

# Laporan Praktikum Kontrol Cerdas

Nama : Fryscha Febryanti Puspitasari

NIM : 224308008

Kelas : TKA 6A

Akun Github (Tautan) : <https://github.com/Fryscha>

Student Lab Assistant : Yulia Brilianty

## 1. Judul Percobaan

Introduction to Intelligent Control Systems

## 2. Tujuan Percobaan

Percobaan ini dilakukan dengan tujuan sebagai berikut.

- Memahami konsep dasar kontrol cerdas (intelligent control systems).
- Memanfaatkan GitHub untuk version control dan Kaggle sebagai sumber dataset.
- Mempelajari penerapan Computer Vision dalam sistem kontrol berbasis AI.
- Mengenali peran AI, Machine Learning (ML), dan Deep Learning (DL) dalam sistem kendali.
- Menggunakan Python dan OpenCV untuk mendeteksi objek secara sederhana.

## 3. Landasan Teori

Kontrol cerdas merupakan salah satu percabangan dari ilmu kontrol dengan menggunakan kecerdasan buatan dalam meningkatkan kinerja sistem kontrol. Sistem ini dirancang agar mampu menangani ketidakpastian dalam sistem dinamis. Selain kecerdasan buatan, kontrol cerdas dapat mencakup Machine Learning (ML) dan Deep Learning (DL).

Kecerdasan buatan (AI) yaitu bidang ilmu yang berfokus pada pengembangan sistem yang dirancang menyerupai kecerdasan manusia. Hal ini dapat meliputi pengenalan suara, pengenalan gambar, pengambilan keputusan, dan lain sebagainya. Segmen kecerdasan buatan dibagi menjadi tiga yaitu berdasarkan tingkat kemiripan, berdasarkan teknik yang digunakan, dan berdasarkan aplikasi. Kecerdasan buatan berdasarkan tingkat kemiripan dibagi menjadi tiga yakni Narrow AI, General AI, dan Superintelligence. Kemudian, kecerdasan buatan berdasarkan teknik yang digunakan dibagi menjadi lima yaitu Machine Learning (ML),

Deep Learning (DL), Natural Language Processing (NLP), Computer Vision, dan Robotic. Computer Vision merupakan salah satu segmen kecerdasan buatan berdasarkan teknik yang digunakan untuk memahami dan memproses gambar dan video. Penerapan Computer Visual dalam sistem kontrol berbasis kecerdasan buatan mencakup deteksi objek, pengolahan citra, dan penggunaan algoritma untuk meningkatkan akurasi deteksi. Lalu, terdapat kecerdasan buatan berdasarkan aplikasi yang terdiri dari kecerdasan buatan untuk bisnis, kecerdasan buatan untuk kesehatan, kecerdasan buatan untuk keuangan, kecerdasan buatan untuk transportasi, dan kecerdasan buatan untuk hiburan. Peran kecerdasan buatan dalam sistem kendali yaitu memberikan kemampuan untuk membuat keputusan berdasarkan data dan pengalaman, menangani ketidakpastian, dan beradaptasi dengan lingkungan yang berubah.

Machine Learning (ML) adalah salah satu percabangan dari kecerdasan buatan yang berfokus pada pengembangan algoritma dan teknik yang memungkinkan untuk belajar dan membuat prediksi atau keputusan berdasarkan data. Prinsip dasar dari Machine Learning terdiri dari data, model, fungsi kerugian (loss function), dan optimisasi. Metode Machine Learning terbagi menjadi tiga bagian yaitu Supervised Learning, Unsupervised Learning, dan Reinforcement Learning. Peran Machine Learning dalam sistem kendali yakni menggunakan algoritma untuk menganalisis data dan membuat prediksi tanpa pemrograman eksplisit. Hal ini dapat memungkinkan sistem untuk belajar dari data dan meningkatkan kinerjanya seiring waktu.

