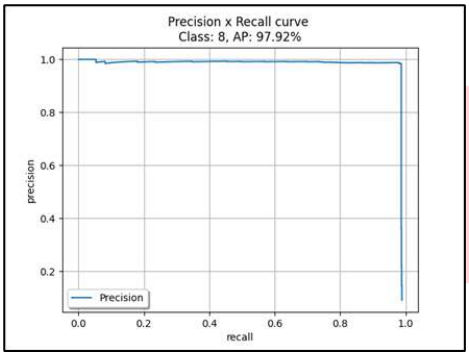
Setelah dilakukan *training model* dengan *epochs* ke-10, didapatkan hasil sebagai berikut.

TABEL 10

Parameter	Nilai
Accuracy	99.60%
Precision	99.32%
Recall	91.80%
F1-*score*	95.19%

B. Pengujian *Average Precision* pada Gerakan *Momtong Jireugi*

Hasil AP (*Average Precision*) pada gerakan *momtong jireugi* adalah 97.92% seperti pada gambar 8.



GAMBAR 8
AP pada gerakan momtong jireugi

Jika *precision* dan *recall* dibuat contoh perhitungan manual berdasarkan gambar 8 menggunakan metode PASCAL VOC, maka hasilnya akan ditampilkan pada tabel 11.

TABEL 11
Interpolated precision pada AP *momtong jireugi*

Nomor	*Precision*	*Recall*
1	1.00	0.01
2	0.99	0.03
3	0.99	0.06
4	0.99	0.07
5	0.99	0.08
6	0.94	0.86
7	0.94	0.88
8	0.40	0.98
9	0.17	0.99
10	0.01	1.00

Diketahui:

$$A_n = (recall - recall_{n-1}) \times precision$$

A1 : (0.01 – 0) x 1 = 0.01
A2 : (0.03 – 0.01) x 0.99 = 0.0198
A3: (0.05 – 0.03) x 0.99 = 0.0198
A4: (0.07 – 0.05) x 0.99 = 0.0198
A5: (0.08 – 0.07) x 0.99 = 0.0099
A6: (0.86 – 0.08) x 0.94 = 0.7332
A7: (0.88 – 0.86) x 0.94 = 0.0188
A8: (0.98 – 0.88) x 0.87 = 0.087
A9: (0.99 – 0.98) x 0.17 = 0.0017
A10: (1 – 0.99) x 0.01 = 0.0001

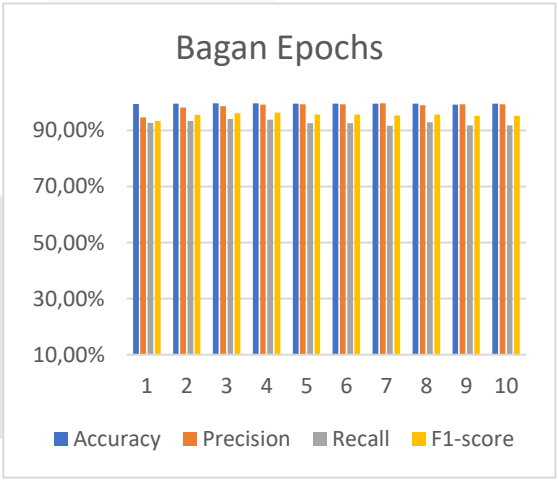
Penyelesaian:
AP = A1+A2+A3+A4+A5+A6+A7+A8+A9+A10
AP = 0.01 + 0.0396 + 0.0099 + 0.0099 + 0.0099 + 0.7332 + 0.0188 + 0.087 + 0.0017 + 0.0001
AP = 0.9184
AP *momtong jireugi* = 91.84%

C. Analisis Hasil *Training Epochs*

Epochs pada *training* merupakan proses ketika seluruh *train data* sudah selesai melakukan proses *training* hingga kembali lagi ke posisi awal dalam sekali periode.

TABEL 12
Hasil *Epochs*

Epochs	*Accuracy*	*Precision*	*Recall*	F1-*score*
1	99.40%	94.60%	92.70%	93.35%
2	99.60%	98.14%	93.40%	95.51%
3	99.70%	98.68%	94.10%	96.18%
4	99.70%	99.18%	93.90%	96.31%
5	99.60%	99.34%	92.60%	95.66%
6	99.60%	99.34%	92.60%	95.66%
7	99.60%	99.63%	91.70%	95.28%
8	99.50%	99.03%	92.90%	95.62%
9	99.20%	99.32%	91.80%	95.19%
10	99.60%	99.32%	91.80%	95.19%
Rata-rata	99.55%	98.66%	92.75%	95.40%



GAMBAR 3
Grafik *Epochs*

Rata-rata *accuracy* paling tinggi di antara keempat variabel yaitu 99.55% dikarenakan nilai pembagi FP dan FN sangat kecil sehingga memengaruhi perbandingan TP dan TN dengan TP, TN, FP, dan FN.

