

# Laporan Praktikum Kontrol Cerdas

Nama : Rhehan Adi Prakoso  
NIM : 224308094  
Kelas : TKA-7D  
Akun Github (Tautan) : <https://github.com/RhehanAdi>  
Student Lab Assistant : -

## 1. Judul Percobaan

*Week 3: Deep Learning for Intelligent Control Systems.*

## 2. Tujuan Percobaan

Tujuan dari dilakukannya praktikum “*Deep Learning for Intelligent Control Systems*” adalah:

- a) Memahami konsep *Deep Learning* dalam sistem kendali.
- b) Mengimplementasikan *Convolutional Neural Network* (CNN) untuk klasifikasi objek.
- c) Menggunakan *TensorFlow* dan Keras untuk membangun model *Deep Learning*.
- d) Mengintegrasikan model CNN dengan *Computer Vision* untuk deteksi objek secara *real-time*.
- e) Menggunakan dataset dari Kaggle untuk pelatihan model.
- f) Mengembangkan mode *Night Vision* untuk deteksi objek dalam kondisi pencahayaan rendah.

## 3. Landasan Teori

- A. ***Deep Learning* (DL)** adalah cabang dari *Machine Learning* (ML) yang menggunakan jaringan saraf tiruan berlapis-lapis (*artificial neural networks*) untuk mempelajari representasi data secara otomatis. Pada DL, data mentah dapat diproses melalui beberapa lapisan neuron yang saling terhubung, dimana setiap lapisan bertugas untuk mengekstraksi fitur yang semakin kompleks dari data tersebut (Santoso, 2023). Proses pelatihan DL dilakukan dengan *backpropagation* dan *gradient descent*, sehingga bobot pada setiap neuron diperbarui secara bertahap untuk meminimalisir kesalahan prediksi. Dengan begitu model mampu mengenali pola yang sangat kompleks dan non-linear

sehingga banyak diterapkan pada sistem yang membutuhkan adaptasi terhadap suatu perubahan seperti *face recognition* dan *autonomous vehicles* (Hastuti *et al.*, 2025).

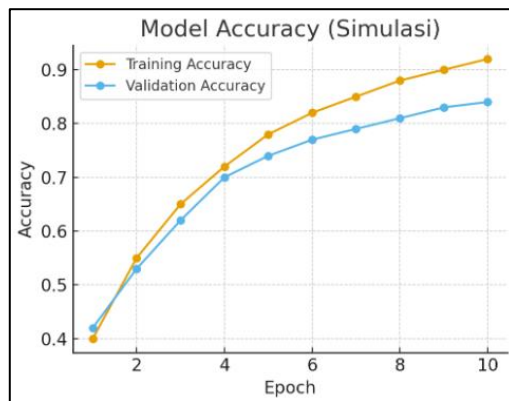
- B. ***Convolutional Neural Network (CNN)*** adalah jenis jaringan saraf tiruan yang dikembangkan khusus untuk memproses data berbentuk *grid*, seperti citra dua dimensi. CNN mampu mengenali pola dan fitur dalam data secara otomatis, tanpa memerlukan ekstraksi fitur manual, sehingga efektif untuk tugas pengolahan citra dan komputer vision. Dengan kemampuannya mempelajari representasi kompleks dari data *input*, CNN banyak diterapkan dalam pengenalan wajah, deteksi objek, klasifikasi citra, dan berbagai sistem berbasis visual lainnya (Manza *and* Wayahdi, 2025). Dibalik CNN yang efektif dalam melakukan pengelolaan citra, CNN membutuhkan jumlah data yang besar sehingga model dapat belajar representasi fitur dengan baik dan menghindari *overfitting* (Model belajar terlalu detail pada data pelatihan sehingga mampu memprediksi data *training* dengan sangat baik, namun gagal menggeneralisasi pada data baru atau data uji) yang menurunkan performa DL (Shalev-Shwartz *and* Ben-David, 2014).
- C. ***Convolutional Layer*** adalah komponen utama dalam CNN yang bertugas untuk mengekstraksi fitur dari data input, seperti citra. Convolutional layer menggunakan filter atau kernel yang bergerak melintasi *input* untuk mendeteksi pola lokal, seperti tepi, garis, atau tekstur. Hasil konvolusi menghasilkan *feature map*, yang merepresentasikan lokasi dan keberadaan fitur tertentu. Melalui fitur tersebut *convolutional layer* memungkinkan untuk mengenali pola secara otomatis tanpa perlu ekstraksi fitur eksternal.
- D. ***Pooling Layer*** adalah lapisan dalam CNN yang berfungsi untuk mengurangi dimensi *feature map* dari *convolutional layer* sebelumnya sambil mempertahankan informasi penting. *Pooling layer* bekerja dengan mengambil nilai maksimum (*max pooling*) atau rata-rata (*average pooling*) dari area kecil pada *feature map* sehingga membantu menyederhanakan representasi data, mengurangi komputasi, dan meningkatkan ketahanan model terhadap pergeseran atau rotasi pada citra. Dengan *pooling layer*, CNN dapat fokus pada fitur yang paling signifikan tanpa kehilangan konteks penting dalam data.

