

# Implementasi Perhitungan Operasi Matematika Sederhana pada *Jetson Nano* untuk Aplikasi *Smart Whiteboard*

Achmad Pahlevy Aminullah Nizaruddin, Eko Mulyanto Yuniarno dan Dion Hayu Fandiartoro  
Departemen Teknik Komputer, Institut Teknologi Sepuluh Nopember  
*e-mail*: ekomulyanto@ee.its.ac.id

**Abstrak**—Dalam era modern revolusi industri 4.0 saat ini, hampir seluruh kegiatan manusia terkait dengan pemanfaatan teknologi informasi sebagai kebutuhan dalam berbagai kegiatan dan layanan. Salah satu contohnya dalam dunia pendidikan. Media pembelajaran interaktif sebagai salah satu media pembelajaran di dunia pendidikan yang menggunakan teknologi informasi dan komunikasi sehingga mempermudah seorang pendidik dalam menyampaikan materi atau informasi kepada siswa. Salah satu bentuk menggunakan teknologi sebagai media pembelajaran adalah *Smart Whiteboard*. Dalam pembuatan *Smart Whiteboard* membutuhkan suatu sistem yang dapat mendeteksi dan mengklasifikasi apa yang ditulis pada papan tulis salah satunya adalah menghitung suatu operasi matematika. Untuk dapat mendeteksi dan mengklasifikasi objek digunakan metode yaitu *deep learning*. YOLO (*You Only Look Once*) adalah salah satu jenis metode dalam *deep learning* yang dikembangkan dengan algoritma untuk mendeteksi objek dalam waktu nyata berdasarkan CNN (*Convolutional Neural Network*). Penelitian ini bertujuan untuk melakukan implementasi sistem yang dapat mendeteksi angka dan operator aritmatika dan menghitung operasi aritmatika pada tulisan pada papan tulis *Jetson Nano*. Adapun untuk sistem tersebut, jenis YOLO yang digunakan adalah *YOLOv4-tiny* dengan menggunakan *framework Darknet*. YOLO dipilih karena akurasi dan kecepatannya. Sistem deteksi bekerja dengan cara mendeteksi huruf, angka, dan operator aritmatika yang ditulis tangan, dan jika suatu operasi matematika terbentuk, hasilnya akan dihitung. Hasil dari penelitian ini adalah suatu sistem yang dapat mendeteksi operasi matematika menggunakan gambar, video yang direkam, atau video dalam waktu nyata.

**Kata Kunci**—*Jetson Nano*, Media Pembelajaran, *Smart Whiteboard*, YOLO.

## I. PENDAHULUAN

DALAM era modern revolusi industri 4.0 saat ini, hampir semua aktivitas kehidupan manusia tidak dapat dipisahkan dari penggunaan teknologi informasi sebagai kebutuhan berbagai aktivitas dan layanan. Perkembangan teknologi informasi yang sangat mempermudah aktivitas kehidupan manusia telah menciptakan ketergantungan yang sangat kuat pada keberadaan teknologi informasi. Teknologi informasi adalah teknologi yang digunakan untuk mengelola data dan informasi, termasuk pemrosesan, pengadaan, pengumpulan, penyimpanan, dan manipulasi data dalam berbagai cara untuk menghasilkan informasi berkualitas, yaitu informasi yang relevan, akurat, dan tepat waktu yang digunakan untuk kepentingan pribadi dan organisasi.

Peran teknologi dalam pendidikan sangat penting dan dibutuhkan. Dalam Undang-Undang Nomor 20 tahun 2003 tentang Sistem Pendidikan Nasional disebutkan bahwa pengembangan teknologi untuk pendidikan adalah usaha yang sadar dan terstruktur untuk menciptakan suasana dan

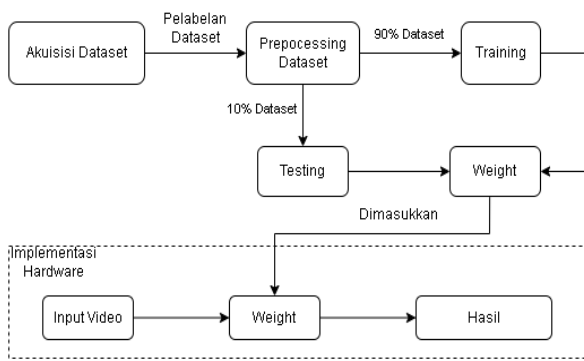
proses belajar sehingga siswa dapat secara aktif membangun potensinya. Dengan memanfaatkan perkembangan teknologi canggih sebagai media pembelajaran di dunia pendidikan, akan mampu mengubah cara seseorang belajar, memperoleh informasi, menyesuaikan informasi dan seterusnya. Media pembelajaran sebagai sarana peningkatan kualitas pendidikan sangat penting dalam proses pembelajaran dan mengajar. Media pembelajaran interaktif sebagai salah satu media pembelajaran yang menggunakan teknologi informasi dan komunikasi memudahkan guru dalam menyampaikan materi atau informasi kepada siswa. Media pembelajaran interaktif dapat memfasilitasi guru dalam mengembangkan teknik pembelajaran agar menghasilkan hasil yang maksimal [1]. Salah satu bentuk penggunaan teknologi sebagai media pembelajaran di dunia pendidikan adalah *Smart Whiteboard*.