Lalu rata-rata *recall* paling rendah di antara keempat variabel yaitu 92.75% dikarenakan nilai pembagi FN lebih besar sehingga memengaruhi perbandingan TP dengan TP dan FN.

Nilai f1-*score* tertinggi pada epochs ke-4, yaitu sebesar 96.31% dikarenakan nilai *precision* dan *recall* yang tinggi yaitu 99.18% dan 93.90% memengaruhi komputasi f1-*score*.

D. Analisis Hasil *Average Precision* pada Gerakan *Momtong Jireugi*

Hasil dari nilai AP *momtong jireugi* pada gambar 8 adalah 97.92%. Jika dibanding dengan perhitungan AP *momtong jireugi* secara manual pada tabel 11 yaitu sebesar 91.84%, maka nilai AP *momtong jireugi* pada gambar 8 lebih tinggi dikarenakan *recall* dan *precision* lebih banyak sehingga tingkat ketelitian pada nilai AP yang lebih tinggi.

V. KESIMPULAN

Berdasarkan hasil dari penelitian serta pengujian dan analisis yang telah dilakukan, sistem deteksi gerakan dasar bela diri Taekwondo menggunakan arsitektur YOWO dengan RGB berhasil menghasilkan f1-*score* terbaik sebesar 96.31% dengan nilai *accuracy* adalah 99.70%, nilai *precision* adalah 99.18%, dan nilai *recall* adalah 93.90%, dengan parameter *batch size*: 16, *learning rate*: 0.0001, *num frames*: 8, 3D-CNN *dimension*: 2, 2D-CNN *dimension*: 1, *epochs*: 10, *num workers*: 5, dan rasio *dataset* 60%:40%. Lalu sistem tersebut berhasil menghasilkan nilai *Average Precision* pada gerakan *momtong jireugi* sebesar 97.92% menggunakan metode PASCAL VOC.

REFERENSI

[1] Penjasorkes. “16 Tahap Tingkatan Sabuk Taekwondo di Indonesia”. Internet: https://www.penjasorkes.com/2019/03/tahap-tingkatan-sabuk-taekwondo-di.html# [Dec. 24, 2021]

[2] Xu, Liang., Lan, Cuiling., Zeng, Wenjun., dan Lu, Cewu. (2021). “Skeleton-Based Mutually Assisted Interacted Object Localization and Human Action Recognition”. arXiv preprint arXiv:2110.14994.

[3] Köpüklü, Okan., Wei, Xiangyu., dan Rigoll, Gerhard. (2019). “You Only Watch Once: A Unified CNN Architecture for Real-Time Spatiotemporal Action Localization”. arXiv preprint arXiv:1911.06644.

[4] Mo, Shentong., Tan, Xiaoqing., Xia, Jingfei., dan Ren, Pinxu. (2020). “Towards Improving Spatiotemporal Action Recognition in Videos”. arXiv preprint arXiv:2012.08097.

[5] J. Redmon dan A. Farhadi. (2017). “Yolo9000: better, faster, stronger”, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263–7271.

[6] K. Hara, H. Kataoka, dan Y. Satoh. (2018). “Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?”, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 6546–6555

[7] R. Girshick. (2015). “Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision”, pp. 1440–1448.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Doll´ar. (2017). “Focal loss for dense object detection”, in: Proceedings of the IEEE international conference on computer vision, pp. 2980–2988.

[9] R. Yacouby dan D. Axman. 2020. “Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models”. In Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, pages 79–91, Online. Association for Computational Linguistics.

[10] Y. Shivy. (2020). “mAP (mean Average Precision) might confuse you!”. Internet: https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2 [Feb. 20, 2023

[11] P. Rafael. (2021). “Object-Detection-Metrics”. Internet: https://github.com/rafaelpadilla/Object-Detection-Metrics#metrics [Feb. 20, 2023]

[12] H. Jonathan. (2018). “mAP (mean Average Precision) for Object Detection”. Internet: https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173 [Feb. 20, 2023]