Deep Learning (DL) merupakan subbidang dari Machine Learning (ML) yang menggunakan jaringan saraf dengan banyak lapisan (deep neural networks) untuk menganalisis data dan membuat prediksi. Prinsip dasar dari Deep Learning yakni jaringan saraf yang terdiri dari neuron yang terhubung dalam lapisan yang terbagi menjadi Input Layer, Hidden Layers, dan Output Layer. Peran Deep Learning dalam sistem kendali adalah menggunakan jaringan saraf dengan banyak lapisan untuk memproses data yang kompleks.

Python adalah salah satu bahasa pemrograman yang digunakan dalam pengembangan sistem kontrol cerdas. Bahasa pemrograman ini sangat diminati oleh pengguna karena kemudahan penggunaannya dan banyaknya pustaka yang tersedia. Selain itu, terdapat OpenCV sebagai pustaka open-source yang digunakan dalam pengolahan citra dan Computer Vision.

GitHub yaitu salah satu platform untuk version control dalam mengembangkan perangkat lunak. Dalam kontrol cerdas, platform ini mampu melakukan kolaborasi antar peneliti dan pengembang untuk berbagi dataset dan model yang digunakan dalam percobaan

atau penelitian, mengelola kode sumber secara efisien, dan melacak perubahan dan revisi dalam proyek.

Kaggle merupakan salah satu platform yang menyediakan berbagai dataset yang dapat digunakan untuk pelatihan dan pengujian model sistem kontrol cerdas. Manfaat dari Kaggle yakni akses ke dataset yang beragam dan berkualitas tinggi, kompetisi yang dapat meningkatkan keterampilan dalam penerapan Machine Learning, juga komunitas yang aktif untuk berbagi pengetahuan dan teknik analisis data.

## 4. Analisis dan Diskusi

### Analisis Hasil:

- a. Apa yang terjadi saat objek berwarna merah muncul di kamera?

Jawab:

Saat objek berwarna merah muncul, sistem akan mendeteksi warna berdasarkan rentang nilai HSV yang telah ditentukan. Apabila objek berwarna merah ada dalam frame, mask akan menghasilkan nilai 255 (putih) untuk pixel yang sesuai dengan rentang ini dan 0 (hitam) untuk pixel lainnya. Kemudian, hasil dari deteksi ini kemudian digabungkan kembali dengan frame asli menggunakan operasi `bitwise_and`, sehingga hanya area yang berwarna merah yang akan terlihat dalam jendela 'Result', di mana hanya objek merah yang terlihat.

- b. Bagaimana sistem mendeteksi dan memfilter warna merah?

Jawab:

Sistem mendeteksi dan memfilter warna merah dengan langkah pertama yaitu konversi warna. Frame dari kamera yang awalnya dalam format BGR diubah ke ruang warna HSV menggunakan `cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)`. Langkah berikutnya yaitu membuat mask dengan menggunakan `cv2.inRange(hsv, lower_red, upper_red)`. Dengan hal ini, sistem membuat mask hanya mengizinkan pixel yang berada dalam rentang HSV yang ditentukan. Mask ini adalah gambar grayscale di mana pixel yang sesuai dengan rentang warna merah memiliki nilai 255 dan yang tidak sesuai memiliki nilai 0. Langkah selanjutnya yakni menggunakan operasi `cv2.bitwise_and(frame, frame, mask=mask)` yang digunakan untuk mengaplikasikan mask ke frame asli, sehingga hanya area yang berwarna merah yang ditampilkan dalam hasil.

- c. Bagaimana metode ini dapat diterapkan dalam intelligent control systems?

Jawab:

Metode ini dapat diterapkan dalam intelligent control systems dengan pengendalian robot yang mengarahkan atau melakukan tindakan berdasarkan deteksi objek berwarna tertentu. Misalnya, robot pengumpul botol dapat didesain untuk mengenali botol berwarna merah dan mengumpulkannya. Kemudian, terdapat contoh lain seperti pengenalan objek. Dalam aplikasi ini, deteksi warna dapat digunakan sebagai fitur awal untuk mengidentifikasi objek dalam gambar atau video.