Saat ini masih banyak sekolah di Indonesia yang masih dalam proses pembelajaran di kelas menggunakan media pembelajaran whiteboard biasa tanpa adanya fitur apapun. Whiteboard biasa memiliki ruang yang terbatas dan materi ajar yang ditulis tidak bisa direkam, sehingga materi yang disampaikan oleh guru tidak bisa diambil dengan baik oleh siswa. Dengan *Smart Whiteboard*, masalah-masalah tersebut bisa diselesaikan dan dapat meningkatkan kualitas pembelajaran siswa di kelas, sehingga membuat suasana belajar dan mengajar efektif, efisien, dan tidak membosankan. Selain itu, *Smart Whiteboard* juga akan mampu meningkatkan motivasi belajar siswa karena dapat memberikan inovasi baru bagi siswa [2].

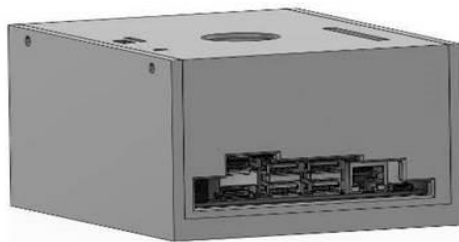
*Smart Whiteboard* sendiri adalah papan tulis menggunakan proyektor yang dapat mendeteksi dan mengklasifikasi apa pun yang ditulis di papan tulis, salah satunya adalah melakukan operasi aritmatika. Untuk dapat mendeteksi dan mengklasifikasi objek, digunakan sebuah metode, yaitu *deep learning*. YOLO adalah jenis metode *deep learning* yang dikembangkan dengan algoritma untuk mendeteksi objek secara real time berdasarkan CNN (*Convolutional Neural Network*). YOLO menerapkan jaringan saraf tunggal ke seluruh gambar. Jaringan ini akan membagi gambar menjadi daerah dan kemudian memprediksi kotak batas dan probabilitas, untuk setiap kotak daerah batas probabilitas diberatkan untuk diklasifikasikan sebagai objek atau tidak.

## II. METODE PENELITIAN

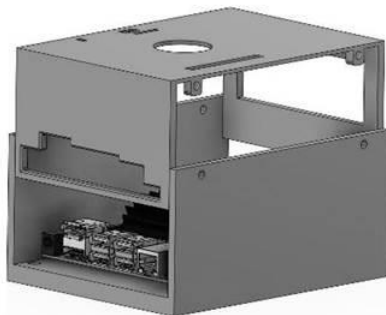
Studi ini adalah studi di bidang visi komputer yang bertujuan untuk mendeteksi angka dan operator aritmatika untuk menghitung operasi matematika pada media whiteboard menggunakan metode *You Only Look Once* (YOLO), khususnya *YOLOv4-tiny*, yang kemudian



Gambar 1. Diagram Blok Metodologi.



Gambar 2. Sisi Kiri Casing.



Gambar 3. Sisi Kanan Casing.

diimplementasikan pada *Jetson Nano*. Gambar 1 menunjukkan diagram blok metodologi.

**A. Hardware Pembantu**

Untuk menghubungkan *Jetson Nano* dengan Whiteboard, Penulis membutuhkan sebuah *casing* yang terhubung dengan sebuah *bracket*. Desain *casing* 3D dapat dilihat pada Gambar 2 dan Gambar 3.

Gambar 2 menunjukkan penampilan desain dari *casing* Alat yang berisi *Jetson Nano* dan Kamera yang dilihat dari sisi kiri. Di sisi belakang *casing* terdapat sebuah lubang yang dibuat sedemikian rupa sehingga menyesuaikan ukuran port *input Jetson Nano*.

Kemudian Gambar 3 menunjukkan penampilan desain dari *casing* Alat yang berisi *Jetson Nano* dan Kamera yang dilihat dari sisi kanan. Di sisi depan *casing* terdapat sebuah lubang yang memiliki fungsi sebagai tempat untuk *webcam* untuk mengambil *input video*, tentu saja lubang tersebut dibuat dengan ukuran yang cocok dengan *webcam* yang digunakan.

Pada Gambar 4 adalah ilustrasi dari *Hardware* yang akan digunakan sebagai implementasi sistem. Komposisi atau isi dari *Hardware* itu sendiri adalah *Jetson Nano* bersama dengan *Fan* dan *Webcam*.



Gambar 4. Tampak Dalam Casing.



Gambar 5. Casing ketika telah terhubung dengan papan tulis.

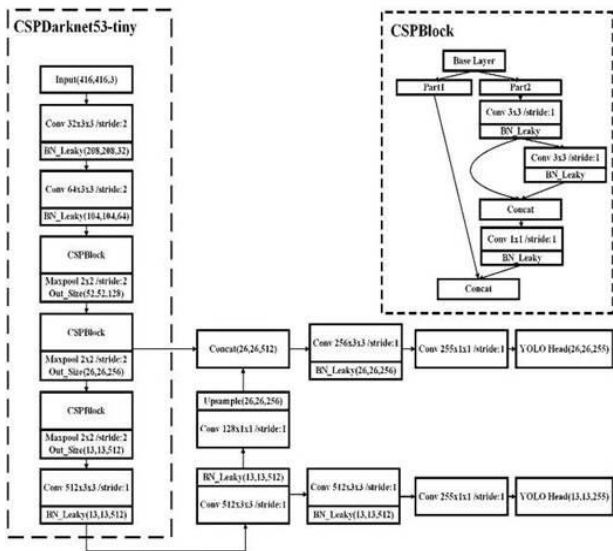
Untuk *casing* alat dapat terhubung dengan papan putih dan mudah diposisikan, sebuah *bracket* pengait dibuat antara *casing* alat dan papan hitam yang dapat diubah panjangnya (jarak antara *casing* dan papan hitam) bersama dengan sudut miring *casing* menuju papan hitam, catatlah bahwa jarak yang dapat disesuaikan adalah 60 - 90cm. Untuk dokumentasi *bracket* dapat dilihat pada Gambar 5.