- E. **Fully Connected Layer** adalah lapisan pada CNN yang menghubungkan setiap neuron dari lapisan sebelumnya dengan semua neuron pada lapisan ini. *Fully connected layer* berfungsi menggabungkan fitur-fitur yang telah diekstraksi oleh *convolutional* dan *pooling layer* untuk melakukan prediksi akhir, seperti klasifikasi atau regresi. *Fully connected layer* menerjemahkan representasi fitur yang kompleks menjadi *output* yang sesuai dengan tujuan model, misalnya menentukan kelas citra dalam tugas klasifikasi.
- F. **Activation function** adalah fungsi matematika yang diterapkan pada output neuron untuk menentukan apakah neuron tersebut “aktif” atau tidak. Fungsi ini memungkinkan jaringan saraf untuk menangkap hubungan *non-linear* dalam data, sehingga model dapat mempelajari pola yang kompleks. Pada CNN, *activation function* diterapkan setelah *convolutional layer* atau *fully connected layer* untuk menambahkan *non-linearitas* pada *output*. Hal tersebut dibutuhkan karena fitur yang diekstraksi dari citra sering bersifat kompleks dan tidak linear. Dengan adanya *activation function*, CNN mampu mempelajari pola yang lebih kompleks dan meningkatkan kemampuan prediksi pada tugas seperti klasifikasi citra atau deteksi objek. Beberapa *activation function* yang umum digunakan antara lain ReLU (*Rectified Linear Unit*), *sigmoid*, dan *tanh*, yang masing-masing memiliki karakteristik berbeda dalam memetakan *input* menjadi *output* dan mempengaruhi proses pelatihan jaringan.
- G. **Night vision** adalah teknologi yang memungkinkan penglihatan atau penginderaan objek pada kondisi pencahayaan rendah atau gelap total. *Night vision* bekerja dengan menangkap radiasi *infrared* atau memperkuat cahaya yang minim, sehingga detail objek tetap dapat terlihat. Penggunaan *night vision* mampu diaplikasikan pada model CNN sebagai input suatu model sehingga sistem mampu mengidentifikasi objek meski dengan kondisi pencahayaan yang minim.

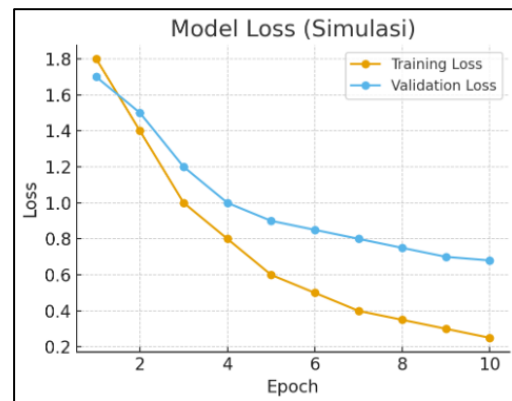
#### 4. Analisis

Pada praktikum ini dilakukan pembuatan model *deep learning* dengan *convolutional neural network* (CNN) sebagai jaringan syaraf yang digunakan pada model *deep learning* untuk mengidentifikasi dan mengklasifikasi objek berdasarkan kelasnya. Terdapat beberapa kelas yang dibuat yakni, *building*, *forest*,