### Diskusi:

- a. Bagaimana AI dapat meningkatkan sistem kontrol berbasis Computer Vision?

Jawab:

AI dapat meningkatkan sistem kontrol berbasis Computer Vision melalui beberapa cara. Pertama, pengenalan pola yang lebih baik. Algoritma Machine Learning dan Deep Learning dapat digunakan untuk melatih model dalam mengenali pola dan objek yang lebih kompleks, bukan hanya berdasarkan warna atau bentuk. Hal ini memungkinkan sistem untuk beradaptasi dengan variasi dalam data input. Kedua, pengolahan data yang lebih efisien. AI dapat membantu dalam pengolahan data secara real-time, memungkinkan sistem untuk membuat keputusan lebih cepat dan lebih akurat. Ketiga, adaptasi terhadap lingkungan. Dengan menggunakan teknik pembelajaran yang mendalam, sistem dapat belajar dari pengalaman dan meningkatkan kinerjanya seiring waktu dan beradaptasi dengan perubahan dalam lingkungan.

- b. Apa kelebihan dan kekurangan metode deteksi objek berbasis warna?

Jawab:

Kelebihan dari metode deteksi objek berbasis warna yaitu sederhana dan cepat. Metode deteksi berbasis warna relatif mudah diimplementasikan dan dapat memberikan hasil yang cepat dalam kondisi pencahayaan yang baik. Kelebihan selanjutnya yaitu biaya yang rendah. Hal ini tidak memerlukan perangkat keras yang mahal atau algoritma kompleks, sehingga lebih ekonomis untuk aplikasi sederhana.

Kekurangan dari metode deteksi objek berbasis warna yaitu sensitif terhadap pencahayaan. Perubahan dalam kondisi pencahayaan dapat mempengaruhi akurasi deteksi warna yang menyebabkan kesalahan dalam identifikasi objek. Selain itu, kekurangan berikutnya yakni keterbatasan dalam variasi warna. Objek dengan warna yang

mirip atau latar belakang yang berwarna sama dapat menyebabkan kesulitan dalam deteksi.

c. Bagaimana cara meningkatkan akurasi sistem deteksi objek?

Jawab:

Cara meningkatkan akurasi sistem deteksi objek dengan menggunakan model Deep Learning seperti Convolutional Neural Networks (CNN) untuk meningkatkan akurasi deteksi objek. Selain itu, cara lainnya ialah menggunakan teknik augmentasi data untuk meningkatkan variasi dalam dataset pelatihan, sehingga model dapat belajar dari lebih banyak contoh.

## 5. Assignment

Tugas assignment yaitu modifikasi kode program untuk mendeteksi warna lain (misalnya: biru atau hijau) dan menambahkan fitur bounding box untuk menandai objek yang terdeteksi. Berikut ini merupakan kode program yang telah dimodifikasi.

```
import cv2
```

```
import numpy as np
```

```
# Inisialisasi kamera
```

```
cap = cv2.VideoCapture(0)
```

```
while True:
```

```
    _, frame = cap.read()
```

```
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

```
# Rentang warna dalam HSV
```

```
color_ranges = {
```

```
    'Merah': ([0, 120, 70], [10, 255, 255], (0, 0, 255)), # BGR Merah
```

```
    'Hijau': ([40, 40, 40], [80, 255, 255], (0, 255, 0)), # BGR Hijau
```

```
'Biru': ([90, 50, 50], [130, 255, 255], (255, 0, 0)) # BGR Biru
}
```

```
for color, (lower, upper, bbox_color) in color_ranges.items():
    lower_np = np.array(lower, np.uint8)
    upper_np = np.array(upper, np.uint8)
    mask = cv2.inRange(hsv, lower_np, upper_np)