**B. YOLO**

YOLO (*You Only Look Once*) adalah sebuah algoritma deteksi objek yang saat ini sangat populer untuk dikembangkan. YOLO menggunakan satu jaringan saraf tunggal untuk diterapkan pada seluruh gambar. Deteksi objek dilakukan dengan membingkai objek yang akan dideteksi sebagai masalah regresi dan memisahkan kotak batas khusus dan probabilitas kelas. Dengan menggunakan satu jaringan saraf tunggal untuk memprediksi kotak batas dan probabilitas kelas dari semua gambar dalam satu evaluasi. Karena metode ini menggunakan satu jaringan saraf tunggal untuk semua pipa deteksi, kinerja deteksi dapat dioptimalkan dari ujung ke ujung [3].

YOLO membagi gambar atau video yang dimasukkan menjadi *grid S X S*. Jika titik pusat koordinat pada GT (*Ground Truth*) dari suatu objek jatuh ke dalam *grid*, maka *grid* tersebut bertanggung jawab untuk mendeteksi objek. Inovasi YOLO adalah memodifikasi kerangka deteksi *Proposed Region*: seri R-CNN membutuhkan untuk memproduksi wilayah proposal dimana untuk melakukan proses klasifikasi dan regresi. Namun ada *overlap* (tumpang tindih) dimana wilayah proposal akan membawa banyak proses iteratif. Namun, YOLO memprediksi kotak batas objek yang ada di semua *grid*, lokasi dan probabilitas semua kelas sekaligus, sehingga YOLO memecahkan semua masalah dalam satu *run*.

Metode *YOLOv4-tiny* dirancang berdasarkan metode *Yolov4* untuk memiliki kecepatan deteksi objek yang lebih



Gambar 6. Struktur Jaringan YOLOv4-tiny.

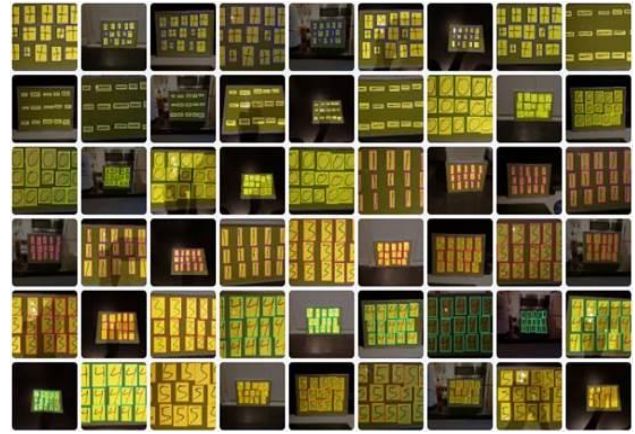
Layer	Type	Filters	Size/Stride	Input	Output
0	Convolutional	32	3 × 3/2	416 × 416 × 3	208 × 208 × 32
1	Convolutional	64	3 × 3/2	208 × 208 × 32	104 × 104 × 64
2	Convolutional	64	3 × 3/1	104 × 104 × 64	104 × 104 × 64
3	Route 2				
4	Convolutional	32	3 × 3/1	104 × 104 × 32	104 × 104 × 32
5	Convolutional	32	3 × 3/1	104 × 104 × 32	104 × 104 × 32
6	Route 5 4				
7	Convolutional	64	1 × 1/1	104 × 104 × 64	104 × 104 × 64
8	Route 2 7				
9	Maxpool		2 × 2/2	104 × 104 × 128	52 × 52 × 128
10	Convolutional	128	3 × 3/1	52 × 52 × 128	52 × 52 × 128
11	Route 10				
12	Convolutional	64	3 × 3/1	52 × 52 × 64	52 × 52 × 64
13	Convolutional	64	3 × 3/1	52 × 52 × 64	52 × 52 × 64
14	Route 13 12				
15	Convolutional	128	1 × 1/1	52 × 52 × 128	52 × 52 × 128
16	Route 10 15				
17	Maxpool		2 × 2/2	52 × 52 × 256	26 × 26 × 256
18	Convolutional	256	3 × 3/1	26 × 26 × 256	26 × 26 × 256
19	Route 18				
20	Convolutional	128	3 × 3/1	26 × 26 × 128	26 × 26 × 128
21	Convolutional	128	3 × 3/1	26 × 26 × 128	26 × 26 × 128
22	Route 21 20				
23	Convolutional	256	1 × 1/1	26 × 26 × 256	26 × 26 × 256
24	Route 18 23				
25	Maxpool		2 × 2/2	26 × 26 × 512	13 × 13 × 512
26	Convolutional	512	3 × 3/1	13 × 13 × 512	13 × 13 × 512
27	Convolutional	256	1 × 1/1	13 × 13 × 512	13 × 13 × 256
28	Convolutional	512	3 × 3/1	13 × 13 × 256	13 × 13 × 512
29	Convolutional	21	1 × 1/1	13 × 13 × 512	13 × 13 × 21
30	YOLO				
31	Route 27				
32	Convolutional	128	1 × 1/1	13 × 13 × 256	13 × 13 × 128
33	Upsample		2x	13 × 13 × 128	26 × 26 × 128
34	Route 33 23				
35	Convolutional	256	3 × 3/1	26 × 26 × 384	26 × 26 × 256
36	Convolutional	21	1 × 1/1	26 × 26 × 256	26 × 26 × 21
37	YOLO				

Gambar 7. Struktur Layer dari YOLOv4-tiny.

cepat. Kecepatan deteksi objek untuk YOLOv4-tiny dapat mencapai 371 frame per detik menggunakan 1080Ti GPU, membuatnya sangat cocok untuk digunakan pada sistem terintegrasi atau perangkat mobile.