*glacier, mountain, sea, dan street*. Dataset yang digunakan untuk melakukan *training* didapatkan melalui platform kaggle. Dataset tersebut berisikan +400 item foto yang berbeda sesuai dengan klasifikasi kelasnya. Dataset tersebut selanjutnya dilakukan proses *training* sehingga model mampu untuk mengidentifikasi pola yang ditangkap berdasarkan citra pada tiap kelas sehingga nantinya mampu beradaptasi terhadap perubahan. Proses training dilakukan pada Vscode dengan epoch yang digunakan sebanyak 10. Epoch adalah satu siklus penuh di mana seluruh data dalam dataset telah diproses atau digunakan untuk melatih model atau algoritma. satu epoch berarti setiap sampel dalam dataset telah dilalui sekali untuk mempelajari pola atau hubungan antar data. Grafik hubungan epoch dengan model akurasi dan model *loss* terlihat pada gambar 1 dan gambar 2 dibawah ini



Gambar 1. Hubungan Jumlah Epoch dengan Accuracy Model



Gambar 2. Hubungan Jumlah Epoch dengan Loss Model

Berdasarkan gambar 1. Hubungan antara epoch dengan akurasi berbanding lurus. Semakin tinggi jumlah epoch yang digunakan maka semakin tinggi juga nilai akurasi pada model tersebut. Berdasarkan gambar 2. Hubungan antara epoch dengan tingkat loss berbanding terbalik. Semakin kecil jumlah epoch yang digunakan maka semakin tinggi nilai loss nya, sebaliknya semakin banyak epochnya maka nilai loss model semakin kecil. Ketika proses *training* selesai dilakukan maka akan memunculkan file h5 yang merupakan hasil training dari dataset.

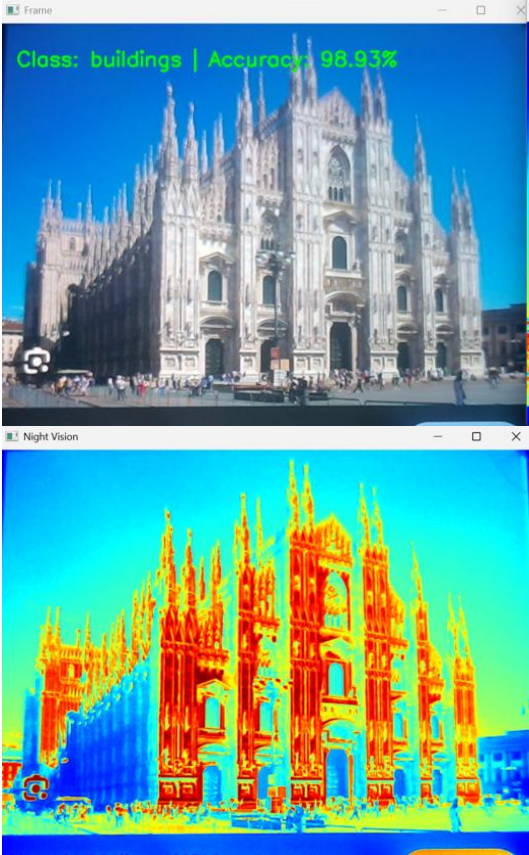
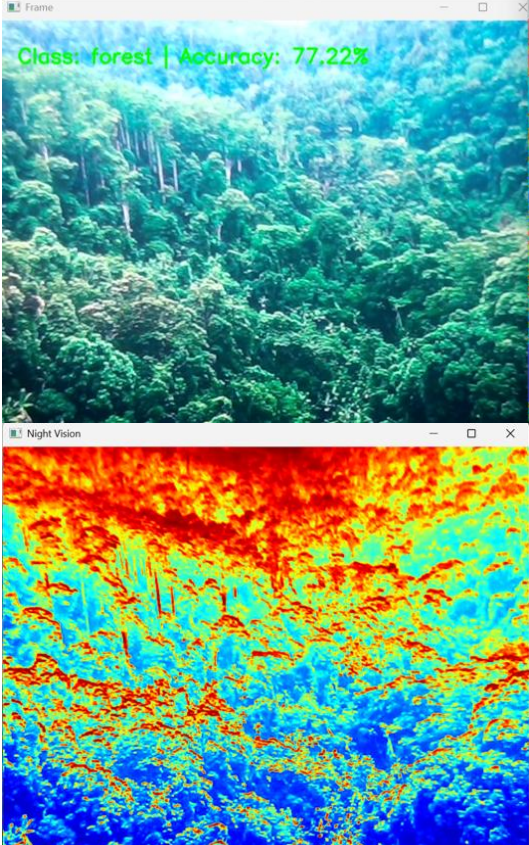
Selanjutnya proses pengujian dilakukan dengan membuat program membaca file dataset yang telah dilakukan training sebelumnya. Proses pengujian dilakukan dengan menggunakan *webcam* yang disorotkan dengan beberapa item foto sesuai dengan klasifikasinya. Berdasarkan hasil pengujian model mampu mengidentifikasi