    # Temukan kontur
    contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    for contour in contours:
        if cv2.contourArea(contour) > 500: # Filter area kecil
            x, y, w, h = cv2.boundingRect(contour)
            cv2.rectangle(frame, (x, y), (x + w, y + h), bbox_color, 2)
            cv2.putText(frame, color, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, bbox_color, 2)

    # Menampilkan hasil hanya dalam satu tampilan
    cv2.imshow("Deteksi Warna", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Penjelasan:

Terdapat import Library di bawah ini.

```
import cv2
```

```
import numpy as np
```

Kode ini berfungsi untuk mengimpor library OpenCV (cv2) untuk pengolahan citra dan NumPy (np) untuk manipulasi array.

Kemudian, terdapat inisialisasi kamera di bawah ini.

```
cap = cv2.VideoCapture(0)
```

Fungsinya untuk menginisialisasi kamera untuk menangkap video. 0 biasanya merujuk pada kamera default di perangkat.

Lalu, terdapat loop utama di bawah ini

```
while True:
```

```
    _, frame = cap.read()
```

```
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

Dalam loop ini, program terus membaca frame dari kamera. Setiap frame kemudian diubah dari format BGR (Blue, Green, Red) ke format HSV (Hue, Saturation, Value) menggunakan cv2.cvtColor(). Format HSV lebih efektif untuk deteksi warna.

Selanjutnya, terdapat rentang warna di bawah ini.

```
color_ranges = {
```

```
    'Merah': ([0, 120, 70], [10, 255, 255], (0, 0, 255)), # BGR Merah
```

```
    'Hijau': ([40, 40, 40], [80, 255, 255], (0, 255, 0)), # BGR Hijau
```

```
    'Biru': ([90, 50, 50], [130, 255, 255], (255, 0, 0)) # BGR Biru
```

```
}
```

Kode ini mendefinisikan rentang warna untuk merah, hijau, dan biru dalam format HSV. Setiap warna memiliki batas bawah dan atas, serta warna untuk bounding box yang akan ditampilkan.

Setelah itu, terdapat deteksi warna di bawah ini.

```
for color, (lower, upper, bbox_color) in color_ranges.items():
```

```
    lower_np = np.array(lower, np.uint8)
```

```
upper_np = np.array(upper, np.uint8)
```

```
mask = cv2.inRange(hsv, lower_np, upper_np)
```

Loop ini iterasi melalui setiap warna yang didefinisikan. `cv2.inRange()` digunakan untuk membuat mask yang menunjukkan area dalam frame yang sesuai dengan rentang warna yang ditentukan.

Kemudian, proses menemukan kontur.

```
contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
```

```
for contour in contours:
```

```
    if cv2.contourArea(contour) > 500: # Filter area kecil
```

```
        x, y, w, h = cv2.boundingRect(contour)
```

```
        cv2.rectangle(frame, (x, y), (x + w, y + h), bbox_color, 2)
```

```
        cv2.putText(frame, color, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, bbox_color, 2)
```

Kontur dari mask yang dihasilkan ditemukan menggunakan `cv2.findContours()`. Hanya kontur dengan area lebih besar dari 500 piksel yang diproses lebih lanjut. Untuk setiap kontur yang valid, bounding box Digambar di sekitar objek yang terdeteksi menggunakan `cv2.boundingRect()`, dan nama warna ditambahkan di atas bounding box tersebut.

Selanjutnya, menampilkan hasil.

```
cv2.imshow("Deteksi Warna", frame)
```

Hasil akhir ditampilkan dalam satu jendela dengan nama “Deteksi Warna”. Frame yang ditampilkan mencakup objek yang terdeteksi dengan bounding box dan label warna.

Menutup program