Metode YOLOv4-tiny menggunakan jaringan CSPDarknet53-tiny sebagai jaringan dasar bukan jaringan CSPDarknet53 yang digunakan dalam metode YOLOv4. Jaringan CSPDarknet53-tiny menggunakan modul CSPBlock dalam jaringan partial tahap silang bukan modul ResBlock dalam jaringan residual.

Modul CSPBlock membagi feature map menjadi dua bagian dan menggabungkan kedua bagian dengan tepi residual melintang. Modul CSPBlock dapat meningkatkan kemampuan belajar jaringan konvolusi dibandingkan dengan modul ResBlock. Sementara ini meningkatkan komputasi sebesar 10% hingga 20%, juga meningkatkan akurasi. Untuk mengurangi jumlah perhitungan, hal ini menghilangkan kebuntuan komputasi yang memiliki jumlah perhitungan lebih tinggi di modul CSPBlock.



Gambar 8. Dataset Math Object Detection Dataset.



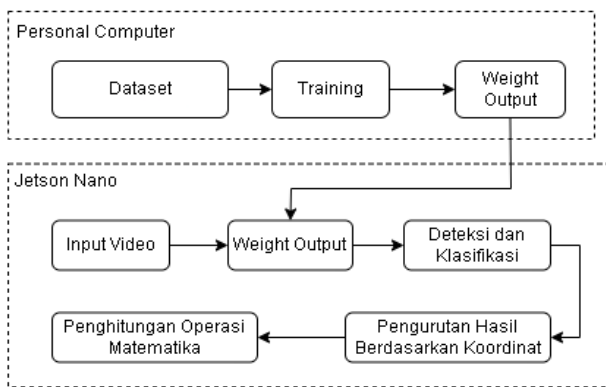
Gambar 9. Dataset Deteksi Angka dan Operator Aritmatika Menggunakan YOLO untuk Menghitung Operasi Matematika Sederhana pada Smart Whiteboard.

Hal ini meningkatkan akurasi dari metode YOLOv4-tiny dalam hal komputasi yang konstan atau bahkan dikurangi. Untuk mempermudah proses komputasi, metode YOLOv4-tiny menggunakan fungsi LeakyReLU sebagai fungsi aktivasi pada jaringan CSPDarknet53-tiny tanpa menggunakan fungsi aktivasi Mish yang digunakan di YOLOv4. Pada bagian fusi fitur, metode YOLOv4-tiny menggunakan jaringan piramida fitur untuk mengekstrak feature map dengan skala yang berbeda untuk meningkatkan kecepatan deteksi objek, tanpa menggunakan pemadatan pola spasial dan jaringan agregasi jalan yang digunakan pada metode YOLOv4. Pada saat yang sama, YOLOv4-tiny menggunakan dua skala map fitur yang berbeda, yaitu 13x13 dan 26x26 untuk memprediksi hasil deteksi.

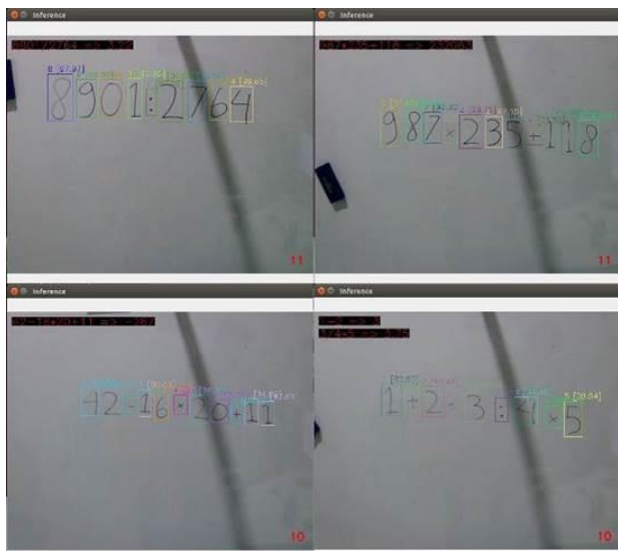
Struktur jaringan dan tata letak lapisan dari YOLOv4-tiny ditunjukkan pada Gambar 6 dan Gambar 7.

### C. Akuisisi Dataset

Penulis menggunakan 2 dataset, yaitu dataset kustom yang diambil dari Deteksi Angka dan Operator Aritmatika Menggunakan YOLO untuk Menghitung Operasi Matematika Sederhana pada Smart Whiteboard yang dapat dilihat pada Gambar 9 dalam bentuk gambar tangan (a, b, c, n, x, y, z), angka, dan operator aritmatika (+, -, ×, :, =) pada papan tulis dengan jumlah total 346 gambar dan dataset foto tangan pada Math Object Detection yang dapat dilihat pada Gambar 8 Dataset dengan jumlah total 101 gambar sehingga jumlah totalnya menjadi 447 gambar. Dalam proses ini, akan diperoleh 22 kelas yang mewakili setiap huruf, angka, dan operator aritmatika tangan. Proses ini dilakukan dengan menuliskan sejumlah huruf, angka, dan operator matematika



Gambar 10. Diagram Blok Implementasi.



