**Glosarium**

Berikut adalah glosarium dengan istilah umum yang digunakan pada kelas ini. Anda dapat membaca sekilas materi berikut untuk mengenali istilah-istilah umum yang ada di modul kelas ini. Selain itu, Anda juga dapat mengunjungi kembali halaman ini setiap kali menemukan istilah yang belum dimengerti. Carilah istilah tersebut pada halaman glosarium ini untuk mengidentifikasi makna atau definisinya. Jika masih terdapat kosakata yang tidak Anda pahami dan belum masuk di daftar ini, Anda dapat memberikan saran melalui fitur Laporan Materi.

**A**

**Action**

Dalam reinforcement learning, action adalah setiap keputusan yang diambil oleh agent.

**Activation Function**

Fungsi yang menerima jumlah bobot semua masukan dari layer sebelumnya, kemudian menghasilkan dan meneruskan nilai keluaran ke layer berikutnya. Contoh: ReLU atau sigmoid.

**Akurasi**

Jumlah data yang diprediksi dengan benar oleh machine learning dibagi jumlah seluruh data poin.

**Algoritma**

Dalam ilmu komputer, algoritma adalah sekumpulan aturan atau instruksi yang didesain untuk melakukan tugas dan menyelesaikan permasalahan.

**Anomaly Detection**

Proses untuk menemukan anomali pada dataset.

**Artificial Intelligence**

Program atau model yang dapat menyelesaikan tugas-tugas canggih. Machine learning adalah sub-bidang dari artificial intelligence.

**B**

**Bobot**

Koefisien untuk fitur dalam model linier atau parameter yang dipelajari oleh sebuah perceptron untuk menunjukkan kekuatan node tertentu dalam jaringan syaraf tiruan.

**C**

**Clustering**

Pengelompokan data yang memiliki kesamaan ke dalam grup tertentu.

**CSV**

Comma-Separated Value, format file teks dalam bentuk tabular dengan pemisah berupa koma

**D**

**Data Kategorik**

Fitur-fitur yang memiliki sekumpulan nilai diskrit dan bisa dibagi ke dalam grup. Sering disebut juga data diskrit.

**Data Numerik**

Fitur-fitur yang direpresentasikan sebagai bilangan bulat atau bilangan riil. Sering disebut juga fitur berkelanjutan.

**Data Time Series**

Sekumpulan data atau observasi pada interval waktu tertentu.

**Dataset**

Sekumpulan data atau contoh-contoh yang terdiri dari satu atau lebih fitur.

**Deep Learning**

Cabang machine learning dengan algoritma jaringan syaraf tiruan yang dapat belajar dan beradaptasi terhadap sejumlah besar data.

**Dimension Reduction**

Mengurangi jumlah dimensi yang digunakan untuk merepresentasikan fitur tertentu.

**E**

**Elbow**

Metode untuk menentukan jumlah cluster pada dataset.

**Epoch**

Satu proses pelatihan penuh atas seluruh dataset sehingga setiap contoh data telah melewati model sebanyak satu kali.

**Environment**

Dalam reinforcement learning, environment adalah sarana untuk berinteraksi, yang dapat menerima action dan memberikan respon berupa hasil maupun data berupa satu set observasi baru.

**F**

**Feedback**

Dalam machine learning, situasi di mana prediksi model mempengaruhi data pelatihan.

**Fitur**

Input variabel yang digunakan untuk membuat prediksi.

**G**

**Good Fit**

Kondisi saat model model machine learning melakukan generalisasi data dengan baik.

**Google Colaboratory**

Notebooks yang memungkinkan kita menulis dan mengeksekusi code pada browser.

**H**

**Hidden Layer**

Lapisan sintetis dalam jaringan saraf antara lapisan masukan (fitur) dan lapisan keluaran (prediksi). Hidden layer biasanya berisi fungsi aktivasi untuk pelatihan.

**Hyperparameter**

Variabel yang digunakan untuk mengontrol proses pelatihan model. Contohnya: epoch.