```
if cv2.waitKey(1) & 0xFF == ord('q'):
```

```
    break
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```

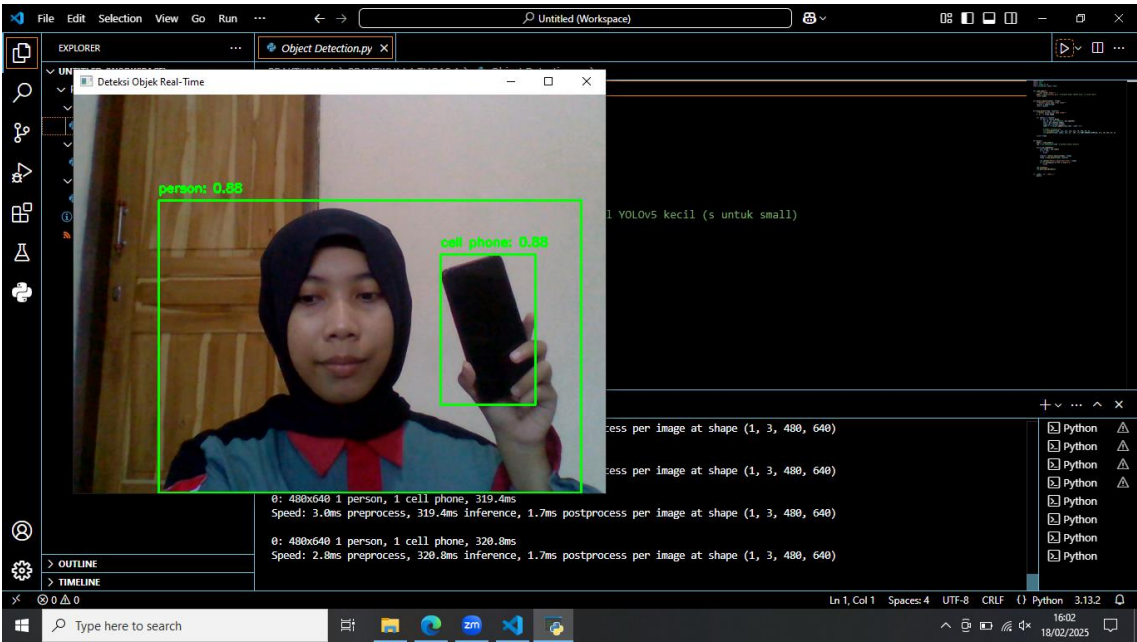


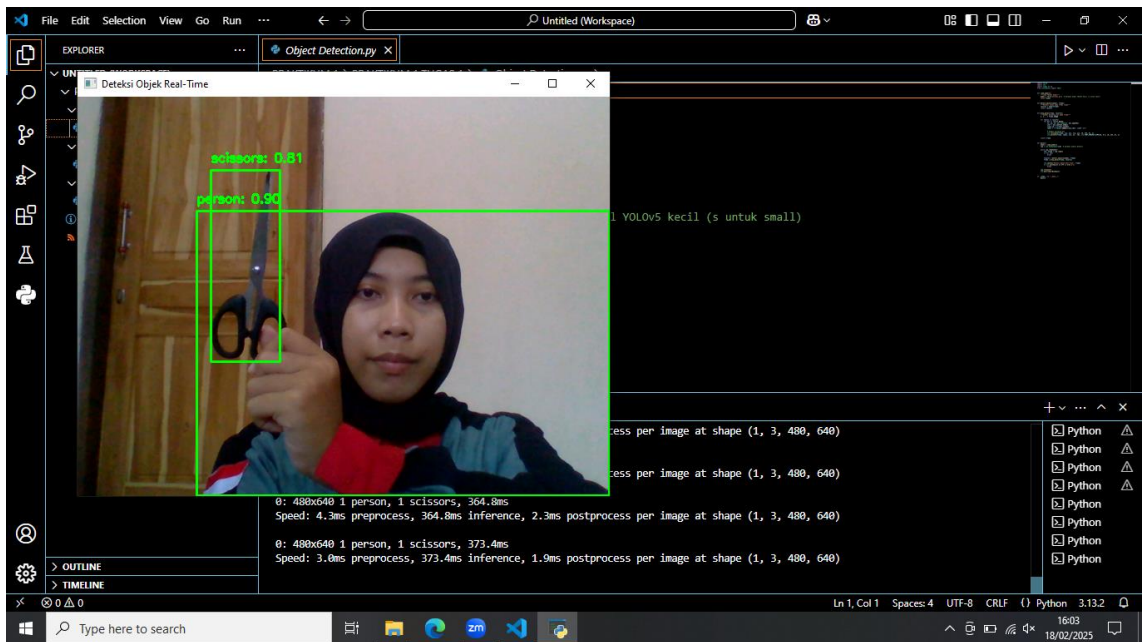
Program akan terus berjalan hingga tombol ‘q’ ditekan. Setelah itu, kamera dilepaskan dan semua jendela ditutup.