dan mengklasifikasi objek yang ditangkap dengan benar. Namun pada kondisi tertentu model melakukan klasifikasi objek secara tidak tepat. Khususnya pada kelas *glacier*. Kondisi tersebut terjadi karena keterbatasan dataset yang digunakan. Model *deep learning* memiliki kelemahan yakni penggunaan dataset yang banyak untuk menghasilkan kepresisian yang tinggi. Sehingga untuk meningkatkan keakuratan perlu adanya penambahan dataset serta penggunaan epoch yang ditingkatkan. Selain itu faktor kemiripan fitur pada setiap objek juga menjadi salah satu penyebab model salah dalam mengklasifikasi. Beberapa objek yang memiliki fitur visual yang mirip misal seperti warna, bentuk, dan tekstur menyebabkan model kesulitan dalam membedakan antar kelas sehingga memungkinkan kesalahan klasifikasi pada objek. Selain itu tingkat resolusi dan pencahayaan yang digunakan juga mempengaruhi model dalam pengambilan keputusan sehingga dibutuhkan resolusi yang tinggi pada webcam serta pencahayaan yang stabil untuk mendapatkan tingkat presisi yang tinggi dalam melakukan identifikasi serta klasifikasi suatu objek berdasarkan kelasnya.

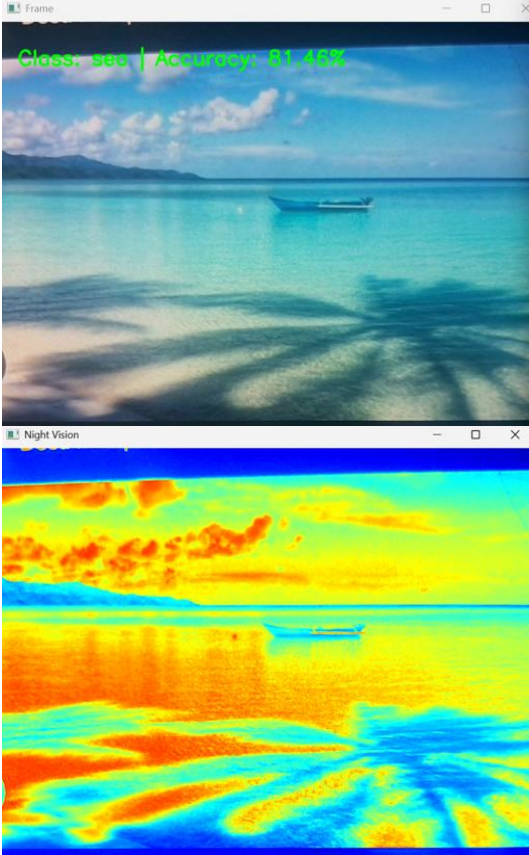
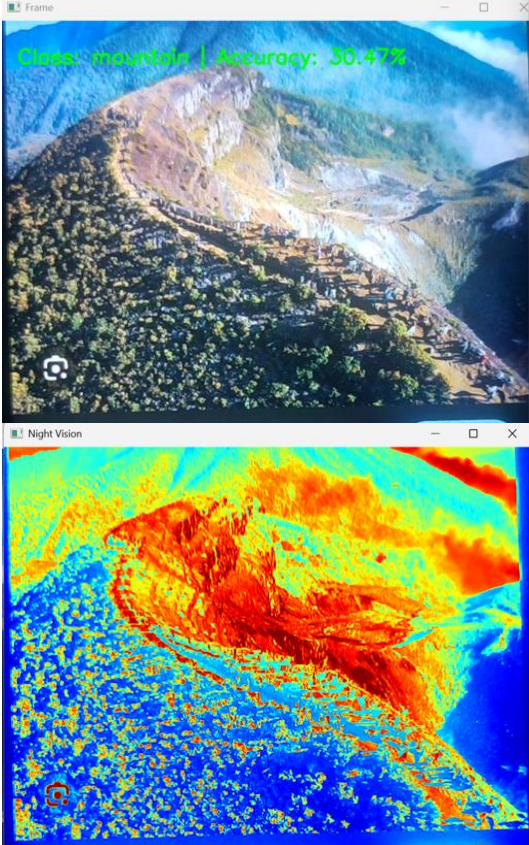
## **5. Assignment**

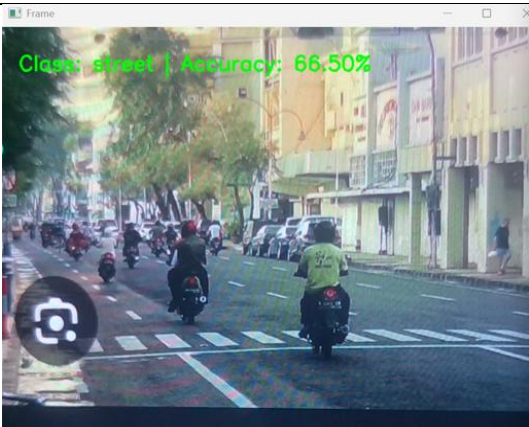

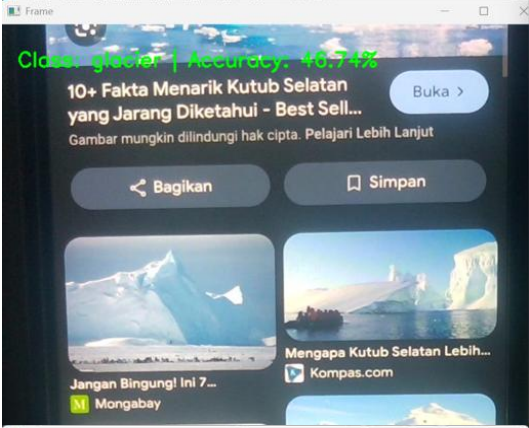

- a) Studi literatur terkait *deep learning* serta cara kerja CNN pada model *deep learning*.
- b) Melakukan *download* dataset pada kaggle untuk klasifikasi objek.
- c) Membuat pemrograman pada python dengan model *deep learning* berbasis CNN
- d) Melakukan proses training pada vscode menggunakan program yang dibuat
- e) Melakukan pengujian klasifikasi object berdasarkan kelasnya
- f) Melakukan upload hasil praktikum pada repository Github