Gambar 11. Keluaran Deteksi Pengujian Berdasarkan Jumlah Operator dan Digit.

yang akan digunakan sebagai kelas pada media papan tulis kemudian mengambil gambar. Proses ini diulang beberapa kali untuk setiap kelas. Gambar tangan dalam bentuk tangan diambil menggunakan kamera ponsel dengan posisi di depan dan bawah papan tulis pada jarak dan jarak dekat dan dalam setiap gambar yang diambil ada satu kelas atau beberapa kelas.

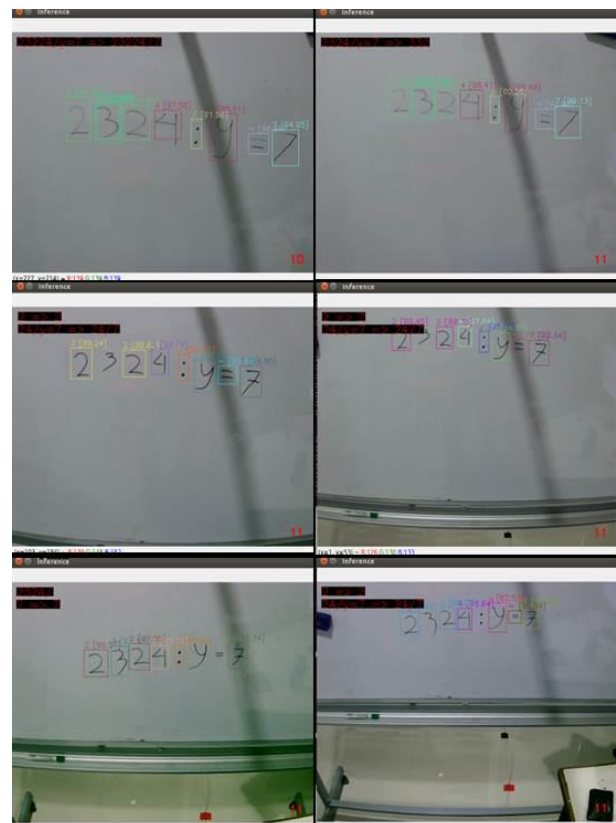
**D. Pelabelan Dataset**

Pemasangan label bertujuan untuk memberikan informasi berupa nama kelas dengan menyediakan kotak batas objek yang akan dideteksi pada setiap gambar yang telah dikumpulkan. Pemasangan label dilakukan oleh Saudara Muhammad Royhan dengan bantuan sebuah perangkat lunak yaitu *labelImg*. Proses pemasangan label dilakukan dengan 22 kelas.

Hasil pada tahap ini adalah gambar dengan kotak batas pada setiap objek yang ingin dideteksi dan koordinat kotak batas akan disimpan pada berkas dengan format txt. Distribusi total anotasi kotak batas pada setiap kelas dapat dilihat pada Tabel 1.

**E. Preproses Dataset**

Dataset yang sudah diberi label akan dipreproses, yang pada tahap ini dilakukan oleh Bapak Muhammad Royhan dengan bantuan *Roboflow*. Jenis preproses yang dilakukan adalah *resize* dan rotasi *bounding box*. Dalam proses *resize*,



Gambar 12. Keluaran Deteksi Pengujian Berdasarkan Jarak dan Kemiringan Kamera.

ukuran gambar akan diresize menjadi ukuran 416x416, ini bertujuan untuk mengurangi ukuran *file* dan mempercepat proses *training* karena banyaknya gambar dan banyaknya kelas.

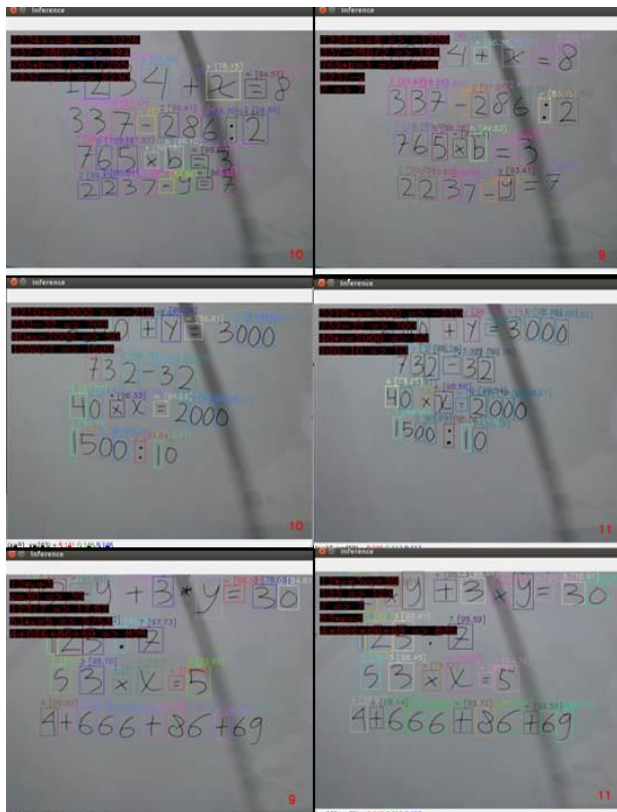
Kemudian proses augmentasi *rotasi bounding box* dimana rotasi dilakukan pada skala *bounding box*, ini bertujuan untuk menambah variasi sehingga diharapkan hasil deteksi akan lebih akurat saat gambar tidak diambil dari arah depan objek. Proses rotasi *bounding box* dilakukan dengan menggunakan sudut rotasi besar antara  $-5^\circ$  dan  $+5^\circ$ . Dalam proses ini, jumlah dataset akan dikalikan dengan 3 sehingga dataset yang awalnya berisi 447 gambar akan menjadi 1341 gambar.