Hasil kode program yaitu program berjalan sesuai dengan perintah dan mampu mendeteksi warna yaitu merah, biru, dan hijau. Bounding box berwarna di sekitar objek yang terdeteksi (merah, biru, dan hijau) menandai lokasi objek dalam frame. Nama warna ditampilkan di atas bounding box yang memberikan informasi visual yang jelas tentang warna yang terdeteksi.

## 6. Data dan Output Hasil Pengamatan

Sajikan data dan hasil yang diperoleh selama percobaan. Gunakan tabel untuk menyajikan data jika diperlukan.

No	Variabel	Hasil Pengamatan
1	Person & Cell Phone	<p>Bounding box menandai adanya objek yang terdeteksi yaitu person dengan akurasi 0.88 dan cell phone dengan akurasi 0.88.</p> 
2	Scissors	<p>Bounding box menandai adanya objek yang terdeteksi yaitu person dengan akurasi 0.90 dan scissors dengan akurasi 0.81.</p>



3

Bottle

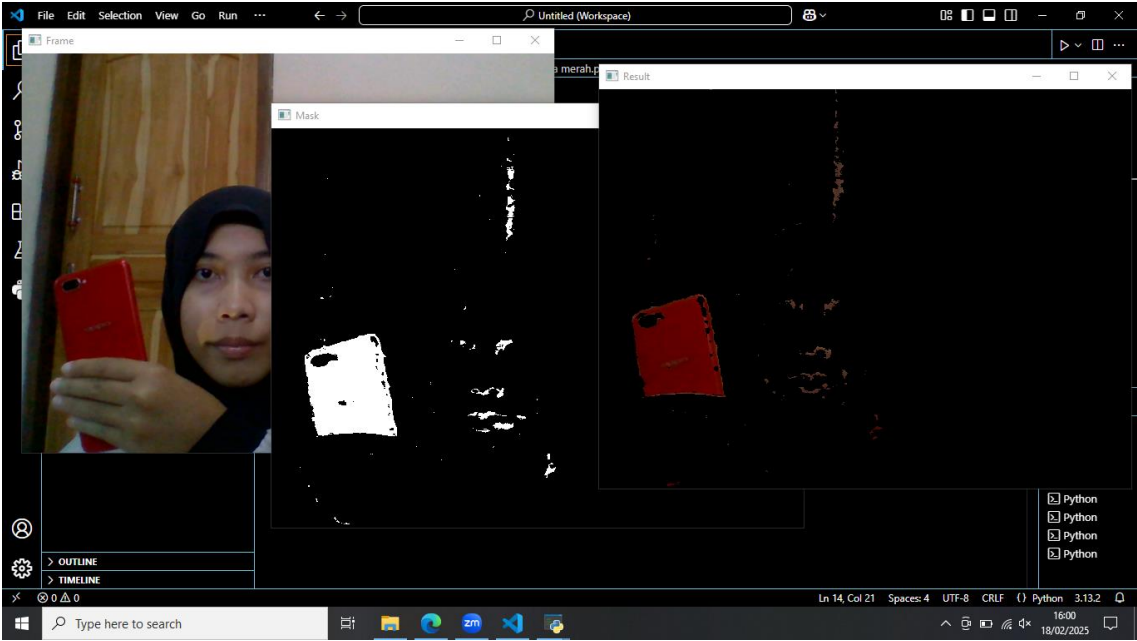
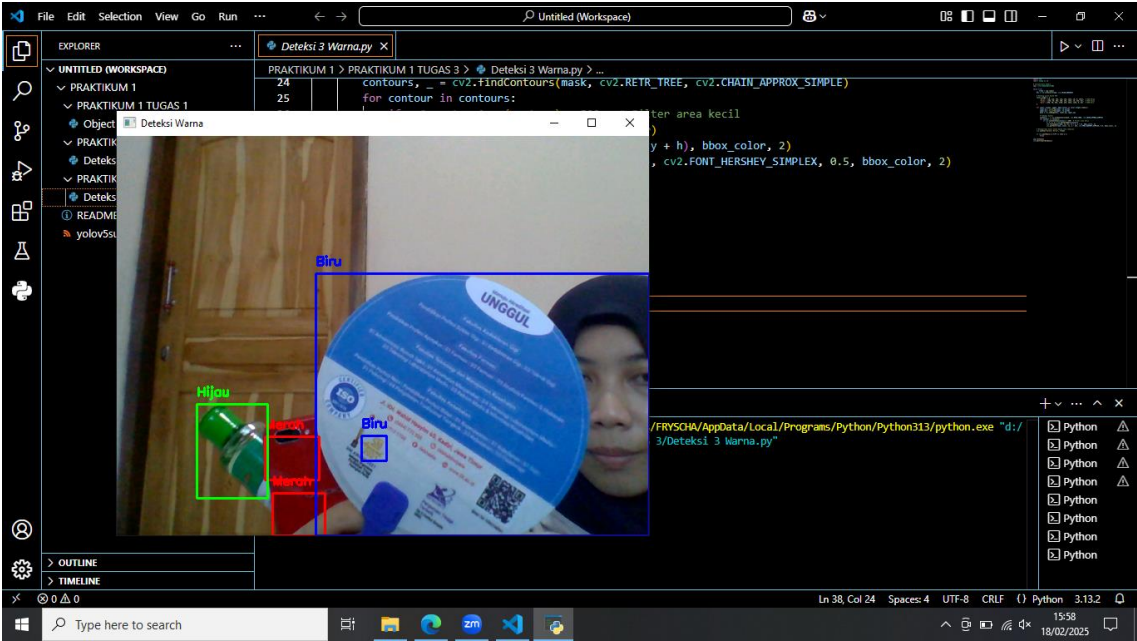
Bounding box menandai adanya objek yang terdeteksi yaitu person dengan akurasi 0.92 dan bottle dengan akurasi 0.73.