## **6. Data dan Output Hasil Pengamatan**

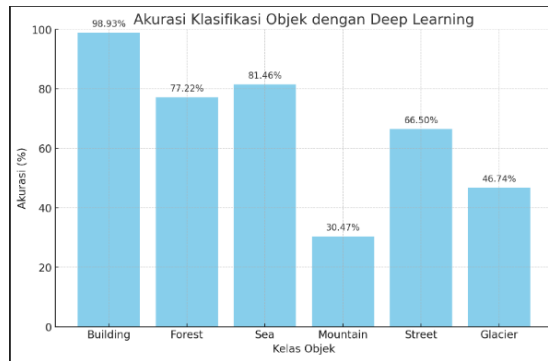
No	Kelas	Hasil Percobaan	Keterangan
1	<i>Building</i>		Objek foto berupa gereja mampu diklasifikasikan dengan benar oleh model sebagai <i>building</i> dengan tingkat akurasi 98,93%
2	<i>Forest</i>		Objek foto berupa hutan mampu diklasifikasikan dengan benar oleh model sebagai <i>forest</i> dengan tingkat akurasi 77,22%



No	Kelas	Hasil Percobaan	Keterangan
3	<i>Sea</i>		Objek foto berupa pantai mampu diklasifikasikan dengan benar oleh model sebagai <i>sea</i> dengan tingkat akurasi 81,46%
4	<i>Mountain</i>		Objek foto berupa gunung mampu diklasifikasikan dengan benar oleh model sebagai <i>mountain</i> dengan tingkat akurasi 30,47%

No	Kelas	Hasil Percobaan	Keterangan
5	Street	 	Objek foto berupa jalan raya mampu diklasifikasikan dengan benar oleh model sebagai <i>street</i> dengan tingkat akurasi 66,50%
6	Glacier	 	Objek foto berupa <i>glacier</i> mampu diklasifikasikan dengan benar oleh model sebagai <i>glacier</i> dengan tingkat akurasi 46,74%





Gambar 3. Hubungan Kelas Objek terhadap Tingkat Akurasinya

## 7. Kesimpulan

Dari praktikum yang telah dilakukan, disimpulkan bahwa pemahaman konsep *Deep Learning* sangat penting dalam penerapan sistem kendali modern, khususnya pada bidang *Computer Vision*. Implementasi *Convolutional Neural Network* (CNN) berhasil digunakan untuk klasifikasi objek dengan tingkat akurasi yang memuaskan, menunjukkan kemampuan CNN dalam mengekstraksi fitur penting dari citra secara otomatis. Penggunaan *TensorFlow* dan Keras mempermudah pembangunan dan pelatihan model *Deep Learning*. Integrasi model CNN dengan sistem *Computer Vision* memungkinkan deteksi objek secara *real-time*, sehingga model tidak hanya mampu mengenali objek dari dataset, tetapi juga memproses citra langsung dari kamera. Dataset yang diperoleh dari Kaggle terbukti efektif sebagai sumber pelatihan, menyediakan variasi objek yang cukup untuk meningkatkan generalisasi model. Selain itu, pengembangan mode Night Vision menunjukkan bahwa sistem mampu beradaptasi dalam kondisi pencahayaan rendah, memperluas kemampuan deteksi objek pada berbagai situasi lingkungan.

## 8. Saran

Pada model *deep learning*, dataset harus memiliki item yang banyak dan bervariasi dan proses *training* model nantinya dilakukan dengan epoch yang banyak juga model mampu meningkatkan akurasinya dalam melakukan klasifikasi pada suatu objek.

## 9. Daftar Pustaka

- Hastuti, S., Ansar, A., Hermawan, N., 2025. Penerapan Teknologi Deep Learning Dalam Pendidikan Digital. Jurnal Penelitian Ilmu Pendidikan Indonesia.
- Manza, Y., Wayahdi, M.R., 2025. Teknologi Kecerdasan Buatan dalam

Pengembangan Sistem Cerdas: Tantangan dan Peluang. Jurnal Teknologi.  
Santoso, J.T., 2023. KECERDASAN BUATAN (Artificial Intelligence).  
YAYASAN PRIMA AGUS TEKNIK, Semarang.  
Shalev-Shwartz, S., Ben-David, S., 2014. Understanding Machine Learning: From  
Theory to Algorithms, 1st ed. Cambridge University Press.  
<https://doi.org/10.1017/CBO9781107298019>