**F. Training Dataset**

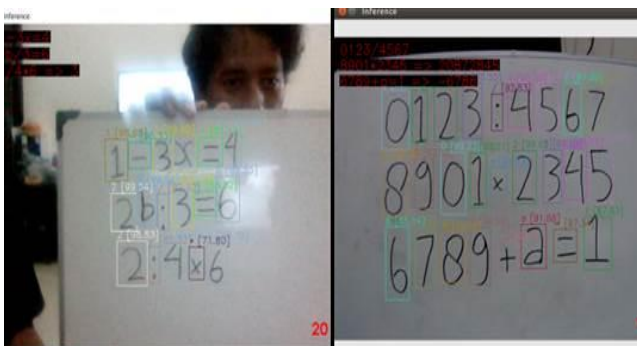
Setelah memperoleh dataset yang telah diberi label dan diproses sebelumnya, proses *training* dilanjutkan. Proses ini bertujuan untuk melatih komputer dengan pengolahan gambar dan anotasi yang telah dilakukan sehingga pola atau karakteristik dari setiap kelas yang ada terbentuk yang akan dipertimbangkan oleh komputer untuk membuat keputusan atau prediksi.

Dalam proses *training*, dibutuhkan konfigurasi hyperparameter tertentu seperti ukuran *batch*, ukuran *subdivisi*, dan *max\_batches*. Metode *YOLOv4-tiny* digunakan dalam studi ini karena *YOLOv4-tiny* adalah versi ringan dari *YOLOv4* sehingga memiliki kecepatan deteksi objek dan proses *training* yang lebih cepat dibanding *YOLOv4*. Konfigurasi yang digunakan penulis untuk proses *training* pada *YOLOv4-tiny* ditunjukkan pada Tabel 2.

Ukuran *batch* menentukan jumlah gambar yang akan diproses dalam satu iterasi. Ukuran *subdivisi* bertujuan untuk membagi ukuran *batch* menjadi *mini batch* yang kemudian akan diproses sekaligus pada GPU. Nilai lebar dan tinggi



Gambar 13. Keluaran Deteksi Pengujian Berdasarkan Penulis yang Berbeda.



Gambar 14. Keluaran Deteksi Pengujian Berdasarkan hardware yang Berbeda.

adalah dimensi gambar yang akan diajarkan. Karena jaringan YOLO

mengurangi sampel masukan sebesar 32, perlu diperhatikan bahwa lebar dan tinggi harus merupakan kelipatan dari 32, dan dalam penelitian ini lebar dan tinggi yang digunakan adalah 416 x 416. Kelas adalah jumlah kelas yang ingin dideteksi. Nilai *max\_batches* adalah batas iterasi untuk *training*. Jika iterasi sudah mencapai 44000, proses *training* akan berhenti secara otomatis. Nilai *steps* adalah nilai *checkpoint* (jumlah iterasi) di mana skala pembelajaran akan diterapkan, dalam penelitian ini nilai step yang digunakan adalah 35200 dengan skala pembelajaran yang diterapkan adalah 0,1 dan 39600 dengan skala pembelajaran yang diterapkan adalah 0,1. Nilai *filter* diperoleh karena *YOLOv4-tiny* mendeteksi 3 kotak per sel fitur dan untuk setiap kotak akan memprediksi probabilitas kelas + lebar + tinggi + x + y + *confidence*. Nilai *max\_batches*, *steps*, dan *filters* diperoleh dengan cara berikut:

1.  $max\_batches = number\_class \times 2000$
2.  $steps = (80\% \times max\_batches), (90\% \times max\_batches)$
3.  $filters = (number\_class + 5) \times 3$

Tabel 1. Annotasi Kotak Batas Dalam Setiap Kelas

Kelas	Jumlah Anotasi
*	310
+	506
-	494
/	276
0	326
1	380
2	350
3	326
4	326
5	375
6	326
7	311
8	341
9	325
=	626
a	267
b	266
c	266
n	270
x	265
y	257
z	258

Tabel 2. Konfigurasi pada YOLOv4-tiny

Konfigurasi	Informasi
batch	64
subdivision	16
width	416
height	416
max_batches	44000
steps	35200, 39600
scales	.1, .1
classes	22
filters	81

Dalam penelitian ini, ada perangkat laptop yang digunakan untuk melakukan proses *training* dengan spesifikasi seperti yang ditunjukkan pada Tabel 3.

G. Pengujian dan Output

Proses ini dimulai dengan melakukan uji coba pada model (bobot) hasil *training* yang telah diperoleh menggunakan dataset Testing sehingga didapatkan nilai TP, FP, dan FN. Setelah itu, uji coba juga dilakukan dengan memasukkan gambar atau video, baik video rekaman maupun video real-time. Kemudian dilakukan deteksi dan klasifikasi huruf tangan, angka, dan operator aritmatika pada gambar atau video tulisan pada papan tulis yang telah dimasukkan menggunakan model yang sebelumnya diperoleh. Ukuran tulisan tangan pada penelitian ini memiliki tinggi rata-rata 4 cm. Setelah gambar atau video dimasukkan, akan didapatkan dimensi objek yang terdeteksi. Dengan dimensi objek, yaitu huruf tangan, angka, dan operator aritmatika, keluaran dari sistem akan memberikan bounding box untuk objek yang terdeteksi. Kemudian juga akan diberikan label pada bounding box tersebut.