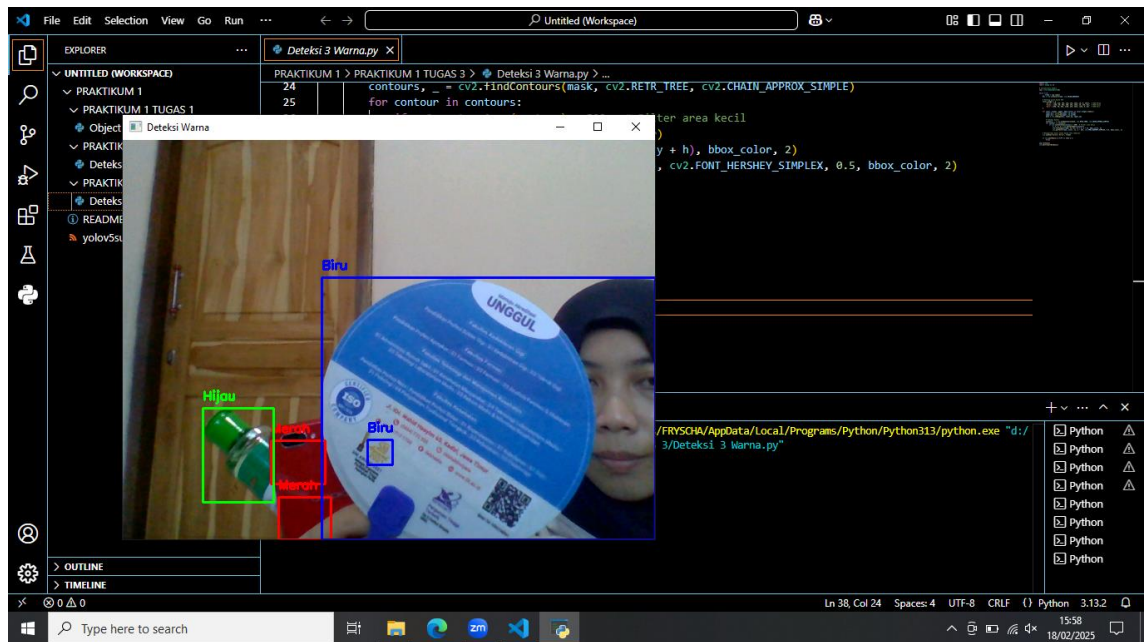


4

Warna Merah

Adanya objek yang terdeteksi yaitu warna merah pada bagian 'Result'.

		
5	Warna Biru	<p>Bounding box menandai adanya objek yang terdeteksi yaitu berwarna biru.</p> 
6	Warna Hijau	<p>Bounding box menandai adanya objek yang terdeteksi yaitu berwarna hijau.</p>



## 7. Kesimpulan

Berdasarkan percobaan dan analisis data dapat disimpulkan bahwa

- Kontrol cerdas merupakan salah satu percabangan dari ilmu kontrol dengan menggunakan kecerdasan buatan dalam meningkatkan kinerja sistem kontrol.
- Kecerdasan buatan (AI) yaitu bidang ilmu yang berfokus pada pengembangan sistem yang dirancang menyerupai kecerdasan manusia.
- Machine Learning (ML) adalah salah satu percabangan dari kecerdasan buatan yang berfokus pada pengembangan algoritma dan teknik yang memungkinkan untuk belajar dan membuat prediksi atau keputusan berdasarkan data.
- Deep Learning (DL) merupakan subbidang dari Machine Learning (ML) yang menggunakan jaringan saraf dengan banyak lapisan (deep neural networks) untuk menganalisis data dan membuat prediksi.
- Dengan menggunakan rentang warna dalam format HSV, sistem dapat mendeteksi objek dengan akurasi yang lebih baik dibandingkan dengan menggunakan format BGR.

## 8. Saran

Lebih menambahkan variasi dalam deteksi warna, seperti akurasi ketepatan warna yang terdeteksi pada objek.

## 9. Daftar Pustaka

- Muttaqin, Arafah M., Jaya A.K., Suryawan M.A., Gustiana Z., Banjarnahor A.R., Bukidz D.P., Hazriani, Simanjuntak M., Saputra N., Fajrillah. 2023. *Implementasi Artificial Intelligence (AI) dalam Kehidupan*. Cetakan ke-1. Yayasan Kita Menulis. Langsa.
- Santoso J.T., 2023. *KECERDASAN BUATAN (Artificial Intelligence)*. Yayasan Prima Agus Teknik Bekerja sama dengan Universitas Sains & Teknologi Komputer (Universitas STEKOM). Semarang.
- Siregar, M. H., & Mulyana, D. I. (2024). Teknologi Artificial Intelligence (AI) Vision Swift dalam Sistem Pemantauan Latihan Bulu Tangkis dengan Algoritma Optical Flow. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, 5(3), 3349–3361.  
<https://doi.org/10.35870/jimik.v5i3.1027>

---

Gunakan template ini sebagai panduan untuk menyusun laporan percobaan Anda. Pastikan untuk mengisi setiap bagian dengan informasi yang relevan dan lengkap.