**I**

**Input Layer**

Layer pertama yang menerima data masukan dalam jaringan saraf tiruan

**K**

**Keras**

API Python machine learning yang populer. Keras bisa dijalankan pada beberapa kerangka kerja deep learning, termasuk TensorFlow yang tersedia sebagai tf.keras.

**Klasifikasi**

Tipe model machine learning untuk membedakan antara dua atau lebih kelas. Misalnya klasifikasi apakah suatu email merupakan email spam atau bukan.

**L**

**Label**

Dalam supervised learning, label adalah “jawaban” atau “hasil” dari sebuah contoh.

**Layer**

Sekumpulan neuron dalam jaringan saraf yang memproses fitur-fitur masukan atau keluaran dari neuron tersebut.

**Linear Regression**

Menggunakan keluaran y’ dari model linier sebagai prediksi aktual dalam model regresi. Tujuan dari permasalahan regresi adalah untuk membuat prediksi yang bernilai benar.

**Logistic Regression**

Model klasifikasi yang menggunakan fungsi sigmoid untuk mengubah prediksi y’ pada model linier menjadi nilai antara 0 dan 1.

**M**

**Machine Learning**

Bidang studi yang memberi komputer kemampuan untuk belajar tanpa diprogram secara eksplisit.

**Matplotlib**

Open-source library pada Python untuk membuat plot 2D. Matplotlib membantu Anda memvisualisasikan berbagai aspek dalam machine learning.

**Missing Value**

Tidak tersedia nilai data pada variabel tertentu dalam sebuah observasi.

**Model**

Representasi dari apa yang telah dipelajari oleh sistem machine learning dari data pelatihan. Model mendefinisikan relasi antara fitur dan label.

**N**

**Neural Network**

Sebuah model yang mengambil inspirasi dari otak, terdiri dari beberapa layers yang memiliki neuron-neuron yang saling terhubung.

**Normalization**

Proses mengonversi rentang nilai aktual menjadi rentang nilai standar, biasanya dari -1 hingga +1, atau 0 hingga 1.

**Numpy**

Library matematika open-source yang menyediakan operasi array effisien dengan Python. Pandas dibangun di atas NumPy.

**O**

**One-hot-encoding**

Mengubah data kategorik dengan membuat kolom baru untuk setiap kategori.

**Outlier**

Sebuah nilai yang jauh berbeda dari kumpulan nilai lainnya dan dapat mengacaukan hasil dari sebuah analisis statistik.

**Output layer**

Lapisan “terakhir” dari jaringan saraf tiruan. Lapisan yang berisi jawaban.

**Overfitting**

Kondisi model sangat cocok dengan data pelatihan, tetapi model gagal membuat prediksi yang benar pada data baru.

**P**

**Pandas**

API analisis data yang berorientasi pada kolom. Banyak framework machine learning termasuk TensorFlow mendukung struktur data panda sebagai input.

**Parameter**

Variabel model yang dilatih oleh sistem machine learning. Sebagai contoh: weight/bobot adalah parameter yang nilainya dipelajari secara bertahap oleh sistem machine learning.

**Propagasi balik**

Algoritma untuk melakukan penurunan gradien pada jaringan saraf tiruan. Pertama, nilai keluaran dari setiap node dihitung dalam sebuah forward pass. Kemudian, turunan parsial dari eror/kesalahan yang terkait dengan setiap parameter dihitung dalam hitungan mundur melalui grafik.

**R**

**Reinforcement Learning**

Algoritma yang belajar menggunakan sistem reward dan pinalti. Algoritma Reinforcement Learning belajar agar terus mendapatkan reward dan menghindari penalti.

**Reward**

Dalam reinforcement learning, reward adalah hasil numerik dari mengambil tindakan dalam suatu keadaan seperti yang didefinisikan oleh environment. Reward diberikan saat agent berhasil menyelesaikan tantangan.

**S**

**Scikit learn**

Platform machine learning open-source yang populer. Lihat [www.scikit-learn.org](https://scikit-learn.org/stable/).

**SVM**

Algoritma training yang bertujuan untuk memaksimalkan margin antara pola pelatihan dan batas keputusan (decision boundary).