H. Implementasi Jetson Nano

Pada Gambar 10 yang menunjukkan bagian implementasi dari Gambar 1 secara detail, setelah melakukan uji coba dan mendapatkan hasil *output* berupa hasil deteksi, berat yang diuji masuk ke *Jetson Nano*, setelah itu dibuat program yang melakukan proses pembentukan operasi matematika yang dimulai dengan menentukan titik koordinat bounding box objek tulisan tangan yang terdeteksi. Setelah diketahui koordinat bounding box, bounding box akan diurutkan

Tabel 3.  
Spesifikasi Laptop

Prosesor	AMD RYZEN 5600H
RAM	8 GB SO-DIMM DDR4-3200 dan 8 GBSO-DIMM DDR4-2400
Penyimpanan	512 GB SSD M.2 2280 PCIe 3.0x4NVMe
Grafik	NVIDIA GeForce GTX 1650 4GBGDDR6
Sistem Operasi	Windows 11 Home

Tabel 4.  
Evaluasi Deteksi Pengujian Berdasarkan Jumlah Operator dan Digit

Kategori	Jumlah Objek pada Gambar	Objek Terdeteksi Benar	Operasi Matematika yang Terbentuk Benar
1 Operator 4 Digit Angka	9	9	1
2 Operator 3 Digit Angka	11	11	1
3 Operator 2 Digit Angka	11	11	1
4 Operator 1 Digit Angka	9	8	0

Tabel 5.  
Evaluasi Deteksi Pengujian Berdasarkan Jarak dan Kemiringan Kamera

Kategori	Jumlah Objek pada Gambar	Objek Terdeteksi Benar	Operasi Matematika yang Terbentuk Benar
20° dan 70 cm	8	7	0
20° dan 90 cm	8	8	1
30° dan 70 cm	8	7	0
30° dan 90 cm	8	7	0
45° dan 70 cm	8	6	0
45° dan 90 cm	8	7	0

Tabel 6.  
Evaluasi Deteksi Pengujian Berdasarkan Penulis yang Berbeda

Kategori	Jumlah Objek pada Gambar	Objek Terdeteksi Benar	Operasi Matematika yang Terbentuk Benar
Penulis 1 Pengulangan 1	32	32	4
Penulis 1 Pengulangan 2	32	31	3
Penulis 1 Pengulangan 3	32	31	3
Penulis 2 Pengulangan 1	33	32	3
Penulis 2 Pengulangan 2	33	32	3
Penulis 2 Pengulangan 3	33	32	3
Penulis 3 Pengulangan 1	33	30	1
Penulis 3 Pengulangan 2	33	31	3
Penulis 3 Pengulangan 3	33	32	3

berdasarkan koordinat mulai dari koordinat y terlebih dahulu, kemudian dilanjutkan dengan koordinat x. Jika jarak antara bounding box tidak melebihi nilai ambang yang sudah ditentukan maka mereka akan masuk ke dalam satu kelompok, jika melebihi nilai ambang tersebut maka akan dibentuk kelompok baru dimana nilai ambang pada koordinat x sebesar 53 pixel dan pada koordinat y sebesar 33 pixel. Jika terbentuk operasi matematika dari urutan bounding box, maka akan dilakukan perhitungan dan hasil operasi matematika akan ditampilkan dalam bentuk teks beserta operasi matematika di sebelah kiri gambar atau video. Jika tidak terbentuk operasi matematika, hanya objek tulisan tangan huruf, angka, dan operator matematika yang terdeteksi yang akan ditampilkan yang sudah diurutkan berdasarkan koordinat.

### III. HASIL DAN DISKUSI

Dalam penelitian ini, akan dilakukan pengujian dan analisis untuk hasil deteksi operator angka dan aritmatika untuk menghitung operasi matematika sederhana yang dikembangkan. Uji coba dilakukan menggunakan *webcam* dan *Jetson Nano 4gb*.

#### A. Tes Deteksi berdasarkan Jumlah Operator dan Digit

Dalam menguji deteksi berdasarkan jumlah operator dan digit dengan *whiteboard*, tujuannya adalah untuk mengetahui apakah model dapat menghitung operasi matematika yang

memiliki banyak operator dan digit dan sesuai dengan aturan prioritas operator aritmatika atau tidak.

Performa model pada skenario tes ini disimpulkan pada Tabel 4. Gambar 11 menunjukkan hasil deteksi model pada beberapa gambar yang diambil.

#### B. Tes Deteksi berdasarkan Jarak dan Kemiringan Kamera terhadap Papan Tulis

Dalam tes deteksi berdasarkan jarak dan sudut kamera dengan papan tulis putih, bertujuan untuk menentukan akurasi model dalam mendeteksi operasi matematika pada berbagai jarak bersama dengan kemiringan atau sudut kamera. Kamera yang digunakan adalah kamera *webcam*.

Hasil *output* dari hasil deteksi berupa kotak pembatas pada objek yang diakui, yaitu huruf tangan, angka, dan operator aritmatika dan nilai *confidence*. Kemudian dari objek tangan akan didapatkan operasi matematika dan hasilnya.

Kinerja model pada skenario uji ini disimpulkan dalam Tabel 5. Gambar 12 menunjukkan hasil deteksi model pada beberapa gambar yang diambil.