**T**

**Tensorflow**

Platform machine learning berskala besar dan terdistribusi. Istilah ini juga merujuk pada layer API dasar yang mendukung komputasi umum pada grafik aliran data.

**Test set**

Bagian dari dataset yang digunakan untuk menguji sebuah model setelah model melalui pemeriksaan awal oleh validation set.

**Training set**

Bagian dari dataset yang digunakan untuk melatih model.

**U**

**Underfitting**

Model machine learning memiliki kemampuan prediksi yang buruk karena model tersebut belum menangkap kompleksitas data pelatihan.

**Unsupervised learning**

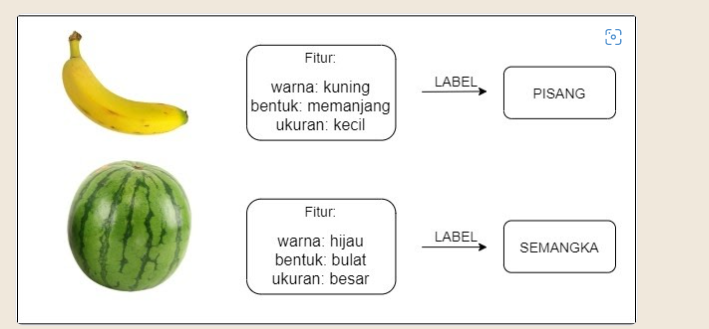
Model machine learning yang belajar dan menemukan pola dalam sekumpulan data tanpa label.

**V**

**Validation set**

Bagian dari dataset (berbeda dari training set) yang digunakan sebagai validasi.

Pada supervised learning kita melatih sekumpulan data yang memiliki label. Label adalah pengenal dari sebuah data. Misal buah A memiliki atribut atau fitur berwarna hijau, bentuknya bulat, dan ukurannya sebesar bola sepak. Buah B atributnya berwarna kuning, bentuknya agak memanjang, dan besarnya segenggaman tangan. Buah yang memiliki atribut seperti yang disebutkan pada buah A dikenali sebagai semangka, sedangkan buah dengan atribut seperti disebutkan pada buah B dikenal sebagai pisang. Maka label dari buah A adalah semangka dan label buah B adalah pisang.



Submission

Proyek Akhir : Klasifikasi Gambar

* [**Pengantar**](https://www.dicoding.com/academies/184/tutorials/8547/submission-guidance#pills-pengantar)

* [**Kriteria**](https://www.dicoding.com/academies/184/tutorials/8547/submission-guidance#pills-kriteria)

* [**Penilaian**](https://www.dicoding.com/academies/184/tutorials/8547/submission-guidance#pills-penilaian)

* [**Lainnya**](https://www.dicoding.com/academies/184/tutorials/8547/submission-guidance#pills-lainnya)

Selamat, Anda telah berada di akhir pembelajaran dalam akademi ini. Anda sudah mempelajari dasar-dasar machine learning dan bagaimana jaringan saraf bekerja. Untuk bisa lulus dari akademi ini, Anda harus mengirimkan submission berupa program jaringan saraf tiruan menggunakan TensorFlow. Program Anda harus mampu mengenali bentuk tangan yang membentuk gunting, batu, atau kertas.

Berikut kriteria submission yang harus Anda penuhi:

* Dataset yang dipakai haruslah dataset berikut : [rockpaperscissors](https://github.com/dicodingacademy/assets/releases/download/release/rockpaperscissors.zip), atau gunakan link ini pada wget command: <https://github.com/dicodingacademy/assets/releases/download/release/rockpaperscissors.zip>.
* Dataset harus **dibagi** menjadi **train set** dan **validation set**.
* Ukuran validation set harus **40%** dari total dataset (data training memiliki 1314 sampel, dan data validasi sebanyak 874 sampel).
* Harus mengimplementasikan **augmentasi gambar.**
* Menggunakan image **data generator**.
* Model harus menggunakan **model sequential**.
* Pelatihan model tidak melebihi **waktu 30 menit**.
* Program dikerjakan pada **Google Colaboratory**.
* **Akurasi** dari model minimal **85%**.
* Dapat **memprediksi gambar** yang diunggah ke Colab seperti gambar di bawah.
* Manambahkan **data diri** (sesuai profil Dicoding) pada **submission/project** yang dikirimkan.