#### C. Tes Deteksi berdasarkan Penulis yang Berbeda

Tes deteksi berdasarkan berbagai penulis bertujuan untuk menentukan akurasi model dalam mendeteksi tulisan tangan yang ditulis oleh orang-orang yang berbeda. Ada 4 penulis yang digunakan dalam tes ini dimana setiap tulisannya diulang 3 kali untuk setiap penulis. Kinerja model pada skenario tes ini disimpulkan dalam Tabel 6. Gambar 13

Tabel 7.

Evaluasi Deteksi Pengujian Berdasarkan Hardware yang Berbeda	
Kategori	Rata-rata FPS (Frames Per Second)
Laptop	20
Jetson Nano	10

Tabel 8.

Evaluasi Matriks dari YOLOv4-tiny						
EPOCH	4000	8000	12000	16000	20000	24000
TP	2233	2236	2238	2238	2238	2237
FP	16	17	3	4	2	4
FN	6	3	1	1	1	2

EPOCH	28000	32000	36000	40000	44000
TP	2238	2237	2239	2239	2239
FP	1	4	1	1	1
FN	1	2	0	0	0

menunjukkan hasil deteksi model pada beberapa gambar yang diambil.

D. Tes Deteksi berdasarkan Hardware yang Berbeda

Dalam pengujian berdasarkan tipe perangkat keras yang berjalan, tujuannya adalah untuk menemukan perbandingan FPS ketika sistem dijalankan pada laptop dan Jetson Nano.

Kinerja model pada skenario pengujian ini disimpulkan dalam Tabel 7. Gambar 14 menunjukkan hasil deteksi model pada beberapa gambar yang diambil.

E. Hasil Pengujian

Hasil dilakukan dengan menggunakan model (bobot) yang diperoleh dari proses *training*. Dalam penelitian ini, jumlah dataset yang digunakan untuk proses *training* adalah 1341 gambar yang terdiri dari 1207 gambar untuk data *training* dan 134 gambar untuk data uji. Proses *training* dilakukan sampai epoch ke-44000. Uji hasil *training* dilakukan dengan mengambil data True Positives (TP), False Positive (FP), dan False Negative (FN). TP adalah objek tulisan tangan dari gambar yang berhasil terdeteksi oleh algoritma YOLO. FP adalah objek yang salah terdeteksi sebagai operator aritmatika numerik tulisan tangan. FN adalah ketika algoritma tidak dapat mengenali objek tulisan tangan operator aritmatika dalam gambar. Secara umum, semakin banyak epoch, semakin tinggi nilai TP, sementara nilai FP dan FN semakin rendah. Dari nilai TP, FP dan FN, nilai Precision, Recall, dan F1-Score dapat diperoleh. Hasil uji setiap 4000 epoch dapat dilihat pada Tabel 8. Nilai Precision, Recall, dan F1-Score dapat dihitung menggunakan rumus berikut:

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

Tabel 9.

Nilai Precision, Recall, dan F1-Score			
Epoch	Precision	Recall	F1-Score
4000	99,29%	88,73%	99,50%
8000	99,24%	99,86%	99,55%
12000	99,87%	99,95%	99,91%
16000	99,82%	99,95%	99,89%
20000	99,91%	99,95%	99,93%
24000	99,82%	99,91%	99,87%
28000	99,95%	99,95%	99,95%
32000	99,82%	99,91%	99,87%
36000	99,95%	100%	99,98%
40000	99,98%	100%	99,98%
44000	99,98%	100%	99,98%

$$F1 - Score = 2\bar{A} \frac{Precision\bar{A}-Recall}{Precision+Recall} \tag{3}$$

Berdasarkan Tabel 8, nilai TP tertinggi adalah 2239. Nilai FP dan FN terendah masing-masing adalah 1 dan 0.

Berdasarkan Tabel 9, nilai Precision, Recall, dan F1-Score terbaik masing-masing adalah 99.98%, 100%, dan 99.98%.

IV. KESIMPULAN

Berdasarkan tes yang telah dilakukan pada implementasi sistem yang dibuat dalam penelitian ini, dapat diambil beberapa kesimpulan sebagai berikut: (1) Operasi matematika yang terdeteksi dengan baik menunjukkan nilai terbaik untuk Precision, Recall, F1-Score sebesar 99,98%, 100%, dan 99,98% pada epoch-44000; (2) Model ini dapat mendeteksi beberapa digit, operator, dan operasi dengan tingkat keberhasilan sebesar 97,5% pada tes berbasis objek dan 75% pada tes berbasis operasi; (3) Tes pada jarak kurang dari 80 cm atau dengan kemiringan kamera lebih dari 30° menghasilkan akurasi rendah 0% hingga 41,67%; (4) Model ini dapat mendeteksi tulisan tangan dari penulis yang berbeda dengan tingkat keberhasilan rata-rata sebesar 96,28% pada tes berbasis objek dan 72,22% pada tes berbasis operasi, tetapi kesulitan dalam operasi yang melibatkan variabel.

DAFTAR PUSTAKA

- [1] T. Tafonao, "Peranan media pembelajaran dalam meningkatkan minat belajar mahasiswa," *J. Komun. Pendidik.*, vol. 2, no. 2, p. 103, 2018.
- [2] A. Yusdianto, A. Sukmaaji, and T. Sutanto, "Rancang bangun aplikasi interactive whiteboard untuk mendukung pembelajaran menggunakan game controlling," *J. Sist. Inf. Univ. Din.*, vol. 1, no. 2, 2012.
- [3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779-788.