Submission Anda akan dinilai oleh reviewer dengan **penilaian bintang** **berskala 1-5** berdasarkan dari parameter yang ada.

Anda dapat menerapkan beberapa **saran** untuk mendapatkan nilai tinggi, berikut sarannya:

* Akurasi dari model di atas 85%
* Anda menggunakan lebih dari 1 hidden layer.
* Menerapkan lebih banyak augmentasi gambar.
* Anda menggunakan optimizer dan loss-function yang tidak diajarkan di kelas.

Detail penilaian submission:



Semua ketentuan terpenuhi, namun terindikasi melakukan plagiat.



Semua ketentuan terpenuhi, namun penulisan kode masih perlu diperbaiki.



Semua ketentuan terpenuhi namun hanya mengikuti seperti apa yang ada pada modul.



Semua ketentuan terpenuhi dan akurasi dari program di atas 95%.



Semua ketentuan terpenuhi, akurasi di atas 96%, dan menggunakan tiga atau lebih teknik yang tidak diajarkan di modul seperti penggunaan Callback.

Jika **submission Anda ditolak** maka **tidak ada penilaian**. Kriteria penilaian bintang di atas hanya berlaku **jika submission Anda lulus**.

### Submission yang Tidak Sesuai Kriteria

Jika submission Anda tidak sesuai dengan kriteria, maka akan ditolak oleh reviewer, berikut poin-poinnya:

* Akurasi dari model Anda di bawah 85%.
* Proses pelatihan model Anda memakan waktu lebih dari 30 menit.
* Tidak menambahkan **data diri** (sesuai profil Dicoding) pada **submission/project** yang dikirimkan.

### Resources dan Tips

Berikut **Tips** yang dapat Anda terapkan untuk mempermudah Anda menyelesaikan submission:

* Model merupakan klasifikasi multi kelas sehingga loss function yang digunakan bukan binary\_crossentropy.
* Pastikan Anda membagi direktori untuk image data generator sesuai dengan jumlah label.
* Untuk export project yang Anda kerjakan di Colaboratory sebagai berkas .ipynb, klik tombol file yang berada di **pojok kiri atas** Colaboratory dan pilih **download** .ipynb.

### Forum Diskusi

Jika mengalami kesulitan, Anda bisa menanyakan langsung ke forum diskusi. [https://www.dicoding.com/academies/184/discussions](https://www.dicoding.com/academies/184/discussions?query=&query_criteria=&sort=&sort_direction=&title=&tutorial=8547&keywords=&creator=).

Beberapa diskusi berikut ini dapat digunakan untuk membantu submission Anda:

* [diskusi1](https://www.dicoding.com/academies/184/discussions/68117),
* [diskusi2](https://www.dicoding.com/academies/184/discussions/50629), atau
* [diskusi3](https://www.dicoding.com/academies/184/discussions/51999).

### Ketentuan Berkas Submission

Beberapa poin yang perlu diperhatikan ketika mengirimkan berkas submission:

* Menggunakan bahasa pemrograman **Python**.
* Dataset menggunakan data yang disediakan di **resource**.
* Mengirimkan pekerjaan Anda dalam bentuk berkas**.ipynb**.
* Program yang Anda kirim pastikan adalah **berkas .ipynb yang sudah dieksekusi/dijalankan**. Sederhananya, jalankan semua cell pada submission Anda. Ketika seluruh output telah keluar, baru simpan program anda dalam format .ipynb.

### Ketentuan Proses Review

Beberapa hal yang perlu Anda ketahui mengenai proses review:

* Tim penilai akan mengulas submission Anda dalam waktu **selambatnya 3 (tiga) hari kerja**, tidak termasuk hari sabtu, minggu dan libur nasional.
* Tidak disarankan untuk melakukan submit berkali-kali karena akan memperlama proses penilaian yang dilakukan tim penilai.
* Anda akan mendapat notifikasi hasil pengumpulan submission Anda via email, atau Anda dapat mengecek status submission pada akun Dicoding Anda.